Structural Results and Applications for Perturbed Markov Chains

by

Daniel Vial

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2020

Doctoral Committee:

Associate Professor Vijay Subramanian, Chair
Assistant Professor Siddhartha Banerjee, Cornell University
Researcher Laurent Massoulie, INRIA
Professor Mark Rudelson
Professor R. Srikant, University of Illinois at Urbana-Champaign
Professor Lei Ying

Daniel Vial
dvial@umich.edu
ORCID iD: 0000-0003-2426-5604

ACKNOWLEDGEMENTS

First and foremost, thanks to my advisor, Vijay Subramanian, for an incredibly enriching Ph.D. experience. I especially appreciate your patience as I found my footing in the early years of my Ph.D., and your flexibility as my interests evolved in later years. I know you'll be a mentor, collaborator, and friend moving forward, and I look forward to it. I'd also like to acknowledge my other committee members: Sid Banerjee, Laurent Massoulie, Mark Rudelson, R. Srikant, and Lei Ying. I admire you all greatly as researchers, so it was both an honor to share my work with you and invaluable to receive your feedback. In addition to Vijay, Mark, and Lei, I learned a lot from many other faculty at Michigan; thanks especially to Demos Teneketzis for excellent technical instruction but also deeper lessons about the research process. Moving back further, thanks to Professors Soura Dasgupta, Anton Kruger, and Raghuraman Mudumbai at the University of Iowa, who first encouraged me to pursue a Ph.D. and provided the instruction and research opportunities to prepare me for it.

My time in Ann Arbor was made far more enjoyable by my fellow graduate students. Thanks to all of Vijay's students, especially Hsu Kao, Mehrdad Moharrami, Shih-Tang Su, and Dengwang Tang, who I spent over four years with. I learned a lot from each of you – most notably, Mehrdad's random graphs reading group helped me write Chapters III and VI of this thesis, and Dengwang's mixing times reading group helped me write Chapter V. You were all great colleagues and even better friends. I'll remember in particular our long drives to conferences spent debating soccer, politics, and everything in between. I also appreciate my other office mates over the years, including Greg Ledva, John Lipor, and Mohammad Rasouli, for providing just the right amount of distraction to keep me productive.

On a more personal note, thanks to my parents, Mary and Paul, for their years of unwavering support. Writing a thesis requires great deal of curiosity and a good work ethic; you instilled both of these traits in me at a young age. Thanks also to Jeff and Amy, my oldest friends; the time we spent together during my Ph.D. years was always the perfect tonic for the stress of graduate school. Finally, thanks to Mathilde; your sense of adventure first inspired me to forgo a traditional Silicon Valley job for the uncertainty of Ph.D. research.

There are many others to thank, but in hopes of not prolonging an already-lengthy thesis, I'll just say: my deepest gratitude to all those who have befriended me, taught me, encouraged me, or otherwise helped me achieve this dream.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Each day, most of us interact with a myriad of networks: we search for information on the web, connect with friends on social media platforms, and power our homes using the electrical grid. Many of these interactions have improved our lives, but some have caused new societal issues — social media facilitating the rise of fake news, for example. The goal of this thesis is to advance our understanding of these systems, in hopes improving beneficial interactions with networks while reducing the harm of detrimental ones.

Our primary contributions are threefold. First, we devise new algorithms for estimating Personalized PageRank (PPR), a measure of similarity between nodes in a network used in applications like web search and recommendation systems. In contrast to most existing PPR estimators, our algorithms exploit local graph structure to reduce estimation complexity. We show the analysis of such algorithms is tractable for certain random graph models, and that the key insights obtained from these models hold empirically for real graphs.

Our second contribution is to apply ideas from the PPR literature to two other problems. First, we show that PPR estimators can be adapted to the policy evaluation problem in reinforcement learning. More specifically, we devise policy evaluation algorithms inspired by existing PPR estimators that reduce the sample complexity of existing methods when certain side information is available. Second, we use analytical ideas from the PPR literature to show that convergence behavior and robustness are intimately related for a certain class of Markov chains.

Finally, we study social learning over networks as a model for the spread of fake news. For this model, we characterize the learning outcome in terms of a novel measure of the "density" of users spreading fake news. Using this characterization, we devise optimal strategies for seeding fake news spreaders so as to disrupt learning. These strategies empirically outperform intuitive heuristics on real social networks (despite not being provably optimal for such graphs) and thus provide new insights regarding vulnerabilities in social learning.

While the topics studied in this thesis are diverse, a unifying mathematical theme is that of perturbed Markov chains. This includes perturbations that yield useful interpretations in various applications, that provide algorithmic and analytical advantages, and that disrupt some underlying system or process. Throughout the thesis, the perturbed Markov chain theme guides our analysis and suggests more general methodologies.

CHAPTER I

Introduction

1.1 Motivation

In today's world, we constantly interact with networks, defined simply as sets of pairwise-connected objects. Examples include the Internet (websites connected by hyperlinks), social networks (users connected by friendships), and even the human brain (neurons connected by synapses). Among these networks, those that have emerged recently have dramatically disrupted how we interact with friends and colleagues, distribute and acquire information, conduct business, and undergo countless other activities. While some of these changes have been beneficial – the Internet facilitating access to information, for example – others have been detrimental – social networks enabling the rise of fake news, for example. In both cases, however, networks lie at the heart of important societal issues.

The primary goal of this thesis is to advance our understanding of networks, in hopes of either improving the solutions they present or mitigating the problems they cause. At a high level, our contributions are as follows. First, we develop algorithms that can help practitioners better utilize network-related data, with motivating applications including Internet search and recommendation systems. Second, we prove structural results that give new insights regarding how networks are organized and how processes unfold over networks, with search and recommendation again as motivation, and also with an eye toward fake news.

Developing such algorithms and proving such results is challenging for (at least) two reasons. First, modern networks have complex interconnections. Owing to this, an algorithm that works well for a given network may dramatically fail if the network's topology changes slightly, processes occurring over seemingly-similar networks can have strikingly different evolutions, etc. Second, modern networks are massive; studying them manually is impossible, while studying them algorithmically requires extensive computational power, clever algorithmic implementation, or (as is often the case) a combination of the two. To combat these challenges, we often abstract a network to a simplified mathematical model – namely,

a random graph – and exploit the mathematical tractability of this model to obtain rigorous insights into the behavior of an algorithm, process, or other phenomena. However, like all models, these random graphs are crude representations of real-world networks. Thus, throughout the thesis, we also validate our insights computationally on real networks.

In addition to these algorithms and structural results, a third contribution of the thesis is to apply our insights to settings beyond the study of networks. These applications are both practical and theoretical. On the practical side, we use network algorithms as inspiration for a reinforcement learning algorithm. On the theoretical side, we use network analysis tools to study the robustness and convergence behavior of certain stochastic models, and specifically the relationship between robustness and convergence.

In short, this thesis studies a diverse set of topics involving network algorithms, network organization, processes occurring over networks, reinforcement learning, and robustness of stochastic models. A unifying mathematical theme appearing across our treatment of these topics is that of perturbed Markov chains. At times, this involves perturbing some underlying chain in a manner that yields a useful interpretation for a certain application, and/or that provides desirable analytical or algorithmic properties. At other times, the perturbation is generated adversarially and disrupts some underlying system or process, and we aim to understand the deleterious effects of the perturbation.

To describe this through-line of perturbed Markov chains in more detail, and to more precisely define the problems we address, we next discuss some technical preliminaries. We then return to provide a brief overview of the thesis and a summary of our contributions.

1.2 Preliminaries

We begin by introducing the perturbed Markov chains studied most extensively in this thesis, the *PageRank* chain and its generalization to *Personalized PageRank*. We then discuss some applications of these Markov chains found in the literature. Finally, we describe several important mathematical properties that will be exploited throughout the thesis.

1.2.1 PageRank

The PageRank chain is defined in terms of a given discrete-time, time-homogeneous, finite-state Markov chain $\{X_t\}_{t\in\mathbb{Z}_+}$, where $\mathbb{Z}_+ = \{0, 1, \ldots\}$. We assume for simplicity that $\{X_t\}_{t\in\mathbb{Z}_+}$ has state space $[n] = \{1, \ldots, n\}$, and we denote its transition matrix by P, i.e.

$$\mathbb{P}(X_{t+1} = j | X_t = i) = P(i, j) \ \forall \ i, j \in [n], t \in \mathbb{Z}_+.$$

Then for $\alpha \in (0, 1)$, the PageRank chain corresponding to P is the chain with transition matrix $(1 - \alpha)P + \alpha 1_n 1_n^\intercal / n$, where 1_n is the length-n column vector of ones. Thus, we

obtain the PageRank chain by an α-bounded perturbation of the given transition matrix P. This is one example of a perturbation that yields a desirable analytical property: since $(1-\alpha)P + \alpha 1_n 1_n^\mathsf{T}/n$ is irreducible and aperiodic (without assumption on P), the PageRank chain has a unique stationary distribution, i.e. a unique nonnegative row vector π satisfying

$$\pi = \pi\left((1-\alpha)P + \frac{\alpha 1_n 1_n^\mathsf{T}}{n}\right), \quad \sum_{i=1}^{n} \pi(i) = 1.$$

In the network science literature, $\{X_t\}_{t\in\mathbb{Z}_+}$ is typically the simple random walk[1] on some underlying graph $G = (V, E)$, where V is a set of n nodes and E is a set of directed edges of the form $i \to j$ for $i, j \in V$. In this case, the PageRank chain has a particularly simple interpretation: from the current state, flip a coin that lands heads with probability $1-\alpha$; if heads, take a random walk step; if tails, "restart" the walk by choosing the next state uniformly at random from V. This interpretation was proposed in [1] as a model for web browsing called the *random surfer* model. Here V represents a set of web pages, and $i \to j \in E$ means page $i \in V$ contains a hyperlink leading to page $j \in V$. Thus, the random surfer navigates the web by either clicking a random hyperlink (i.e. taking a random walk step) or typing a random page's web address into the address bar (i.e. restarting the walk). If i is a particularly popular web page, in the sense that many other pages link to it, the random surfer frequently visits i, and $\pi(i)$ is large. Owing to this, PageRank was first used to identify popular web pages for ordering Internet search results. It has since been viewed as a centrality measure for networks in diverse domains; we discuss some examples soon.

1.2.2 Personalized PageRank (PPR)

PPR is a natural generalization of PageRank. Given a distribution σ over $[n]$ (viewed as a column vector), the PPR vector π_σ is the stationary distribution of the chain with transition matrix $(1-\alpha)P + \alpha 1_n \sigma^\mathsf{T}$. This corresponds to a random surfer who either chooses X_{t+1} from X_t's neighbors (as in Section 1.2.1) or restarts at a state sampled from σ. Note the latter case generalizes the uniform restart of the PageRank chain; put differently, the PageRank vector is precisely the PPR vector in the special case $\sigma = 1_n/n$, i.e. $\pi = \pi_{1_n/n}$. In the same manner as PageRank, we view the PPR chain as a perturbation of the original chain.

An important special case is $\sigma = e_i$, where e_i the vector with 1 in the i-th coordinate and zeroes elsewhere; to simplify notation, we write such PPR vectors as $\pi_i = \pi_{e_i}$. Note that on the corresponding PPR chain, conditioned on restarting, the random surfer deterministically restarts at i. Conceptually, this leads to a simple inverse-distance interpretation: if j is

[1] By simple random walk, we mean X_{t+1} is chosen uniformly at random from X_t's outgoing neighbors in G, i.e. from those $j \in V$ with $X_t \to j \in E$. In the PageRank literature, one typically adds a self-loop to states with no outgoing neighbors; note $\{X_t\}_{t\in\mathbb{Z}_+}$ is not irreducible in this case, but the PageRank chain still is.

"close" to i in the graph, the random surfer will often visit j between restarts at i, and thus the PPR value $\pi_i(j)$ will be large. Moreover, many networks exhibit *homophily*, meaning "similar" nodes tend to be connected – for example, social network users from nearby geographic areas and of similar ages are more likely to be friends. Thus, if i and j are similar, they will be close in the graph (perhaps friends themselves, or sharing a mutual friend), and $\pi_i(j)$ will be large by the inverse-distance viewpoint. For this reason, PPR values are often interpreted as measures of similarity or relevance between nodes.

The case $\sigma = e_i$ is of particular interest owing to the linearity property derived in Section 1.2.4, which states that any PPR vector π_σ can be written as a convex combination of $\{\pi_i\}_{i=1}^n$. Put differently, if we are given $\{\pi_i\}_{i=1}^n$, we can compute any PPR vector π_σ, including the PageRank vector π. For this reason, one should view $\{\pi_i\}_{i=1}^n$ as the primitive objects of our study. Given the importance of these primitives, we use the notation Π for the matrix with rows $\{\pi_i\}_{i=1}^n$, and at times we call Π the *PPR matrix*.

1.2.3 Motivating applications

Having explained the centrality/influence interpretation of PageRank and the similarity/relevance interpretation of PPR, we describe some applications that have exploited these viewpoints. As already mentioned, PageRank was originally used to rank Internet search results [1]. One downside to this approach is that it models all Internet users by the same random surfer; namely, one that restarts at a uniformly random web page. More realistically, each user has personal preferences that influence which pages they are likely to restart at. These preferences are naturally encoded by a distribution σ over the set of pages; the PPR vector π_σ can then be used to rank search results while accounting for these preferences. This idea of personalized web search was in fact the genesis of PPR [2].

Beyond web search, PageRank and PPR have been used in many other practical settings. For example, Twitter has employed PPR to provide users with personalized recommendations of who to follow [3]. Here nodes in the underlying graph represent Twitter users and edges represent follower relationships. Thus, user i's PPR vector π_i can be used to identify similar users j that i does not currently follow (those j for which $\pi_i(j)$ is large but $i \to j \notin E$), and Twitter can recommend that i follow j. A similar idea was used for personalized video recommendations on YouTube [4]. More broadly, PageRank and PPR have been used in diverse fields such as bioinformatics [5, 6].

In addition to these practical examples, PageRank and PPR have proven useful in graph-theoretic problems. For instance, PPR has been used to detect communities near a seed node: the set of j for which $\pi_i(j)$ is large can be viewed as a community (i.e. a subset of densely-connected nodes) surrounding i. This intuitive viewpoint can in fact be made

rigorous [7, 8, 9]; for instance, [7] shows that the resulting community has low *conductance*, a traditional measure of how tightly interconnected a community is. As another example, PPR has been used as a primitive to assess structural similarity between graphs [10].

The examples discussed in this section are far from exhaustive; we point the reader to [11] for a survey of applications. Moving forward, we discuss specific applications infrequently, as we will typically be concerned with abstract estimation problems and structural properties. Nevertheless, it is worth noting the widespread utility of PageRank and PPR; they can potentially be useful in any application where a graph arises.

1.2.4 Key properties

We next describe some key properties of PageRank and PPR used throughout the thesis. The first property is a closed-form expression for PPR: to derive it, first observe

$$\pi_\sigma = \pi_\sigma \left((1-\alpha)P + \alpha 1_n \sigma^\mathsf{T}\right) = (1-\alpha)\pi_\sigma P + \alpha\sigma^\mathsf{T},$$

where the first equality holds by definition of π_σ and the second holds since $\pi_\sigma 1_n = 1$ (assuming we normalize π_σ so it sums to 1). We then solve for π_σ to obtain[2]

$$\pi_\sigma = \alpha\sigma^\mathsf{T}(I-(1-\alpha)P)^{-1} = \alpha\sigma^\mathsf{T}\sum_{t=0}^{\infty}(1-\alpha)^t P^t. \tag{1.1}$$

Consequently, recalling from Section 1.2.2 that Π is the matrix with rows $\{\pi_i\}_{i=1}^n$, we have

$$\Pi = \alpha(I-(1-\alpha)P)^{-1} = \alpha\sum_{t=0}^{\infty}(1-\alpha)^t P^t. \tag{1.2}$$

We refer to (1.1) and (1.2) as the *power iteration*, since the summations can be estimated by iteratively computing certain powers. For example, in (1.2) we can compute the $(t+1)$-th summand from the t-th as $(1-\alpha)^{t+1}P^{t+1} = (1-\alpha)^t P^t \times (1-\alpha)P$.

From (1.1), we derive a linearity property: given two distributions σ_1, σ_2 over $[n]$ (viewed as column vectors as in Section 1.2.2) and some $\lambda \in (0,1)$, (1.1) immediately implies

$$\pi_{\lambda\sigma_1+(1-\lambda)\sigma_2} = \lambda\pi_{\sigma_1} + (1-\lambda)\pi_{\sigma_2}. \tag{1.3}$$

[2]One can argue $(I-(1-\alpha)P)$ is invertible using the Perron-Frobenious theorem: if instead $(I-(1-\alpha)P)x = 0$ for some $x \neq 0$, then $Px = x/(1-\alpha)$, but P cannot have eigenvalue $1/(1-\alpha) > 1$ since it is row stochastic.

Note this extends to any finite mixture of distributions. In particular, we can write

$$\pi = \pi_{1_n/n} = \pi_{\sum_{i=1}^n e_i/n} = \frac{1}{n}\sum_{i=1}^n \pi_{e_i} = \frac{1}{n}\sum_{i=1}^n \pi_i,$$

i.e. the PageRank vector is the average of the PPR vectors.

The final property is a consequence of (1.1) and (1.3): if the initial state X_0 is distributed as σ, and if T is a Geometric(α) random variable independent of $\{X_t\}_{t\in\mathbb{Z}_+}$, we have

$$\begin{aligned}\mathbb{P}(X_T = j) &= \sum_{i=1}^n \sum_{t=0}^\infty \mathbb{P}(X_T = j | T = t, X_0 = i)\mathbb{P}(T = t)\mathbb{P}(X_0 = i) \qquad (1.4)\\ &= \sum_{i=1}^n \sum_{t=0}^\infty P^t(i,j) \times \alpha(1-\alpha)^t \times \sigma(i) = \sum_{i=1}^n \sigma(i)\pi_i(j) = \pi_\sigma(j).\end{aligned}$$

In words, (1.4) says we can sample from the distribution π_σ by simulating a Geometric(α)-length trajectory beginning at a state drawn from σ; we call this the *perfect sampling* property (chains exhibiting this property are more generally called Doeblin chains [12, 13]).

It is worth noting that (1.1), (1.3), and (1.4) are extremely special properties that need not hold for general Markov chains. Thus, as alluded to in Section 1.1, we can view PageRank and PPR as perturbations that yield desirable analytical properties. Moreover, as will be discussed shortly, these properties lead to algorithmic advantages as well.

1.3 Overview of the thesis

Equipped with the definitions and key properties of PageRank and PPR, and having presented motivating applications, we provide an overview of the thesis and our contributions. Thematically, the thesis is organized into three parts, which we discuss in turn.

1.3.1 Part 1: Exploiting local structure in PPR estimation

The first part of thesis considers algorithms for estimating certain submatrices of Π, or estimating Π itself. Such estimators are necessary because computing Π via the matrix inversion in (1.2) is infeasible in many of the applications discussed in Section 1.2.3, since this computation has $O(n^3)$ complexity and n may be on the order of 10^9 or greater. Before discussing our contributions, we give a brief survey of the most relevant existing algorithms so as to contextualize our work. We also note that Sections 2.2 and 3.5.2 contain more thorough PPR literature reviews and more detailed comparisons to our algorithms.

1.3.1.1 Existing algorithms and context of our work

Most PPR estimators are derived in some manner from the power iteration (1.2) and/or the perfect sampling property (1.4). Indeed, we have already interpreted (1.2) algorithmically in our discussion of the power iteration (see Section 1.2.4). Note in particular that for large T, $\Pi(\cdot, i)$ (the i-th column of Π) can be estimated with complexity $O(Tn^2)$ using T matrix-vector multiplications. Conceptually, this approach works backward from i to explore all paths of length at most T leading to i in the underlying graph. Improving upon this idea, [8] provides an algorithm called `Approx-Contributions` that estimates $\Pi(\cdot, i)$ by exploring only "high-probability" paths at lower complexity. Analogously, [7] proposes an algorithm called `Approx-PageRank` that estimates $\pi_i = \Pi(i, \cdot)$ by forward exploration of high-probability paths. From a probablistic perspective, (1.4) immediately suggests estimating π_i via Monte Carlo, i.e. by sampling many Geometric(α)-length trajectories from i. Several variants of this scheme were proposed in [14]. The ideas from [8] and [14] were later combined in [15] for an algorithm called `Bidirectional-PPR`. As its name suggests, `Bidirectional-PPR` estimates a single PPR value $\pi_s(t) = \Pi(s, t)$ in two stages: random walks are sampled forward from source node $s \in V$ and `Approx-Contributions` is run backward from target node $t \in V$.

The algorithms of [7, 8, 14, 15] all feature rigorous accuracy and complexity guarantees but estimate only a subset of the entries of Π. Clearly, these algorithms can be run repeatedly to estimate all entries of Π – e.g. one can use the algorithm of [8] to separately estimate each column of Π – but this is intuitively wasteful as it ignores dependencies across entries arising from the common underlying graph. For instance, if $\pi_i(j)$ and $\pi_j(k)$ are both large, the inverse-distance interpretation from Section 1.2.2 suggests $\pi_i(k)$ will be large as well. Prior work, such as [16, 17], has attempted to exploit these dependencies to reduce complexity, but these works typically lack rigorous guarantees. The fundamental difficulty is that these dependencies rely heavily on the local structure of the underlying graph G. In contrast, works with rigorous-yet-tractable analyses typically ignore local structure and express complexity in terms of macro-level graph parameters (number of nodes, number of edges, etc.)

A major contribution of this thesis is to bridge these approaches, i.e. to account for local structure while still providing rigorous guarantees. This is made possible by the random graph abstraction mentioned in Section 1.1: we consider random graphs with well-behaved local structures, which makes it tractable to exploit dependencies while providing rigorous guarantees. On the other hand, we consider models that preserve the key properties of real graphs, and thus our insights hold empirically for real graphs as well. We next discuss two settings in which we exploit local structure to accelerate PPR estimation in this fashion.

1.3.1.2 Overview of Chapter II

In Chapter II, we consider estimating the sub-matrix of Π with rows $S \subset V$ and columns $T \subset V$, denoted $\Pi(S, T)$. For example, if S represents a set of users searching for friends on a social network and T represents a set of users matching the search queries, estimating $\Pi(S, T)$ allows us to order the search results for each searching user. Our algorithms are based on the aforementioned `Bidirectional-PPR` estimator. Clearly, we can use `Bidirectional-PPR` to estimate $\Pi(S, T)$ by separately sampling random walks forward from each $s \in S$ and separately running `Approx-Contributions` backward from each $t \in T$. However, this ignores possible dependencies across the entries of $\Pi(S, T)$ that we can potentially exploit. Thus, in Chapter II we propose two accelerations of this naive approach. First, we show that random walks can be shared across S, reducing sample complexity. Second, we develop an `Approx-Contributions` variant that runs jointly across T and eliminates wasteful computations that may occur if the algorithm is run separately for each $t \in T$.

Our analysis in Chapter II shows that the complexity reduction resulting from these accelerations is most significant when S and/or T are clustered, i.e. when most $s \in S$ and/or most $t \in T$ are close in the graph. As a concrete example, we study our walk-sharing scheme on the *stochastic block model* (SBM), a common model for graphs with this clustering property. For the SBM, our method requires as few as $O(\log n / \log \log n)$ random walks when $|S| = \sqrt{n}$, whereas the naive approach requires $\Omega(\sqrt{n})$. Moreover, empirical results show that our algorithms significantly reduce the runtime of the naive approach when S, T are clustered on real-world graphs (and moderately reduce runtime when S, T are not clustered). As an application of these results, we propose a distributed PPR estimation scheme that simultaneously samples walks and partitions S so as to reduce runtime, without using a separate (and likely costly) partitioning scheme. In short, we exploit the local structural property of clustering to accelerate PPR estimation in Chapter II.

1.3.1.3 Overview of Chapter III

The connection between clustering and complexity explored in Chapter II hints at deeper structural properties, which we address in Chapter III. Here the specific property of interest is dimensionality and arises from the following (apparent) paradox: the entires of Π have a naturally transitive structure (see Section 1.3.1.1), yet is Π is full rank for any underlying graph G (see Section 1.2.4). Put differently, rank is too coarse to capture the intuitively small dimension of Π. Hence, in Chapter III we consider a different dimensionality measure; roughly, the minimal rank among a certain set of matrices ε-close to Π (in the l_∞ operator norm, which is natural for row stochastic matrices like Π). This quantity is difficult to analyze for a fixed graph, so we restrict attention to a sequence of random graphs $\{G_n = ([n], E_n)\}_{n \in \mathbb{N}}$

generated via the *directed configuration model* (DCM) [18], a means of randomly constructing a graph with a pre-specified degree distribution.

Our analysis shows that under certain assumptions, this dimensionality measure scales as $O(n^{c_1})$, $c_1 \in (0,1)$, resolving the aforementioned paradox. Our key assumption is that the in-degree distribution is sparse but heavy-tailed; for example, this roughly occurs on Twitter, where typical users follow a tiny subset of all other users and celebrities have a great number of followers. Furthermore, we show that this dimensionality measure dictates the complexity of estimating Π: since only $O(n^{c_1})$ rows of Π are truly independent, one can estimate only these rows, then recover the remaining rows as linear combinations. This allows us to show Π can be estimated with complexity $O(n^{c_2})$, $c_2 \in (1,2)$, improving upon all existing algorithms (when our assumptions hold). We note the algorithm in Chapter III is conceptually similar to those from the aforementioned [16, 17], which account for dependencies across Π but lack rigorous guarantees. Thus, as in Chapter II, abstracting to a random graph allows us to exploit structural properties and (rigorously) accelerate PPR estimation.

1.3.2 Part 2: Applications of PPR algorithms and analysis

The second part of the thesis applies algorithmic and analytical ideas from the PPR literature to two problems beyond networks. We discuss each in turn.

1.3.2.1 Overview of Chapter IV

In Chapter IV, we adapt PPR algorithms to the *policy evaluation* problem in reinforcement learning (RL). The basic object of study is a finite, discrete-time Markov decision process $(\mathcal{S}, \mathcal{A}, Q, c)$: $\mathcal{S}$ is a set of $S \in \mathbb{N}$ states, $\mathcal{A}$ is a set of $A \in \mathbb{N}$ actions, and, given state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, we incur instantaneous cost $c(s,a) \in \mathbb{R}_+$ and transition to $s' \in \mathcal{S}$ with probability $Q(s'|s,a)$. Mappings $\pi : \mathcal{S} \to \mathcal{A}$ are called *policies* and dictate the action taken at each state[3]. Thus, each policy π induces an instantaneous cost vector $c_\pi(s) = c(s, \pi(s))$ and a transition matrix $Q_\pi(s,\cdot) = Q(\cdot|s,\pi(s))$ and that depend only on s. The discounted cost incurred when using policy π and starting from state s is then

$$v_\pi(s) = \mathbb{E}\left[(1-\alpha)\sum_{t=0}^{\infty} \alpha^t c_\pi(S_t^\pi) \middle| S_0^\pi = s\right] = (1-\alpha)e_s^\mathsf{T} \sum_{t=0}^{\infty} \alpha^t Q_\pi^t c_\pi, \tag{1.5}$$

where α is a discount factor that reflects a trade-off between short- and long-term costs, and where $\{S_t^\pi\}_{t\in\mathbb{Z}_+}$ is an $\mathcal{S}$-valued Markov chain with transition matrix Q_π. Policy evaluation then refers to estimating the vector $v_\pi = \{v_\pi(s)\}_{s\in\mathcal{S}}$ for a fixed policy π.

Policy evaluation has a clear connection to PageRank and PPR: by (1.2) and (1.5), $v_\pi(s)$

[3]In Chapter IV, we use π to denote a policy for consistency with the RL literature; this is not to be confused with the PageRank vector π used in other chapters.

is the expected cost of a random state sampled from s's PPR vector on Q_π[4]. However, a key distinction in the RL setting is that we do not know Q_π explicitly and can only sample from it; more precisely, for any $s \in \mathcal{S}$, we can sample a random state distributed as $Q_\pi(s, \cdot)$ (the s-th row of Q_π, a distribution over $\mathcal{S}$). Thus, the problem considered in Chapter IV is to accurately estimate (1.5) with as few samples as possible.

In this setting, the existing approach from [19] estimates $v_\pi(s)$ by simulating many trajectories on the Q_π chain beginning at s; repeating this for each $s \in \mathcal{S}$ yields an estimate of v_π. Conceptually, this existing approach is analogous to separately estimating each primitive PPR vector. But when Q_π is known and $c_\pi = e_{s^*}$ for some $s^* \in \mathcal{S}$, estimating v_π amounts to estimating the s^*-th column of the PPR matrix, and the PPR literature suggests it is more efficient to use `Approx-Contributions` or its bidirectional variant (see Section 1.3.1.1). Motivated by this observation, our goal in Chapter IV is to extend these PPR algorithms to the setting where Q_π is not explicitly known and c_π is a general cost vector.

There is, however, a fundamental issue with our approach: `Approx-Contributions` is based on the idea of backward exploration and thus requires us to understand *columns* of Q_π, but we are only allowed to sample from *rows* of Q_π. To overcome this issue, we assume additional side information is provided; namely, a graph whose edges are a superset of those in the graph induced by Q_π. We call this the *supergraph* and argue in Chapter IV that such side information is likely available in many applications of interest.

Equipped with the supergraph, we devise an analogue of `Approx-Contributions` for the policy evaluation problem. We prove that its sample complexity is asymptotically equivalent to that of the existing approach in the worst case, and in the average case it can be significantly better. For instance, if the supergraph and cost vector are maximally sparse (in certain senses), the average-case sample complexity of our approach is $O(\log S)$, compared to $O(S \log S)$ for the existing approach. We also devise an analogue of `Bidirectional-PPR`, which we argue has lower sample complexity than other approaches if a highly-accurate estimate is desired. Finally, we discuss several other settings where our analysis could potentially be recycled to extend PPR estimators and related algorithms to other RL problems.

1.3.2.2 Overview of Chapter V

In Chapter V, we apply analytical ideas from the PPR literature to study the robustness of Markov models. More precisely, given a Markov chain with transition matrix P_n and state space $[n]$, a distribution σ_n over $[n]$, and some $\alpha_n \in (0, 1)$, we study the PPR-like perturbation $P_{\alpha_n, \sigma_n} = (1 - \alpha_n)P_n + \alpha_n 1_n \sigma_n^\mathsf{T}$. We refer to such perturbations as *restart perturbations* in Chapter V, and we view them as part of a larger class of perturbations which change each

[4]The roles of α and $1 - \alpha$ are reversed compared to other chapters for consistency with RL conventions.

row of P_n by at most α_n in total variation distance. Our main goal is to understand how this perturbation of the transition matrix affects the long-run behavior of the chain. Thus, mathematically, we study the relationship between the perturbation magnitude α_n and the error magnitude $\|\pi_n - \pi_{\alpha_n,\sigma_n}\|_{TV}$, where π_n and π_{α_n,σ_n} are the stationary distributions of the original and perturbed chains, and where $\|\cdot\|_{TV}$ denotes total variation distance.

We prove two main results in Chapter V. The first shows that for a certain class of chains, the asymptotics of $\|\pi_n - \pi_{\alpha_n,\sigma_n}\|_{TV}$ are fully characterized by the relative asymptotics of α_n and the mixing time $t_{\text{mix}}^{(n)}(\varepsilon)$ of the P_n chain (roughly, the number of steps before the distribution of the P_n chain is ε-close to π_n). More precisely, we show the following:

- If $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to 0$, $\|\pi_n - \pi_{\alpha_n,\sigma_n}\|_{TV} \to 0$ for any sequence of distributions $\{\sigma_n\}_{n\in\mathbb{N}}$.
- If $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$, $\|\pi_n - \pi_{\alpha_n,\sigma_n}\|_{TV} \to 1$ for some sequence of distributions $\{\sigma_n\}_{n\in\mathbb{N}}$.
- If $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to c \in (0,\infty)$, $\limsup_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\|_{TV} \leq 1 - e^{-c}$ for any sequence of distributions $\{\sigma_n\}_{n\in\mathbb{N}}$, and some such sequence attains the bound.

This "trichotomy" of cases echoes the results of [20, 21], along with a connection between PPR dimensionality and mixing times discussed in Section 3.7.4, which similarly show that some property of the original chain is unaffected when $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to 0$, is changed maximally when $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$, and exhibits an intermediate behavior when $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to (0,\infty)$. However, [20, 21] and Section 3.7.4 consider generative models for the underlying chain, all of which have *cutoff*, meaning

$$\lim_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon) = 1 \;\forall\; \varepsilon \in (0,1/2). \tag{1.6}$$

In contrast, our result applies to *all* chains with cutoff, provided these chains are irreducible, lazy, and reversible. In other words, we show the "trichotomy" phenomena is more general than suggested by [20, 21] and Section 3.7.4. We also note parts of our analysis – in particular, our upper bounds – apply to larger classes of chains; see Chapter V for details.

Our assumptions of irreducibility, laziness, and reversibility are common restrictions in the mixing times literature; in contrast, the assumption of cutoff is quite strong. Our second result addresses whether such an assumption is necessary. We partially answer this by showing that the weaker notion of *pre-cutoff* (which only requires the ratios in (1.6) to be uniformly bounded as $n \to \infty$) is equivalent to certain notion of "sensitivity to perturbation" (meaning $\|\pi_n - \pi_{\alpha_n,\sigma_n}\|_{TV} \to 1$ for certain α_n, σ_n). Intuitively, cutoff and pre-cutoff describe a sharp or sudden convergence to stationarity, and thus our second result suggests this sharp convergence is intimately related to perturbation sensitivity. This second result also complements the main result of [22], which shows that (1.6) is equivalent to a certain notion of "hitting time cutoff". The utility of these results is that, while different notions of cutoff

have been established for many different chains, there is at present a lack of general theory.

1.3.3 Part 3: Social learning and fake news

The third and final part of the thesis is unique in that PageRank and PPR do not appear in our analysis. Instead, Chapter VI considers a model for the spread of fake news over social networks. The model includes n agents attempting to learn an underlying true state of the world in an iterative fashion (modeling, for example, social network users debating a pair of candidates running for office). At each iteration, these agents update their beliefs about the state based on noisy observations (modeling news articles) and the beliefs of a subset of other agents (modeling discussions on social networks). These subsets may include a special type of agent we call *bots*, who attempt to convince others of an erroneous true state, rather than learn (modeling users spreading fake news). The precise form of the belief update is taken from the recent empirical work [23] and bears resemblance to the non-Bayesian social learning model of [24].

Under this model, two competing forces emerge as the learning horizon (i.e. the number of iterations) grows: agents receive more observations of the true state (beneficial to learning), but the influence of bots gradually propagates through the system (detrimental to learning). Hence, while the learning horizon has a clear effect on the learning outcome, the nature of this effect is unclear. Moreover, this effect has often been ignored in the literature; for instance, [25, 26, 24] all study models similar to ours but only consider infinite horizons.

We (partially) address this gap by considering a horizon T_n that is finite for each finite n but grows to infinity with n. Assuming the underlying graph is generated via the directed configuration model (see Section 1.3.1.3), our analysis details three potential learning outcomes: agents may learn the true state, mistake the erroneous state promoted by the bots as true, or believe the state falls between the true and erroneous ones. Which outcome occurs depends on the relative asymptotics of T_n and a quantity p_n that describes the "density" of bots in the network. This leads to several interesting consequences; for example, agents initially learn the true state but later "forget" it and believe the erroneous state to be true.

In Chapter VI, we also adopt an adversarial viewpoint and consider the problem of seeding bots so as to maximally disrupt learning. We leverage our analysis of the learning outcome to formulate the adversary's problem as an integer program in terms of the bot density p_n. While this problem can be solved exactly, we also propose an approximate solution that can be obtained at lower computational complexity. The form of this approximate solution suggests that successful adversaries carefully balance agents' influence and susceptibility to influence. For a social network like Twitter, this means targeting users with many followers (i.e. influential users) who follow very few users themselves (so that fake news tweeted by

bots will appear prominently in the targeted users' Twitter feeds). Moreover, the precise form of the approximate solution is non-obvious and empirically outperforms more intuitive heuristics on real social networks. In short, we believe our analysis provides new insights into vulnerabilities of news sharing platforms and social learning models.

Though PageRank does not appear in Chapter VI, it bears much in common with the rest of the thesis. It is conceptually related because a perturbed Markov chain appears: we show analyzing beliefs amounts to analyzing a certain random walk that bots perturb to prevent learning (see end of Section 6.1). Methodologically, it follows a random graph-based analysis similar to Chapter III; we discuss this connection more in Section 7.1.2.

1.4 Summary of contributions

In summary, the major contributions of this thesis are as follows:

- With random graphs as a key tool, we devise PPR estimators that exploit local graph structure but still admit reasonably tractable analyses. This allows us to obtain stronger theoretical guarantees than existing algorithms that similarly exploit structure, while also empirically accelerating existing algorithms that ignore structure. Along the way, we resolve an apparent paradox regarding PPR dimensionality.
- We adapt backward exploration-based PPR estimation algorithms to the problem of policy evaluation in reinforcement learning. In the worst case, our algorithm has similar performance to the existing approach; in the average case, it can offer dramatically better performance, reducing sample complexity from $O(S \log S)$ to as low as $O(\log S)$ if the supergraph and cost vector are sparse in certain senses.
- Viewing the PPR Markov chain as an adversarial perturbation of a given chain, we show the relationship between perturbation magnitude and mixing time dictates the asymptotic change in stationary distribution for a certain class of chains. We also prove that perturbation sensitivity is intimately related to a certain notion of cutoff.
- Motivated by the increasingly-prominent issue of fake news, we study a model of social learning in the presence of malicious agents. Using our random graph-based analysis of the learning outcome, we devise strategies for seeding malicious agents that empirically outperform intuitive heuristics on real graphs. These strategies also give novel insights regarding vulnerabilities in social learning.

Important note: Notation varies across chapters but each is self-contained, i.e. any notation used in a chapter is defined in that chapter. Appendices use the same notation as their corresponding chapter, e.g. Appendix A uses the same notation as Chapter II.

CHAPTER II

On the Role of Clustering in Personalized PageRank Estimation[1]

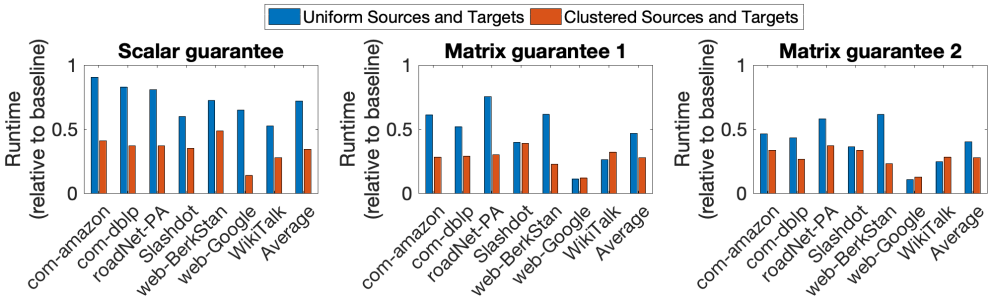
2.1 Introduction

Motivated by the widespread applications (see Section 1.2.3) of Personalized PageRank (PPR) and the need for efficient estimation algorithms (see Section 1.3.1), we consider the following problem in this chapter. We are given a directed graph $G = (V, E)$, a set of *source nodes* or *sources* $S \subset V$, and a set of *target nodes* or *targets* $T \subset V$. Our goal is to estimate $\Pi(S, T) = \{\pi_s(t)\}_{s \in S, t \in T}$, where Π is the PPR matrix and π_s is the PPR vector defined in Section 1.2.2. For instance, S could represent a set of users searching for friends on Twitter, T could represent those users matching the search queries, and estimating $\Pi(S, T)$ would provide a means of ranking search results for each searching user.

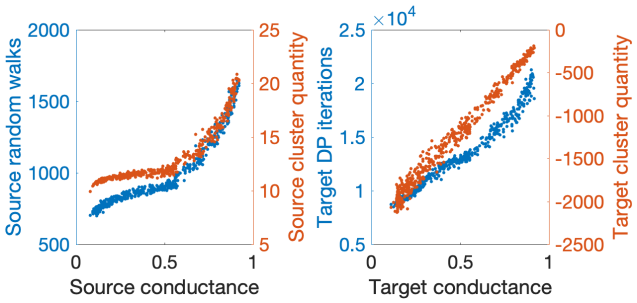
Throughout this chapter, we devise several algorithms for this task and show that their complexity decreases with increased clustering among the given sources and targets. To demonstrate the consequences of our findings, we also consider a distributed setting in which this relationship between complexity and clustering can be leveraged to design more efficient algorithms. More specifically, our contributions in this chapter are as follows:

1. In Section 2.3, we propose a variant of `Bidirectional-PPR` [15], the state-of-the-art PPR estimator for a single source/target pair (i.e. for the case $|S| = |T| = 1$). As the name suggests, `Bidirectional-PPR` estimates PPR in two stages: random walks forward from the source node and dynamic programming (DP) backward from the target node. Our algorithm, called `FW-BW-MCMC`, adds a DP stage forward from the source that allows it to serve as a primitive in the many pair setting. In Appendix A.1, we establish similar guarantees to those for `Bidirectional-PPR`.
2. In Section 2.4, we use `FW-BW-MCMC` as a primitive to estimate PPR for many pairs,

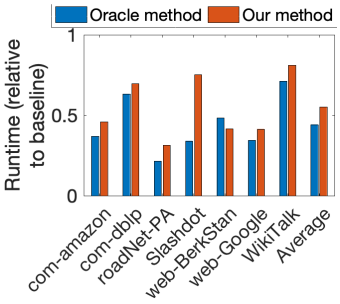
[1]This chapter is adapted from [27]. A preliminary version appeared in the abstract [28].



(a) Across a diverse set of real graphs, our algorithms accelerate baseline methods; these accelerations are most significant when the sources and targets are clustered (experiment details in Section 2.5.2).



(b) The source/target stage complexities of our methods scale with quantities that describe clustering of sources/targets, and that behave like conductance (experiment details in Section 2.5.1.1).



(c) Our findings can be used to identify clustering at runtime and accelerate PPR estimation (see Section 2.6).

Figure 2.1: Summary of empirical results.

proposing methods that accelerate the naive scheme of separately sampling walks for each source and separately running DP for each target. For the sources, we show the forward DP allows walks to be shared, decreasing the number of walks required. For the targets, we define a new iterative update for the backward DP, which eliminates repeated computations that may occur when treating each target separately. Using these ideas, we devise algorithms with accuracy guarantees on each scalar estimate and on the matrix containing all estimates. Across a diverse set of real-world graphs, our methods are roughly 1.1 to 9.3 times faster than baseline methods (Fig. 2.1a).

3. We show analytically in Section 2.4 and empirically in Section 2.5 that the accelerations offered by our algorithms are most significant when the sources and targets are each clustered together in the graph, i.e. PPR estimation is "easier" when clustering occurs. For example, our algorithms typically accelerate baseline methods by factors of 3-4 when clustering occurs (Fig. 2.1a). More specifically, we prove the number of random walks for the sources and the number of DP iterations for the targets scale with quantities that describe clustering among the sources and targets, respectively; we also find empirically that these clustering quantities scale with a more traditional clustering quantity, conductance (Fig. 2.1b). Also, while these clustering quantities are

difficult to analyze for a fixed graph, we provide analytical results for the stochastic block model, the prototypical model for networks with community structure.

4. Finally, in Section 2.6, we demonstrate an application of our results, showing that our findings can be used to devise efficient PPR estimators in a distributed setting. Specifically, we show that quantities computed during the forward DP can be used to predict the random walk sampling time for different assignments of tasks to machines, and we propose a heuristic method to compute an assignment that (locally) minimizes this time. At a high level, our method "learns" the clustering present at runtime; empirically, this learning is quite successful, in the sense that our method performs nearly as well as an oracle method that knows the clustering *a priori* (Fig. 2.1c).

The remainder of the chapter is organized as follows. We begin by discussing related work in Section 2.2. Sections 2.3-2.6 follow the outline above. We close in Section 2.7.

Notational conventions for the chapter: Throughout the chapter, $G = (V, E)$ is a directed graph with $n = |V|$ nodes and $m = |E|$ edges. For $v \in V$, let $N_{\text{out}}(v) = \{u \in V : v \to u \in E\}$ denote v's outgoing neighbors, and let $d_{\text{out}}(v) = |N_{\text{out}}(v)|$ denote the out-degree of v. For simplicity, we assume $d_{\text{out}}(v) > 0 \; \forall \; v \in V$. Similarly define $N_{\text{in}}(v)$ and $d_{\text{in}}(v)$ as v's incoming neighbors and in-degree. We let A denote the adjacency matrix of G and let D be the diagonal matrix with $D(v, v) = d_{\text{out}}(v)$. Thus, $P = D^{-1}A$ is the transition matrix for the simple random walk on G; from P, we define PageRank and PPR as in Section 1.2. In addition to these conventions, other notation will be introduced as needed.

2.2 Related work

Before proceeding, we discuss some existing PPR estimators. Broadly speaking, these can be organized hierarchically: first, those that estimate the entire PPR matrix $\{\pi_s(t)\}_{s \in V, t \in V}$; second, those that estimate a single row $\{\pi_s(t)\}_{t \in V}$ or column $\{\pi_s(t)\}_{s \in V}$ of this matrix, or its column sums (i.e. global PageRank); and third, those that estimate a single entry $\pi_s(t)$.

At the first level, several algorithms have been proposed to accelerate the power iteration or matrix inversion in (1.2). To accelerate the power iteration, [16] provides a decomposition that allows a single row of the PPR matrix to be estimated using previously-estimated rows. Hence, this yields a procedure of first computing a small number of rows and then using these to estimate other rows; we discuss this algorithm more in Chapter III. To obtain less costly matrix inversions, several works, e.g. [29, 30], have leveraged structural assumptions of the graph at hand. For example, Tong *et al.* in [29] propose a decomposition of P into a block diagonal matrix P_1 and $P_2 := P - P_1$; for graphs like social networks, P_2 can be extremely sparse. From the probabilistic viewpoint (1.4), [31] gives an algorithm to estimate any entry of the PPR matrix at runtime using a precomputed database of random walk samples.

At the second level, algorithms include the dynamic programming methods in [7] and [32] that estimate a row and a column of the PPR matrix, respectively; both can be viewed as localized versions of the power iteration in (1.2). The algorithm in [7] yields l_1 and l_∞ error guarantees on the row estimate with complexity $O(m)$, while [32] gives an l_∞ guarantee on the column estimate with complexity $O(m)$. We make use of these algorithms in our methods and will discuss them in more detail in Section 2.3. We also note the approach in [7, 32] is closely related to work by Lee and co-authors [33, 34, 35] that focuses on estimation of the stationary distribution of countable state-space Markov chains, as well as estimation in the context of general linear systems. From the probabilistic viewpoint, an important work is [14], which analyzes Monte Carlo methods for global PageRank estimation, based on both the final step of sampled random walks (as given by (1.4)) and the number of visits along the entire walk (appealing to renewal theory in the latter case). In [14], it is shown that a single walk from each node (i.e. n walks total) suffices to obtain estimates with small relative error for nodes with high global PageRank. Another work in this category is [36], which uses random walk-based methods to detect all nodes with global PageRank exceeding $n^{-\delta}, \delta \in (0, 1)$ with complexity sublinear in n. [36] also contains an algorithm to estimate a row of the PPR matrix with each estimate satisfying a multiplicative plus additive error guarantee; the complexity is linear in n (if the error tolerance is set to match [15]). Several papers have also studied distributed estimation of global PageRank; for example, [37, 38, 39] adopt a stochastic approximation viewpoint, [40] features an algorithm similar to those in the aforementioned [7, 32, 33, 34, 35], and [41] uses Monte Carlo methods.

At the third level, the aforementioned `Bidirectional-PPR` algorithm from [15] combines existing dynamic programming and Monte Carlo methods to estimate a single PPR value with worst-case and average-case complexity $O(n)$ and $O(\sqrt{m})$, respectively. From an accuracy perspective, this algorithm achieves a relative error bound for PPR values exceeding $1/n$, and an absolute error bound otherwise. We discuss in more detail in Section 2.3.

In the context of this body of work, we will consider estimation of a small set of PPR values, $\{\pi_s(t)\}_{s\in S,t\in T}$ for some $S, T \subset V$. While we do not precisely quantify "small", we implicitly assume $|S| \approx |T| = o(\sqrt{m})$. In this setting, the existing methods described above can be applied in two ways. First, using methods such as the power iteration or the dynamic programming schemes (i.e. the first two levels of the above hierarchy), one can estimate entire rows and/or columns of the PPR matrix and then discard unwanted estimates. Such approaches typically have complexity $O(|S|m)$ or $O(|T|m)$. Second, one can run the single pair estimator `Bidirectional-PPR` separately for each pair $(s, t) \in S\times T$. This approach has typical complexity $O(|S||T|\sqrt{m})$. When $|S| \approx |T| = o(\sqrt{m})$, the second approach is more efficient. Hence, we will treat this approach as a baseline for comparison to our methods.

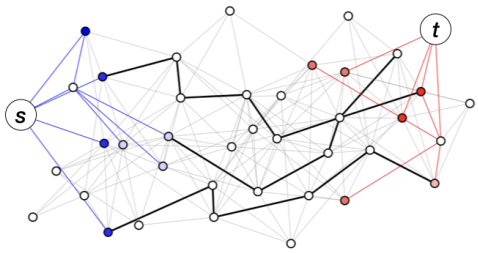


Figure 2.2: Depiction of our algorithm `FW-BW-MCMC`. Blue and red nodes/edges show forward and backward DP, respectively; black edges show random walks.

2.3 Single pair estimation

We begin by proposing a variant – in fact, a generalization – of `Bidirectional-PPR` [15]; we will introduce our algorithm and then describe `Bidirectional-PPR` as a special case. As mentioned in Section 2.2, the idea behind these estimators is to combine dynamic programming (DP) and Markov chain Monte Carlo (MCMC) to estimate $\pi_s(t)$ for some $s, t \in V$. Our algorithm uses two DP stages and one MCMC stage. We will refer to these stages as the forward DP, backward DP, and MCMC stages; hence, we call our estimator `FW-BW-MCMC`. It is depicted pictorially in Fig. 2.2 and defined formally in Algorithm 2.3. Before proceeding, we briefly describe each stage.

The forward DP stage is Algorithm 2.1. This is nearly identical to the `Approximate-PageRank` algorithm of [7], so we use the same name here; however, we change the termination criteria from $\|D^{-1}r^s\|_\infty \leq r^s_{\max}$ to $\|r^s\|_1 \leq r^s_{\max}$, where $r^s_{\max} \in (0, 1)$ is an input to the algorithm (we describe our motivation for this shortly). The algorithm takes as input $s \in V$ and produces $p^s, r^s \in \mathbb{R}^n_+$, shown in [7] to satisfy the invariant (2.1) at each iteration.

$$\pi_s(u) = p^s(u) + \sum_{w \in V} r^s(w)\pi_w(u) \ \forall \ u \in V. \tag{2.1}$$

As mentioned Section 2.2, Algorithm 2.1 can be viewed as a "localized" power iteration. At a high level, it computes elements of the matrices in (1.2) corresponding to high probability paths from s to u (the $p^s(u)$ term) while tracking the error from "uncomputed" paths (the $\sum_{w \in V} r^s(w)\pi_w(u)$ term). These high probability paths are shown as blue edges in Fig. 2.2.

The backward DP stage is `Approximate-Contributions` (Algorithm 2.2, from [32]), which is the "dual" of Algorithm 2.1: while Algorithm 2.1 works along outgoing edges, Algorithm 2.2 works along incoming edges. In [32], it is shown that Algorithm 2.2 maintains invariant (2.2), which is interpreted similarly to (2.1). This stage is shown in red in Fig. 2.2.

$$\pi_v(t) = p^t(v) + \sum_{w \in V} \pi_v(w)r^t(w) \ \forall \ v \in V. \tag{2.2}$$

To motivate the MCMC stage, we combine (2.1) and (2.2) with $u = t$ and $v = s$ to obtain

$$\pi_s(t) = p^t(s) + \langle p^s, r^t \rangle + \sum_{w,w' \in V} r^s(w)\pi_w(w')r^t(w'), \tag{2.3}$$

and so, after running the DP, only the third term in (2.3) is unknown. The goal of the MCMC is to estimate this term. Towards this end, let $\sigma_s = r^s/\|r^s\|_1$ and use (1.3) to write

$$\|r^s\|_1 \sum_{w' \in V} \sum_{w \in V} \sigma_s(w)\pi_w(w')r^t(w') = \|r^s\|_1 \sum_{w' \in V} \pi_{\sigma_s}(w')r^t(w') = \|r^s\|_1 \mathbb{E}_{U \sim \pi_{\sigma_s}} \left[r^t(U) \right].$$

Leveraging the perfect sampling property (1.4), we can then estimate this term via random walks. More specifically, we first sample a starting node from σ_s (blue nodes in Fig. 2.2), and we then sample a random walk beginning at the starting node (black edges in Fig. 2.2). This process of sampling random walks is the MCMC stage of our algorithm.

Algorithm 2.1: $(p^s, r^s) =$ `Approximate-PageRank`$(G, s, \alpha, r^s_{\max})$

Initialize $p^s = 0, r^s = e_s$
while $\|r^s\|_1 > r^s_{\max}$ **do**
  Let $v^* \in \arg\max_{v \in V} r^s(v)/d_{\text{out}}(v)$
  Set $r^s(u) \leftarrow r^s(u) + (1-\alpha)r^s(v^*)/d_{\text{out}}(v^*)\ \forall\ u \in N_{\text{out}}(v^*)$,
   $p^s(v^*) \leftarrow p^s(v^*) + \alpha r^s(v^*), r^s(v^*) = 0$

Algorithm 2.2: $(p^t, r^t) =$ `Approximate-Contributions`$(G, t, \alpha, r^t_{\max})$

Initialize $p^t = 0, r^t = e_t$
while $\|r^t\|_\infty > r^t_{\max}$ **do**
  Let $v^* \in \arg\max_{v \in V} r^t(v)$
  Set $r^t(u) \leftarrow r^t(u) + (1-\alpha)r^t(v^*)/d_{\text{out}}(u)\ \forall\ u \in N_{\text{in}}(v^*)$,
   $p^t(v^*) \leftarrow p^t(v^*) + \alpha r^t(v^*), r^t(v^*) = 0$

Algorithm 2.3: $\hat{\pi}_s(t) =$ `FW-BW-MCMC`$(G, s, t, \alpha, r^s_{\max}, r^t_{\max}, w)$

Let $(p^s, r^s) =$ `Approximate-PageRank`$(G, s, \alpha, r^s_{\max})$ (Algorithm 2.1); set $\sigma_s = \frac{r^s}{\|r^s\|_1}$
Let $(p^t, r^t) =$ `Approximate-Contributions`$(G, t, \alpha, r^t_{\max})$ (Algorithm 2.2)
for $i = 1$ **to** w **do**
  Sample random walk starting at $\nu \sim \sigma_s$ of length $\sim$ geom(α); let $X_i = r^t(U_i)$,
   where U_i is endpoint of walk
Let $\hat{\pi}_s(t) = p^t(s) + \langle p^s, r^t \rangle + \frac{\|r^s\|_1}{w} \sum_{i=1}^{w} X_i$

As mentioned above, the forward DP stage terminates when $\|r^s\|_1 \leq r^s_{\max}$ instead of when $\|D^{-1}r^s\|_\infty \leq r^s_{\max}$ (as in [7]). This is because we require a uniform bound on $\{\|r^s\|_1\}_{s \in S}$

when proving results for a set sources S in later sections. However, this bound is not needed in practice, where we can instead use $\|D^{-1}r^s\|_\infty \leq r^s_{\max}$. We call this variant of our algorithm `FW-BW-MCMC-Practical`; see Algorithm A.2 in Appendix A.2 for a formal definition.

Having defined `FW-BW-MCMC`, we describe the existing algorithm `Bidirectional-PPR`, which operates as follows: run the backward DP from t, take $v = s$ in (2.2), and estimate the unknown term $\mathbb{E}_{U\sim\pi_s}[r^t(U)]$ via random walks from s. We observe this is a special case of `FW-BW-MCMC`; naemly, the case $r^s_{\max} = 1$. We emphasize that walks are sampled from $\nu \sim \sigma_s$ in `FW-BW-MCMC` and from s in `Bidirectional-PPR`, which will be a key distinction later.

Moving forward, we will propose many pair estimators that use either `Bidirectional-PPR` or our variant as a primitive. We will show that using our variant offers runtime accelerations not possible when using `Bidirectional-PPR`. Implicit in this discussion will be an understanding that using either primitive yields similar performance when these accelerations are ignored (so that using our variant offers better performance when the accelerations are accounted for). In particular, we can prove the following results (as single pair estimation is not our focus, we defer formal statements and proofs to Appendix A.1):

1. `FW-BW-MCMC`, `FW-BW-MCMC-Practical`, and `Bidirectional-PPR` offer the same accuracy guarantee (except for mild differences in assumptions)
2. `FW-BW-MCMC` and `Bidirectional-PPR` have $O(n)$ worst-case complexity
3. `FW-BW-MCMC-Practical` and `Bidirectional-PPR` have $O(\sqrt{m})$ average-case complexity (where by average case we mean averaging over uniformly random $t \in V$)

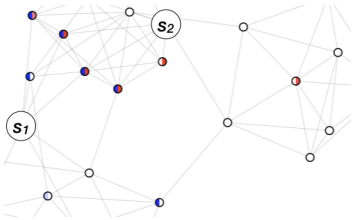
2.4 Many pair estimation

In this section, we consider the problem of estimating PPR for many node pairs, namely the set $\{\pi_s(t)\}_{s\in S,t\in T}$ for some $S, T \subset V$. We consider two variants of this problem. First, in Section 2.4.1, we view $\{\pi_s(t)\}_{s\in S,t\in T}$ as a set of scalars, each of which we aim to accurately estimate. Second, in Section 2.4.2, we view $\{\pi_s(t)\}_{s\in S,t\in T}$ as a matrix, which we aim to approximate accurately in the operator norm. For both variants, we propose algorithms that accelerate existing approaches, and we show the accelerations scale with quantities that can be interpreted as clustering measures of S and T. In addition to these algorithms, we briefly discuss variants that use precomputation in Section 2.4.3.

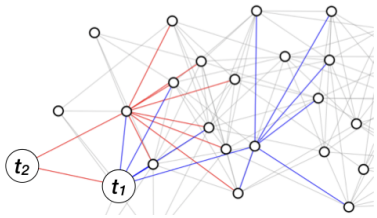
2.4.1 Scalar estimation viewpoint

A natural approach to estimate $\{\pi_s(t)\}_{s\in S,t\in T}$ is to use single pair estimators from Section 2.3 as primitives. In particular, we could use either of the following approaches:

- Run forward DP and sample random walks from $\nu \sim \sigma_s$ for each $s \in S$. Run backward DP from each $t \in T$. Compute estimates as in `FW-BW-MCMC`.
- Sample random walks from each $s \in S$. Run backward DP from each $t \in T$. Compute



(a) Source side: walks are sampled from blue nodes for s_1 and from red nodes for s_2; walks from blue *and* red nodes are shared between s_1 and s_2.



(b) Target side: red paths are computed via `Extend` during t_2 DP; blue paths can be computed via `Merge` during t_2 DP, rather than recomputed via `Extend`.

Figure 2.3: Many-pair accelerations of `FW-BW-MCMC` when $|S| = |T| = 2$.

estimates as in `Bidirectional-PPR`.

As discussed above, the primitives `FW-BW-MCMC` and `Bidirectional-PPR` are roughly equivalent in terms of complexity and accuracy; hence, both approaches have similar complexity. However, in Section 2.4.1.1, we show the source stage of the first approach (forward DP and random walks) can be accelerated in a way not possible for the second approach. Further, in Section 2.4.1.2, we show the target stage (backward DP) can be accelerated as well. Hence, using primitive method `FW-BW-MCMC` and the accelerations of Sections 2.4.1.1-2.4.1.2, we can more efficiently estimate $\{\pi_s(t)\}_{s\in S,t\in T}$.

2.4.1.1 Source stage acceleration

To accelerate the source stage, we define a unified MCMC stage for a set of sources S. At a high level, this scheme allows us to share walks across multiple $s \in S$, thereby decreasing the total number of walks required. We motivate the scheme pictorially in Fig. 2.3a, for the simple case $S = \{s_1, s_2\}$. Here blue and red depict σ_{s_1} and σ_{s_2} values, i.e. blue and red nodes are the starting nodes of random walks used in the π_{s_1} and π_{s_2} estimates, respectively. Observe several nodes have nonzero σ_{s_1} and σ_{s_2} values. The unified MCMC stage allows us to use random walks sampled from such nodes towards both estimates (π_{s_1} and π_{s_2}).

To define the unified MCMC stage, we first define an equivalent MCMC stage for a single source. Recall that in Algorithm 2.3 we sample each of w random walks in two stages: first, we sample starting node $\nu_s \sim \sigma_s$, and second, we sample a walk starting at ν_s. Equivalently, we can first sample starting nodes $\{\nu_s^{(i)}\}_{i=1}^w$ i.i.d. from σ_s, and then sample $X_s^{(w)}(v) := \sum_{i=1}^w 1(\nu_s^{(i)} = v)$ walks starting at v, for each $v \in V$. With this in mind, the unified MCMC stage proceeds as follows. First, for each $s \in S$ we sample starting nodes $\{\nu_s^{(i)}\}_{i=1}^w$ i.i.d. from σ_s (as in the single source case), and we define $X_s^{(w)}(v)$ as above. Next, we sample $X^{(w)}(v) := \max_{s\in S} X_s^{(w)}(v)$ walks starting at each $v \in V$. Letting U_i^v denote the

endpoint of the i-th walk from v, we then estimate $\pi_s(t)$ as

$$\hat{\pi}_s(t) = p^t(s) + \langle p^s, r^t \rangle + \frac{\|r^s\|_1}{w} \sum_{v \in V : X_s^{(w)}(v) > 0} \sum_{i=1}^{X_s^{(w)}(v)} r^t(U_i^v). \tag{2.4}$$

The final term in (2.4) is an unbiased estimate of $\mathbb{E}_{U \sim \pi_{\sigma_s}}[r^t(U)]$ using $\sum_{v \in V} X_s^{(w)}(v) = w$ random walks, so the accuracy guarantee of Algorithm 2.3 holds. To analyze the complexity of this scheme, we bound the total number of walks $\sum_{v \in V} X^{(w)}(v)$ in Theorem 2.1.

Theorem 2.1. Fix $\varepsilon, p_{\text{fail}} \in (0,1)$. Assume

$$w > \frac{3 \log(2 \sum_{s \in S, v \in V} 1(\sigma_s(v) > 0)/p_{\text{fail}})}{\varepsilon^2 \min_{s \in S, v \in V : \sigma_s(v) > 0} \sigma_s(v)}. \tag{2.5}$$

Then with probability at least $1 - p_{\text{fail}}$, the total number of walks $\sum_{v \in V} X^{(w)}(v)$ satisfies

$$\left| \sum_{v \in V} X^{(w)}(v) - w \sum_{v \in V} \max_{s \in S} \sigma_s(v) \right| \leq \varepsilon w \sum_{v \in V} \max_{s \in S} \sigma_s(v). \tag{2.6}$$

Proof. See Appendix A.3. □

Before proceeding, we offer several remarks on this result:

- A lower bound on w is given by (A.1) in Theorem A.1 to guarantee an accurate estimate. Thus, if w exceeds both (2.5) and (A.1), guarantees for scalar accuracy and walk complexity both hold. (In general, it is unclear which of (2.5) and (A.1) is larger.)
- In the worst case, the denominator on the right side of (2.5) may be quite small, so the assumption on w in Theorem 2.1 may be restrictive. However, this only means that the concentration in (2.6) may not provably occur, *not* that the scheme will necessarily have poor performance. We do find that this concentration essentially occurs for practical values of w, see e.g. leftmost plot in Fig. 2.5 and left two plots in Fig. 2.8.
- Moving forward, we will denote the matrix with rows $\{\sigma_s\}_{s \in S}$ by Σ (or by Σ_S, if we wish to emphasize the sources S at hand) and will write the bound in (2.6) as

$$\|\Sigma\|_{\infty,1} = \sum_{v \in V} \max_{s \in S} \sigma_s(v) \tag{2.7}$$

Here we have used the notation of the $L_{p,q}$ matrix norm, defined for a matrix A as

$$\|A\|_{p,q} = \left(\sum_j \left(\sum_i |A(i,j)|^p \right)^{q/p} \right)^{1/q}. \tag{2.8}$$

From Theorem 2.1, we expect to sample approximately $w\|\Sigma\|_{\infty,1}$ walks. It is easy to verify $\|\Sigma\|_{\infty,1} \in [1, |S|]$, so our approach requires $w|S|$ random walks in the worst case, but only w in the best case. In contrast, if we use `Bidirectional-PPR` as a primitive for many pair estimation, the unified MCMC stage is not possible (all walks used to estimate π_s begin at s, so sharing walks is not possible), and $w|S|$ walks are *always* required. In short, `FW-BW-MCMC` with the unified MCMC stage may accelerate the source stage of our many pair estimation approach. Unfortunately, it is difficult to quantify the degree of this acceleration in general, because $\|\Sigma_S\|_{\infty,1}$ depends on the forward DP, which itself is difficult to analyze. However, in Section 2.5, we offer empirical evidence that $\|\Sigma_S\|_{\infty,1}$ scales with the *conductance* of S, a common measure of the clustering of S in the underlying graph (see (2.18)). Furthermore, as will be discussed next, this quantity provably scales with clustering for the *stochastic block model* (SBM), a common model for networks with community structure. In short, when S is clustered, $\|\Sigma_S\|_{\infty,1}$ is typically small, and estimating PPR for many sources is easier.

We now turn to our result for the SBM. We consider the special case for which n is a perfect square and the graph is composed of $\sqrt{n}$ communities, each containing $\sqrt{n}$ nodes. (This allows us to compare the extremes of choosing $\sqrt{n}$ sources from the same community or from distinct communities; however, the analysis can be modified for other cases.) More specifically, we define $V_{n,i} = \{1 + (i-1)\sqrt{n}, \ldots, i\sqrt{n}\}$ and set $V_n = \cup_{i=1}^{\sqrt{n}} V_{n,i}$; we will view each $V_{n,i}$ as a community. For $v \in V_n$, we denote by $i(v)$ the unique $i \in \{1, \ldots, \sqrt{n}\}$ satisfying $v \in V_{n,i}$, i.e. $i(v)$ is the community that v belongs to. We then construct a graph $G_n = (V_n, E_n)$ as follows: for any $u, v \in V_n$ s.t. $u \neq v$, edge $u \to v$ is present with probability p_n if $i(u) = i(v)$ (i.e. if u, v are in the same community), and is present with probability q_n if $i(u) \neq i(v)$ (i.e. if u, v are in different communities), independent of other edges. We define

$$d_{\text{out}}(v) = |\{u \in V_n : v \to u \in E_n\}|, \quad d^-_{\text{out}}(v) = |\{u \in V_n \setminus V_{n,i(v)} : v \to u \in E_n\}| \quad \forall\, v \in V_n.$$

In words, $d_{\text{out}}(v)$ is v's out-degree (as before, though here it is a random variable), and $d^-_{\text{out}}(v)$ is the number of edges pointing from v to other communities.

Our analysis will assume $p_n = p$ is a constant and $q_n = o(1/\sqrt{n})$. In this case, $\mathbb{E}[d_{\text{out}}(v)] = \Theta(\sqrt{n})$ (i.e. the graph is dense) and $\mathbb{E}[d^-_{\text{out}}(v)] = o(\sqrt{n})$ (i.e. nodes prefer to connect to their own community). Also, we assume the forward DP is run for at most $o(\sqrt{n})$ iterations. Since all nodes have out-degree $\Theta(\sqrt{n})$ with high probability (see proof of Theorem 2.2), this means we dedicate at most $o(n)$ complexity to the forward DP. This is consistent with the fact that our algorithm has average-case complexity $O(\sqrt{m})$, since $\sqrt{m} = n^{3/4}$ when all out-degrees are $\Theta(\sqrt{n})$. Hence, this assumption on the number of iterations is minor. Under these assumptions, we can prove the following bounds on $\|\Sigma\|_{\infty,1}$.

Theorem 2.2. Let $\{G_n = (V_n, E_n)\}_{n\in\mathbb{N}:\sqrt{n}\in\mathbb{N}}$ be the sequence of SBMs described above, with $p_n = p$ for some constant $p \in (0,1)$ and $q_n = o(1/\sqrt{n})$. Assume we run the forward DP for at least one iteration, but at most $o(\sqrt{n})$ iterations. Then the following hold:

- Let $S_n = V_{n,i}$ for some i (i.e. all sources belong to the same community). If $q_n = \Omega(\log n/n)$ (i.e. cross-community connections are dense), then for some constant $C > 0$,

$$\lim_{n\to\infty} \mathbb{P}\left(\|\Sigma_{S_n}\|_{\infty,1} \le Cq_n n\right) = 1,$$

  i.e. $\|\Sigma_{S_n}\|_{\infty,1} = O(q_n n) = o(\sqrt{n})$ with high probability. If instead $q_n = \Theta(1/n)$ (i.e. cross-community connections are sparse), then for some constant $C > 0$,

$$\lim_{n\to\infty} \mathbb{P}\left(\|\Sigma_{S_n}\|_{\infty,1} \le C\log n/\log\log n\right) = 1,$$

  i.e. $\|\Sigma_{S_n}\|_{\infty,1} = O(\log n/\log\log n)$ with high probability.
- Let $S_n \subset V_n$ with $|S_n| = \sqrt{n}$ and $i(s) \neq i(s')\ \forall\ s, s' \in S_n$ s.t. $s \neq s'$ (i.e. each source belongs to a distinct community). Then for any constant $\delta \in (0,1)$,

$$\lim_{n\to\infty} \mathbb{P}\left(\|\Sigma_{S_n}\|_{\infty,1} \ge (1-\delta)\sqrt{n}\right) = 1,$$

  i.e. $\|\Sigma_{S_n}\|_{\infty,1} \in [(1-\delta)\sqrt{n}, \sqrt{n}]$ with high probability.

Proof. See Appendix A.4. □

2.4.1.2 Target stage acceleration

Our next goal is to accelerate the target stage of the many pair estimation approach. We motivate our approach in the simple case $T = \{t_1, t_2\}$. Assume that p^{t_1}, r^{t_1} have been computed by Algorithm 2.2, and that p^{t_2}, r^{t_2} are currently being computed. If $r^{t_2}(t_1) > r^t_{\max}$ at some iteration, we can use the alternate update rule

$$p^{t_2} \leftarrow p^{t_2} + r^{t_2}(t_1)p^{t_1}, \quad r^{t_2} \leftarrow r^{t_2} + r^{t_2}(t_1)(r^{t_1} - e_{t_1}). \tag{2.9}$$

When p^{t_2}, r^{t_2} are updated via (2.9), the invariant (2.2) is maintained. Indeed, for any $s \in V$,

$$\begin{aligned} &p^{t_2}(s) + r^{t_2}(t_1)p^{t_1}(s) + \sum_{u\in V}\pi_s(u)(r^{t_2}(u) + r^{t_2}(t_1)(r^{t_1}(u) - e_{t_1}(u))) \\ &\quad = p^{t_2}(s) + \sum_{u\in V}\pi_s(u)r^{t_2}(u) + r^{t_2}(t_1)((p^{t_1}(s) + \sum_{u\in V}\pi_s(u)r^{t_1}(u)) - \pi_s(t_1))) \qquad (2.10) \\ &\quad = \pi_s(t_2) + r^{t_2}(t_1)(\pi_s(t_1) - \pi_s(t_1)) = \pi_s(t_2), \qquad (2.11) \end{aligned}$$

where in (2.10) we rearranged terms and in (2.11) we assume p^{t_1}, r^{t_1} and p^{t_2}, r^{t_2} satisfy (2.2).

We can interpret (2.9) as follows. As discussed in Section 2.3, we view Algorithm 2.2 as a method of traversing paths to t and computing the probability of these paths. For the update in Algorithm 2.2, specific paths are extended by a single step at each iteration; we call this update `Extend`. In contrast, (2.9) extends paths by (potentially) many steps in an iteration; specifically, by appending paths to t_1, with paths from t_1 to t_2, to obtain paths to t_2. We call this update `Merge` to highlight the fact that paths are merged in this manner.

The utility of `Merge` is that the probability of paths to t_2 through t_1 need not be recomputed one step at a time via `Extend`. This is depicted in Fig. 2.3b: red paths are computed via `Extend` during t_2 DP; blue paths, having already been computed via `Extend` during t_1 DP, are used to compute longer paths in a single iteration via `Merge` during t_2 DP. In contrast, blue paths would be recomputed one step at a time via `Extend` during t_2 DP, if separate DP was used. In short, `Merge` may allow Algorithm 2.2 to terminate in fewer iterations. This is made more specific in Proposition 2.1.

Proposition 2.1. Suppose $T = \{t_1, t_2\}$ and $\pi_{t_1}(t_2) > r^t_{\max}$. If we run Algorithm 2.2 for t_2 and use `Merge` whenever $v^* = t_1$, the algorithm terminates in at most $\frac{n\pi(t_2)}{\alpha r^t_{\max}} - \frac{(\|p^{t_1}\|_1 - \alpha)}{\alpha}$ iterations. If `Merge` is not used, the algorithm terminates in at most $\frac{n\pi(t_2)}{\alpha r^t_{\max}}$ iterations.

Proof. See Appendix A.5. □

From Algorithm 2.2, $\|p^{t_1}\|_1 \geq \alpha$. Hence, Proposition 2.1 allows us to tighten the iteration bound by $\frac{(\|p^{t_1}\|_1 - \alpha)}{\alpha} \geq 0$ (with equality if and only if the algorithm terminates in a single iteration for t_1). More generally, the iterations we save roughly scales with the quantity

$$c_T = \sum_{i=1}^{|T|} \left| \left\{ j \in \{1, 2, ..., i-1\} : \pi_{t_j}(t_i) > r^t_{\max} \right\} \right|, \tag{2.12}$$

assuming the nodes in T are chosen in order $\{t_1, t_2, \ldots, t_{|T|}\}$. We note the choice of this order has a clear impact on performance, but optimizing it at runtime is difficult; we discuss this more in Appendix A.7. See Algorithm 2.4 for our many target algorithm.

We next offer a clustering interpretation of the quantity c_T. For this, note $\pi_{t_j}(t_i) > r^t_{\max}$ is a notion of "closeness" between t_i and t_j; hence, c_T is a notion of clustering of the set T, and our analysis suggests estimating PPR for many targets is easier when the targets are clustered. Note that, while the source clustering quantity $\|\Sigma\|_{\infty,1}$ from Section 2.4.1.1 is *smaller* when clustering among sources is more significant, the target clustering quantity c_T is *larger* when clustering among targets is more significant; in Section 2.5, we show $-c_T$ scales with the conductance of T in practice.

Algorithm 2.4: $\{(p^t, r^t)\}_{t\in T} =$ `Approx-Cont-Many-Targets`$(G, T, \alpha, r^t_{\max})$

for $i = 1$ **to** $|T|$ **do**
  $p^{t_i} = 0, r^{t_i} = e_{t_i}$
  while $\|r^{t_i}\|_\infty > r^t_{\max}$ **do**
    $v^* \in \arg\max_{v\in V} r^{t_i}(v)$
    if $v^* \in \{t_1, \ldots, t_{i-1}\}$ **then**
      $p^{t_i} \leftarrow p^{t_i} + r^{t_i}(v^*)p^{v^*}, r^{t_i} \leftarrow r^{t_i} + r^{t_i}(v^*)(r^{v^*} - e_{v^*})$ (i.e. use (2.9))
    else
      $r^{t_i}(u) \leftarrow r^{t_i}(u) + (1-\alpha)\frac{r^{t_i}(v^*)}{d_{\text{out}}(u)}\ \forall\ u \in N_{\text{in}}(v^*)$,
       $p^{t_i}(v^*) \leftarrow p^{t_i}(v^*) + \alpha r^{t_i}(v^*), r^{t_i}(v^*) = 0$

2.4.2 Matrix approximation viewpoint

For the second variant of the many pair estimation problem, we view $\{\pi_s(t)\}_{s\in S,t\in T}$ as a matrix that we aim to accurately approximate. For simplicity, we assume $|S| = |T| = l$, and we denote these sets $S = \{s_i\}_{i=1}^l, T = \{t_i\}_{i=1}^l$. We also assume $V = \{1, 2, ..., n\}$, and we let Π denote the matrix of dimension $n \times n$ whose (i, j)-th element is $\pi_i(j)$. In this notation, we seek an estimate $\hat{\Pi}(S, T)$ of $\Pi(S, T)$ that minimizes $\|\hat{\Pi}(S, T) - \Pi(S, T)\|_2$, where for a matrix A, $A(I, J)$ denotes the submatrix of A containing rows I and columns J, and where $\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2$ is the operator norm.

Before proceeding, we introduce additional notation used in this section. Similar to the $A(I, J)$ notation, $A(I, :)$ and $A(:, J)$ are the submatrices with rows I and all columns, and all rows and columns J, respectively. For a vector x, $x(I)$ is the vector with elements I; when x has nonzero entries, $\text{diag}(x)$ and $\text{diag}(1/x)$ are the diagonal matrices whose i-th diagonal elements are $x(i)$ and $1/x(i)$, respectively. Finally, we will encounter stable rank, which for a matrix A is defined as $\text{srank}(A) = (\|A\|_F/\|A\|_2)^2$, where $\|\cdot\|_F = \|\cdot\|_{2,2}$ is the Frobenius norm, with $\|\cdot\|_{2,2}$ defined as in (2.8). It is straightforward to verify $1 \le \text{srank}(A) \le \text{rank}(A)$ by writing $\|A\|_F^2$ and $\|A\|_2^2$ in terms the singular values of A (see e.g. [42, Section 2.1.15]).

With this notation in mind, we define the following matrices:

$$P_S \in \mathbb{R}^{n\times l} \text{ s.t. } P_S(i, j) = p^{s_j}(i), \quad R_S \in \mathbb{R}^{n\times l} \text{ s.t. } R_S(i, j) = r^{s_j}(i), \tag{2.13}$$

$$P_T \in \mathbb{R}^{n\times l} \text{ s.t. } P_T(i, j) = p^{t_j}(i), \quad R_T \in \mathbb{R}^{n\times l} \text{ s.t. } R_T(i, j) = r^{t_j}(i). \tag{2.14}$$

Here p^{s_j}, r^{s_j} and p^{t_j}, r^{t_j} are computed via Algorithms 2.1 and 2.4, respectively. We may then collect the invariant (2.3) for each (s_i, t_j) pair in matrix form as

$$\Pi(S, T) = P_T(S, :) + P_S^\mathsf{T} R_T + R_S^\mathsf{T} \Pi R_T. \tag{2.15}$$

Observe only $R_S^\mathsf{T}\Pi R_T$ is unknown in (2.15). Hence, we consider estimation of this term. To this end, let σ be any n-length vector satisfying $\sigma(i) > 0 \; \forall \; i \in \{1, 2, ..., n\}$ and $\sum_{i=1}^n \sigma(i) = 1$; note we may view σ as a distribution on V. We then rewrite the unknown term in (2.15) as

$$R_S^\mathsf{T}\Pi R_T = R_S^\mathsf{T}\text{diag}(1/\sigma)\text{diag}(\sigma)\Pi R_T. \tag{2.16}$$

Using (2.16), we can obtain unbiased estimates of $R_S^\mathsf{T}\Pi R_T$ as follows. Let $\{\mu_i\}_{i=1}^w$ be i.i.d. samples from σ. For $i \in \{1, 2, ..., w\}$, let $\nu_i \sim \pi_{\mu_i}$ independently (where we sample from π_{μ_i} using a random walk, as given by (1.4)), and let $X_i = R_S^\mathsf{T}\text{diag}(1/\sigma)e_{\mu_i}e_{\nu_i}^\mathsf{T}R_T$. It is straightforward to see $\mathbb{E}[e_{\mu_i}e_{\nu_i}^\mathsf{T}] = \text{diag}(\sigma)\Pi$; hence, $\mathbb{E}[X_i] = R_S^\mathsf{T}\Pi R_T$. We may then estimate $\Pi(S, T)$ as $\hat{\Pi}(S, T) = P_T(S, :) + P_S^\mathsf{T}R_T + \frac{1}{w}\sum_{i=1}^w X_i$.

We will consider two forms of σ for this approach. Specifically, let us define

$$\sigma_{\text{avg}}(i) = \frac{1}{l}\sum_{s\in S}\sigma_s(i), \quad \sigma_{\max}(i) = \frac{1}{\|\Sigma\|_{\infty,1}}\max_{s\in S}\sigma_s(i), \tag{2.17}$$

where $\sigma_s = r^s/\|r^s\|_1$ as before. Observe that when $\sigma \in \{\sigma_{\text{avg}}, \sigma_{\max}\}$, the assumption $\sum_{i=1}^n \sigma(i) = 1$ is satisfied. Furthermore, we argue that the assumption $\sigma(i) > 0$ is without loss of generality in these cases. Indeed, suppose $\sigma(j) = 0$ for some j and $\sigma(i) > 0$ for $i \neq j$. Then $\mathbb{P}[\mu_i = j] = 0$ by definition, and by (2.17), $r^s(j) = 0 \; \forall \; s \in S$. It is then readily verified that $R_S(V \setminus \{j\}, :)^\mathsf{T}\text{diag}(1/\sigma(V \setminus \{j\}))e_{\mu_i}e_{\nu_i}^\mathsf{T}R_T$ is an unbiased estimate of $R_S^\mathsf{T}\Pi R_T$. Given this simple fix, we assume $\sigma(i) > 0$ moving forward.

Algorithm 2.5: $\hat{\Pi}(S, T) =$ `FW-BW-MCMC-Many-Pair`$(G, S, T, \alpha, r^t_{\max}, r^s_{\max}, w)$

for $i = 1$ **to** l **do**
  $(p^{s_i}, r^{s_i}) =$ `Approximate-PageRank`$(G, s_i, \alpha, r^s_{\max})$ (Algorithm 2.1)
$\{(p^t, r^t)\}_{t\in T} =$ `Approx-Cont-Many-Targets`$(G, T, \alpha, r^t_{\max})$ (Algorithm 2.4)
Construct P_S, R_S, P_T, R_T via (2.13), (2.14); compute $\sigma = \sigma_{\text{avg}}$ or $\sigma = \sigma_{\max}$ via (2.17)
for $i = 1$ **to** w **do**
  Let $X_i = R_S^\mathsf{T}\text{diag}(1/\sigma)e_{\mu_i}e_{\nu_i}^\mathsf{T}R_T$, where ν_i is endpoint of walk starting at $\mu_i \sim \sigma$ of length $\sim \text{geom}(\alpha)$
Let $\hat{\Pi}(S, T) = P_T(S, :) + P_S^\mathsf{T}R_T + \frac{1}{w}\sum_{i=1}^w X_i$

To summarize, we have proposed the matrix approximation scheme formally defined in Algorithm 2.5. Theorem 2.3 provides a guarantee for the accuracy of this scheme.

Theorem 2.3. Fix $\varepsilon > 0$. If $\sigma = \sigma_{\text{avg}}$ in Algorithm 2.5, assume

$$w \geq l^2\sqrt{\text{srank}(\Pi(S, T))}\log(2l/p_{\text{fail}})r^s_{\max}r^t_{\max}(6 + 4\varepsilon)/(3\varepsilon^2).$$

If instead $\sigma = \sigma_{\max}$ in Algorithm 2.5, assume

$$w \geq l^{3/2}\|\Sigma\|_{\infty,1} \log(2l/p_{\text{fail}}) r_{\max}^s r_{\max}^t (6+4\varepsilon)/(3\varepsilon^2).$$

Then for both choices of σ, and with probability at least $1 - p_{\text{fail}}$, Algorithm 2.5 returns an estimate $\hat{\Pi}(S,T)$ satisfying $\|\Pi(S,T) - \hat{\Pi}(S,T)\|_2 \leq \varepsilon \max\{\|\Pi(S,T)\|_2, 1\}$.

Proof. See Appendix A.6 □

Neglecting common factors, Theorem 2.3 states w scales with l^2 and $l^{3/2}$ in the best case for σ_{avg} and $\sigma_{\max}$, respectively; in the worst case, w scales with $l^{5/2}$ for both approaches. In the next section, we compare $\sqrt{l \text{ srank}(\Pi(S,T))}$ with $\|\Sigma\|_{\infty,1}$ empirically to compare the "typical" case. We also observe Theorem 2.3 shows that, as in the scalar estimation viewpoint of Section 2.4.1, PPR matrix approximation is easier when clustering occurs. This is because, when $\sigma = \sigma_{\max}$, complexity scales with $\|\Sigma\|_{\infty,1}$ (which we have argued is measure of clustering of S); when $\sigma = \sigma_{\text{avg}}$, complexity scales with $\text{srank}(\Pi(S,T))$, a measure of matrix dimensionality. Additionally, stable rank is unique from the clustering quantities introduced thus far in that it takes into account both S and T (unlike $\|\Sigma\|_{\infty,1}$, which only accounts for S, or c_T, which only accounts for T). Finally, we comment on a difference for the choices of σ. In particular, when $\sigma = \sigma_{\max}$, one can set w proportional to $\|\Sigma\|_{\infty,1}$ before sampling random walks, leveraging clustering at runtime to increase efficiency. In contrast, when $\sigma = \sigma_{\text{avg}}$, the scaling factor in the w lower bound is the unknown quantity $\text{srank}(\Pi(S,T))$. However, we propose using $\text{srank}(P_T(S,:) + P_S^{\mathsf{T}} R_T)$ (known at runtime) as a surrogate for $\text{srank}(\Pi(S,T))$. In Section 2.5, we show empirically that using this surrogate yields performance similar to using $\text{srank}(\Pi(S,T))$.

2.4.3 Precomputation variants

While we have thus far assumed all computations are done online, one can also consider variants for which some computations are done offline, with the results stored for later use. In fact, in Section 4 of [15], the authors propose several such algorithms for the case of one source $s \in V$ and many targets $T \subset V$, using `Bidirectional-PPR` as a primitive. Each of these variants proceeds as follows. For the offline stage, `Approx-Contributions` is run for every $t \in V$, and the vectors $\{p^t, r^t\}_{t \in V}$ are stored. For the online stage, random walks are sampled from s, and $\{\pi_s(t)\}_{t \in T}$ are estimated using the endpoints of these walks and $\{p^t, r^t\}_{t \in T}$. As mentioned, several such algorithms are proposed; these only differ in how the vectors are stored and how the walks and vectors are combined to generate estimates. In particular, the basic framework of running `Approx-Contributions` offline and sampling walks from s online is used in all of the precomputation algorithms from [15].

Analogous to our extension of `Bidirectional-PPR` from single to many pairs, we can extend these precomputation algorithms from the single source case to the many sources case. Specifically, we can modify each of these algorithms in two ways (but otherwise leave them unchanged). First, we can modify the offline stage by also precomputing and storing $\{p^s, r^s\}_{s \in V}$ via `Approx-PageRank`. Second, we can modify the online stage by sampling walks using the precomputed vectors $\{r^s\}_{s \in S}$ and the walk sharing scheme from Section 2.4.1.1.

To assess the performance of this approach, we compare against the naive extension of [15]'s precomputation algorithms to the case $|S| > 1$; namely, leaving the offline stage unchanged and sampling walks separately from each $s \in S$ online. Clearly, our approach requires more storage (due to running `Approx-PageRank` offline); however, this storage will be roughly double that of the naive extension and thus will not increase the order of the space complexity. On the other hand, our approach will accelerate the online stage of this naive extension, since fewer random walks will typically be sampled. Specifically, per Section 2.4.1.1, we expect to sample $w\|\Sigma\|_{\infty,1}$ walks instead of $w|S|$ walks; as discussed previously, the former quantity can be much smaller if S is clustered.

We also note that Algorithm 2.4 can be used to compute $\{p^t, r^t\}_{t \in V}$ offline, though this is a minor point, since offline computational complexity is generally not a concern. However, this raises another point. When precomputation is not allowed, our source and target accelerations are both used at runtime; when precomputation is allowed, only our source acceleration is used at runtime. Hence, the runtime savings of our schemes may be less significant in the precomputation setting. In spite of this, we believe the savings will still be considerable in general. This belief follows from the fact that, in our experiments, the source acceleration is generally at least as significant as the target acceleration. For example, Fig. 2.4 shows that the number of random walks sampled grows more slowly in $|S|$ than the number of DP iterations grows in $|T|$. Additionally, Fig. 2.7 shows that for fixed $|S|, |T|$, walk savings and DP iteration savings are comparable across a wide range of graphs.

2.5 Experiments

In this section, we demonstrate the empirical performance of our algorithms and the role of clustering in their performance. We conduct experiments using both synthetic and real graphs. On the synthetic side, we use a directed Erdős-Rényi graph and directed stochastic block model (referred to hereafter as `Direct-ER` and `Direct-SBM`, respectively), each with $n = 2 \times 10^3$ and $\mathbb{E}[m] = 2 \times 10^4$. For the real datasets, we use a set of graphs from the Stanford Network Analysis Platform [43] including social networks (`Slashdot`, `Wiki-Talk`), partial web crawls (`web-BerkStan`, `web-Google`), co-purchasing and co-authoring graphs (`com-amazon`, `com-dblp`), and a road network (`roadNet-PA`). In addition to the diverse

domains of these datasets, they differ in terms of sparsity (in order of magnitude, each has 10^6 edges, but the number of nodes ranges from 10^4 to 10^6), so we believe our empirical results are robust. We also note that error bars depict standard deviation across experimental trials, while for scatter plots without error bars, each dot represents a single trial. For further experimental documentation, we point the reader to Appendix A.8. In particular, Table A.2 in Appendix A.8 documents algorithmic parameters used. We chose these parameters so the primitive algorithms `FW-BW-MCMC` and `Bidirectional-PPR` yield similar accuracy ($\approx 10\%$ relative error) while balancing runtime between the DP and MCMC stages of the algorithm in the single pair case. Note the analysis in Appendix A.1 shows that balancing runtime in this manner minimizes overall complexity; hence, for both algorithms, our chosen parameters optimize runtime subject to an accuracy constraint, providing a fair comparison. Finally, our experimental code is available at https://github.com/danielvial/clusteringPpr.

2.5.1 Synthetic data

2.5.1.1 Scalar estimation

We first compare `FW-BW-MCMC` with `Bidirectional-PPR` when computing $\pi_s(t)\ \forall\ (s,t) \in S \times T$ as $|S|$ and $|T|$ grow on `Direct-ER`. More specifically, for `FW-BW-MCMC` we use the $\|D^{-1}r^s\|_\infty \leq r^s_{\max}$ forward DP scheme as in `FW-BW-MCMC-Practical`, sample walks using the scheme from Section 2.4.1.1, and use Algorithm 2.4 for backward DP; for `Bidirectional-PPR`, we sample walks separately from each $s \in S$ and run backward DP separately for each $t \in T$. Results are shown in Fig. 2.4. Note the number of random walks sampled and number of backward DP iterations grow more slowly with $|S| = |T|$ using `FW-BW-MCMC`, due to the accelerations proposed in Sections 2.4.1.1 and 2.4.1.2, respectively. As a result, runtime grows more slowly using `FW-BW-MCMC`. In Fig. 2.4, we also show the clustering quantities (2.7) and (2.12). We observe the source clustering quantity $\|\Sigma\|_{\infty,1}$ has a concave shape, which corresponds to the apparent sublinear growth of random walks as $|S|$ grows. Additionally, the target clustering quantity c_T has a convex shape; since backward DP iteration *savings* scale with c_T, we expect DP iterations to correspondingly "flatten", which indeed occurs. These observations empirically validate the key insights of Section 2.4.1: namely, that the estimation schemes proposed have complexities that scale with the identified clustering quantities $\|\Sigma\|_{\infty,1}$ and c_T. We also plot $\text{srank}(\Pi(S,T))$ on the runtime plot; note it appears to flatten along with runtime as $|S|, |T|$ grow. Finally, these plots remain similar as n grows, though the improvement of our scheme over the existing one increases; see Appendix A.8.

Next, to further examine the effect of clustering, we use `Direct-SBM`. We fix $|S| = |T| = 100$ and sample S and T from decreasingly clustered sets via the following scheme: we first sample S, T from a single community, we then sample S, T from two communities, etc., until

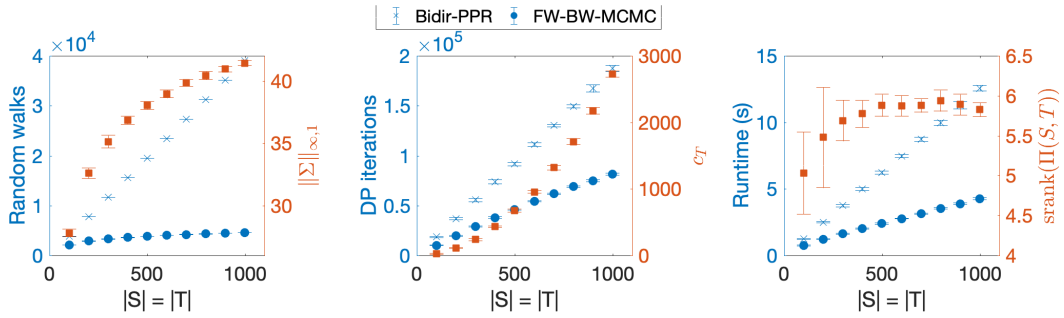


Figure 2.4: On `Direct-ER`, random walks, backward DP iterations, and runtime scale more slowly in $|S|, |T|$ for our method `FW-BW-MCMC` when compared to the existing method `Bidirectional-PPR`.

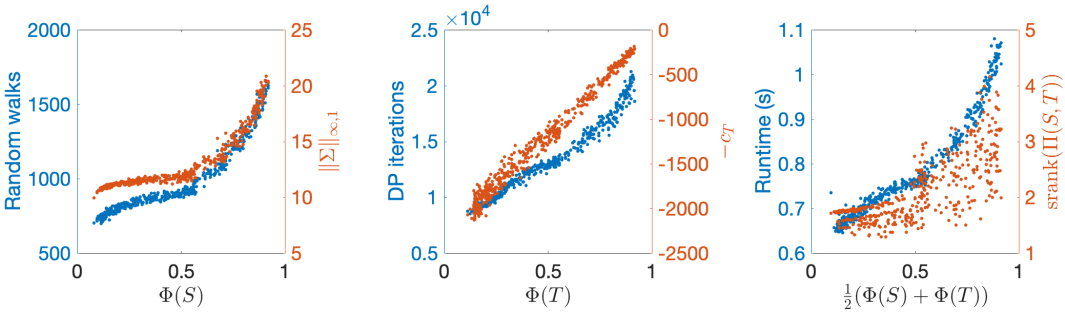


Figure 2.5: When clustering is significant, fewer random walks and backward DP iterations yield faster runtime for our method on `Direct-SBM`; additionally, our clustering measures roughly scale with conductance.

we sample S, T from the entire graph, allowing us to observe a wide range of clustering. As in the previous experiment, we are interested in how algorithmic performance relates to $\|\Sigma\|_{\infty,1}$ and c_T. Here, we also compare these quantities to a clustering measure commonly used in the graph theory literature (see e.g. [7]), *conductance*, defined for $U \subset V$ as

$$\Phi(U) = \frac{\sum_{i \in U, j \notin U} A_{ij}}{\min\{\sum_{u \in U} d_{\text{out}}(u), \sum_{u \notin U} d_{\text{out}}(u)\}}. \tag{2.18}$$

In Fig. 2.5, we observe fewer random walks are sampled when $\Phi(S)$ is small (when S is significantly clustered); similarly, the backward DP converges in fewer iterations when $\Phi(T)$ is small (when T is significantly clustered). Also, Fig. 2.5 shows that $\|\Sigma\|_{\infty,1}$ grows with $\Phi(S)$ and $-c_T$ grows with $\Phi(T)$. In short, our identified clustering quantities behave similar to conductance. In the runtime plot, we again show $\text{srank}(\Pi(S,T))$ as a measure of overall complexity; this quantity (roughly) grows with $\frac{1}{2}(\Phi(S) + \Phi(T))$, as does runtime.

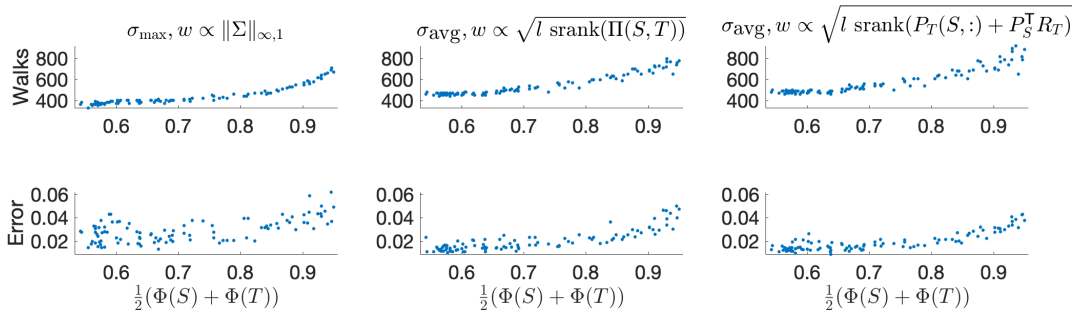


Figure 2.6: On `Direct-SBM`, our matrix approximation schemes are most efficient when clustering is significant; additionally, the surrogate $\text{srank}(P_T(S,:)+P_S^\mathsf{T}R_T))$ performs similar to $\text{srank}(\Pi(S,T))$.

2.5.1.2 Matrix approximation

We now document performance of our matrix approximation scheme (Algorithm 2.5) using `Direct-SBM` and the S, T sampling strategy from the previous experiment. We compare three cases: $\sigma = \sigma_{\max}$ with $w \propto \|\Sigma\|_{\infty,1}$, $\sigma = \sigma_{\text{avg}}$ with $w \propto \sqrt{l \text{ srank}(\Pi(S,T))}$, and $\sigma = \sigma_{\text{avg}}$ with $w \propto \sqrt{l \text{ srank}(P_T(S,:) + P_S^\mathsf{T} R_T)}$. These cases are motivated by Theorem 2.3, which states that the sample requirements for $\sigma = \sigma_{\max}$ and $\sigma = \sigma_{\text{avg}}$ are $\|\Sigma\|_{\infty,1}$ and $\sqrt{l \text{ srank}(\Pi(S,T))}$, respectively (neglecting common factors); additionally, since $\text{srank}(\Pi(S,T))$ is unknown in practice, we proposed using $\text{srank}(P_T(S,:)+P_S^\mathsf{T}R_T)$ as a surrogate in the discussion following the theorem. Results are shown in Fig. 2.6. Observe that for all three cases, fewer walks are sampled when S and T are clustered (i.e. when $\frac{1}{2}(\Phi(S)+\Phi(T))$ is small; nevertheless, error remains roughly constant (in fact, when clustering is present, error is somewhat lower despite fewer walks being sampled). Further, we observe $\sigma_{\max}$ and σ_{avg} have similar performance, in terms of complexity and accuracy. Finally, we note the results for the $\text{srank}(\Pi(S,T))$ and $\text{srank}(P_T(S,:)+P_S^\mathsf{T}R_T)$ cases are quite similar, suggesting that $\text{srank}(P_T(S,:) + P_S^\mathsf{T} R_T)$ is an appropriate surrogate for $\text{srank}(\Pi(S,T))$.

2.5.2 Real data

2.5.2.1 Scalar estimation

We next compare `FW-BW-MCMC` with `Bidirectional-PPR` as in Section 2.5.1.1, but here using real datasets. We fix $|S| = |T| = 1000$ and randomly sample S, T using two different schemes: sampling uniformly among all nodes and using an algorithm described in Appendix A.8 to build clustered subsets of nodes; we find these schemes typically give conductance values ≈ 0.99 and ≈ 0.5, respectively, allowing us to observe two degrees of clustering. In Fig. 2.7, we show random walk count, DP iteration count, and runtime for our method relative to the corresponding values using `Bidirectional-PPR`. Averaging across the diverse

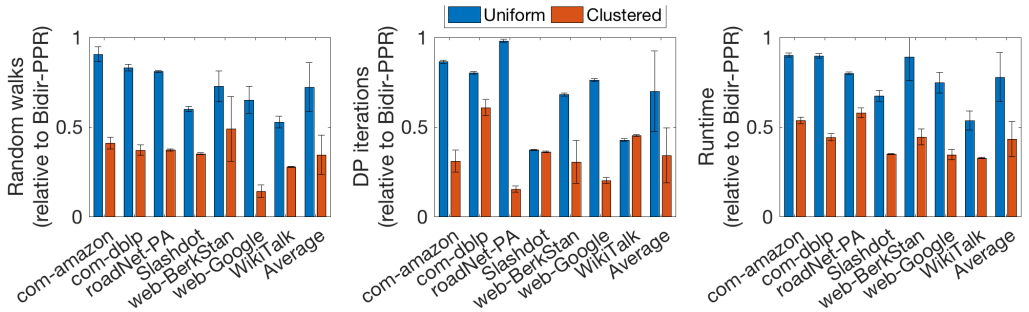


Figure 2.7: On real graphs, our scalar methods are typically 1.4 and 2.9 times faster than existing methods when S, T are chosen uniformly and clustered, respectively, due to fewer random walks and DP iterations.

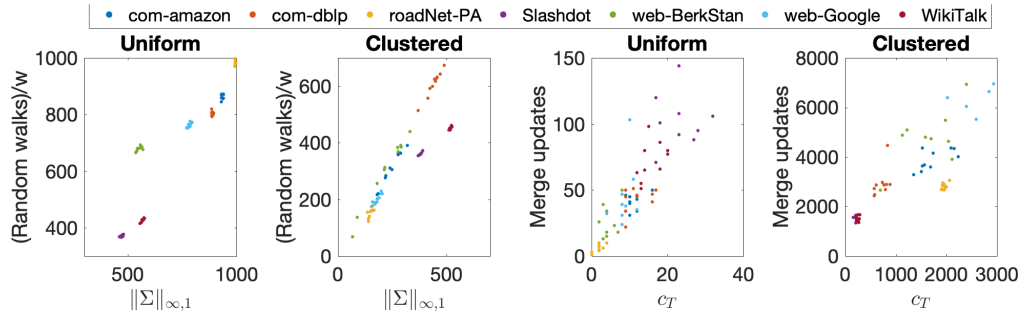


Figure 2.8: On real graphs, random walks and `Merge` updates scale with clustering quantities $\|\Sigma\|_{\infty,1}$ and c_T, empirically validating the analysis of Section 2.4.1.

set of graphs considered, our method is approximately 1.4 times faster in the uniform case and 2.9 times faster in the clustered case, highlighting the efficiency of our algorithms and the impact of clustering on their performance. Additionally, we note our method is at least twice as fast for all datasets in the clustered case. For the same experiment, we also show random walk count (normalized to w) and the number of `Merge` updates (i.e. the number of DP iterations saved when compared to existing methods) in Fig. 2.8. From Theorem 2.1 and Proposition 2.1, we expect these quantities to scale linearly with the identified clustering quantities $\|\Sigma\|_{\infty,1}$ and c_T, respectively; from Fig. 2.8, we observe this scaling roughly occurs.

2.5.2.2 Matrix approximation

Finally, we test our matrix approximation scheme (Algorithm 2.5) on real graphs. Here we also compare to a baseline method that does not leverage clustering among targets and sources. In particular, we run backward DP separately for each target, rather than using the accelerated scheme as in Algorithm 2.5. Additionally, the baseline method uses no forward DP, i.e. we set $r^s_{\max} = 1$ in Algorithm 2.5, so that $p^s = 0, r^s = e_s \ \forall \ s \in S$. Note that, in this case, both the $\sigma_{\max}$ and σ_{avg} schemes reduce to sampling $\mu_i \sim S$ uniformly, sampling $\nu_i \sim \pi_{\nu_i}$ using a random walk, and estimating $\Pi(S, T)$ as $\hat{\Pi}(S, T) = P_T(S, :) + \frac{1}{w}\sum_{i=1}^{w} X_i$,

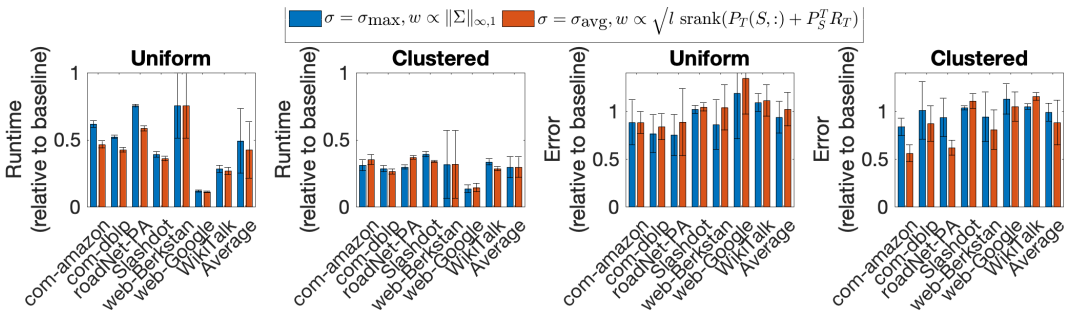


Figure 2.9: On real graphs, our matrix approximation schemes are significantly faster than the baseline method (which uses no forward DP) with comparable accuracy; this is most notable when S, T are clustered.

where $X_i = [e_{s_1}\ e_{s_2}\ \cdots\ e_{s_l}]^\mathsf{T} e_{\mu_i} e_{\nu_i}^\mathsf{T} R_T$ is an unbiased estimate of $\Pi(S,:)R_T$. We reemphasize that walks are *not* shared among sources for this baseline scheme, i.e. clustering among sources is not leveraged to improve performance. For the baseline scheme, we set $w \propto l$, and we compare performance to the $\sigma_{\max}$ scheme with $w \propto \|\Sigma\|_{\infty,1}$ and the σ_{avg} scheme with $w \propto \sqrt{l \text{ srank}(P_T(S,:) + P_S^\mathsf{T} R_T)}$. Results are shown in Fig. 2.9, with quantities shown for the $\sigma_{\max}$ and σ_{avg} schemes relative to the baseline scheme. Averaging across datasets, the $\sigma_{\max}$ and σ_{avg} schemes are over twice as fast as the baseline scheme when S, T are chosen uniformly and 3.4 times faster when S, T are clustered; additionally, the accuracy of both schemes is comparable to the baseline (and slightly better on average). We also note both our schemes are at least twice as fast as the baseline for all graphs in the clustered case.

2.6 Application: distributed random walk sampling

Thus far, our main finding has been that PPR estimation complexity scales with quantities that describe clustering among sources and/or targets. In this section, we demonstrate one application of these findings; namely, that these findings can be used to efficiently estimate $\{\pi_s\}_{s \in S}$ online when several machines are available and when offline storage is permitted. More specifically, we consider the following distributed computational setting:

- k machines are available for parallel computation and a central machine is available to facilitate the parallel computation (for simplicity, we assume $k \in \{|S|, |S|/2, |S|/3, \ldots\}$)
- $\{p^t, r^t\}_{t \in V}$ have been precomputed via Algorithm 2.2 and are stored offline

Using the existing method as a primitive, a baseline strategy for this estimation task is as follows: arbitrarily partition S into k subsets of size $|S|/k$, use the i-th machine to sample random walks from each source s belonging to the i-th subset, and estimate π_s using the endpoints of walks from s and $\{p^t, r^t\}_{t \in V}$ (as in the primitive method `Bidirectional-PPR`).

We propose the following alternative (using `FW-BW-MCMC` as a primitive). First, we arbitrarily partition S into k subsets of size $|S|/k$, and we use the i-th machine to run forward

DP (Algorithm 2.1) for each source s belonging to the i-th subset. Second, we use the central machine to construct another partition $\{S_i\}_{i=1}^k$ of S, in a manner we discuss shortly. Third, we use the i-th machine to run the accelerated source stage from Section 2.4.1.1 for the subset of sources S_i. Finally, we estimate π_s as in the primitive method `FW-BW-MCMC`.

It remains to specify how to construct the partition $\{S_i\}_{i=1}^k$. For this, we turn to Theorem 2.1 and the results of Section 2.5, which indicate that the number of random walks sampled on the i-th machine scales with $\|\Sigma_{S_i}\|_{\infty,1}$, where Σ_{S_i} is the matrix with rows $\{\sigma_s\}_{s \in S_i}$. Hence, as the random walk stage in our approach runs in parallel across i, its runtime scales with

$$\max_{i \in \{1,\ldots,k\}} \|\Sigma_{S_i}\|_{\infty,1}. \tag{2.19}$$

Our goal is thus to construct the partition $\{S_i\}_{i=1}^k$ so as to minimize (2.19). However, as this is a combinatorial optimization problem, we devise an approximate heuristic. To simplify the discussion of this method, we introduce some notation. For $S' \subset S$, let $\sigma_{S'}$ be s.t. $\sigma_{S'}(v) = \max_{s' \in S'} \sigma_{s'}(v) \; \forall \; v \in V$; note $\|\Sigma_{S'}\|_{\infty,1} = \|\sigma_{S'}\|_1$. For $S' \subset S$ and $s \in S \setminus S'$, let

$$d(s, S') = \sum_{v \in V} \max\left\{\sigma_s(v) - \sigma_{S'}(v), 0\right\}. \tag{2.20}$$

It is easy to derive (2.21), i.e. (2.20) gives the increase in $\|\sigma_{S'}\|_1$ if we add s to S'.

$$d(s, S') = \|\sigma_{S' \cup \{s\}}\|_1 - \|\sigma_{S'}\|_1. \tag{2.21}$$

With this notation in place, we may restate the objective function (2.19) as

$$\max_{i \in \{1,\ldots,k\}} \|\sigma_{S_i}\|_1. \tag{2.22}$$

Our heuristic to approximate the minimizer of (2.22) proceeds as follows. First, we assign one node to each S_i, $i \in \{1, \ldots, k\}$, using an initialization method similar to k-means++ [44]: we choose the i-th of these nodes with probability proportional to its distance from the first $(i-1)$ of them, in hopes of choosing initial nodes with σ_s vectors far apart. Next, we iteratively assign the remaining nodes to some S_j, choosing j such that $d(s, S_j) + \|\sigma_{S_j}\|_1$ is minimal; by (2.21), we thus minimize the increase in the objective function (2.22) incurred by assigning s to some S_j. This heuristic method is formally defined in Algorithm 2.6.

We now empirically compare our approach with the baseline scheme. For this experiment, we set $S = \{\tilde{S}_i\}_{i=1}^k$, where each $\tilde{S}_i$ is a clustered subset of nodes constructed as in Section 2.5 (with $k = 10$, $|\tilde{S}_i| = 100 \; \forall \; i$). This yields a set of sources S that is not highly clustered itself, but that contains k subsets that are densely connected internally and sparsely connected to

Algorithm 2.6: $\{S_i\}_{i=1}^k$ = `Source-Partition` $(\{\sigma_s\}_{s \in S}, k)$

Draw $s \sim S$ uniformly, set $S_1 = \{s\}$, $\sigma_{S_1} = \sigma_s$; set $S_i = \emptyset \ \forall \ i \in \{2, \dots, k\}$
for $i = 2$ **to** k **do**
  Draw $s \sim S$ with probability proportional to $\min_{j \in \{1,\dots,i-1\}} \|\sigma_s - \sigma_{S_j}\|_1$; set $S_i = \{s\}$, $\sigma_{S_i} = \sigma_s$
for $i = k+1$ **to** $|S|$ **do**
  Choose any $s \in S \setminus (\cup_{j=1}^k S_j)$ (any s not yet assigned); compute $d(s, S_j) \ \forall \ j \in \{1, \dots, k\}$
  Let $j^* \in \arg\min_j d(s, S_j) + \|\sigma_{S_j}\|_1$, $\sigma_{S_{j^*}}(v) = \max\{\sigma_{S_{j^*}}(v), \sigma_s(v)\} \ \forall \ v \in V$, $S_{j^*} = S_{j^*} \cup \{s\}$

other subsets. In addition to comparing to the baseline, we also test the performance of an "oracle" scheme, which knows the clustering information of the input set S. More specifically, the oracle scheme proceeds in the same manner as our scheme, except instead of using Algorithm 2.6 to construct the partition $\{S_i\}_{i=1}^k$, it simply sets $S_i = \tilde{S}_i \ \forall \ i \in \{1, 2, \dots, k\}$. Put differently, while the heuristic scheme attempts to learn an assignment of sources to machines for which each machine is assigned a clustered set of sources (in the sense that (2.22) is minimal), the oracle scheme knows such an assignment *a priori*.

Results for this experiment are shown in Fig. 2.10, using the set of real graphs from Section 2.5. Averaging across graphs, the oracle and heuristic methods are roughly 1.8 and 2.2 faster than the baseline scheme, respectively (left). (Here total runtime is computed as maximum walk sampling time across machines for the baseline; sum of maximum forward DP time and maximum walk time for the oracle; and sum of maximum forward DP time, maximum walk time, and time to run Algorithm 2.6 for the heuristic.) Additionally, both methods sample approximately $\frac{1}{4}$ of the random walks sampled by the baseline scheme, across graphs (middle). Finally, the heuristic method typically produces a partition $\{S_i\}_{i=1}^k$ of S with objective function value (2.22) similar to that produced by the oracle method (right). Interestingly, the heuristic outperforms the oracle for several datasets. This suggests that the cluster information known by the oracle does not necessarily produce an optimal assignment of sources to machines; rather, the source clustering quantity $\|\sigma_{S_i}\|_1$ identified in Section 2.4.1.1 may be what truly dictates performance.

Before closing, we offer several remarks. First, while we focused on the scalar estimation scheme from Section 2.4.1.1, the framework extends to the $\sigma_{\max}$ matrix approximation scheme from Section 2.4.2. In particular, using the latter scheme in this setting would also involve construction of a partition so as to minimize (2.22), per Theorem 2.3. For this reason, we expect the performance of this scheme to be similar to Fig. 2.10. Second, we note that using the σ_{avg} matrix approximation scheme in this setting requires a partition that minimizes a

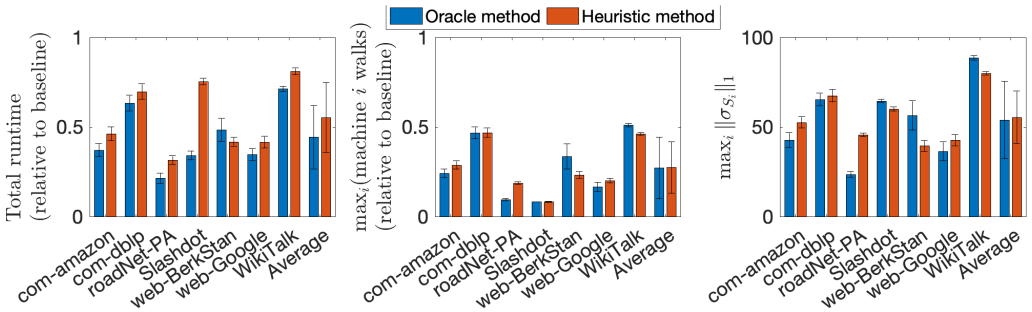


Figure 2.10: In the distributed setting, our heuristic method is typically 1.8 times faster than the baseline, samples $\frac{1}{4}$ of the walks, and produces a low objective function value, with performance similar to an oracle method.

different objective function. In Appendix A.9, we present an algorithm to construct such a partition and empirical results describing performance (in short, our scheme performs similarly to the oracle and noticeably outperforms the baseline, as in Fig. 2.10). Third, we find in practice that our heuristic partitioning schemes naturally balance the number of sources assigned to each machine (see Appendix A.9). Such balance is crucial in the performance of our scheme. This is because we require $\|\sigma_{S_i}\|_1 = O(|S|/k)\ \forall\ i$ to perform as well as the baseline, which may in turn require an extreme degree of clustering if the partition is unbalanced (for example, if $|S_i| = O(|S|)$ for some i). It is worth noting that we also tried to partition $\{\sigma_s\}_{s\in S}$ using k-means++, but this led to highly unbalanced assignments and poor performance. Finally, we note one limitation of our scheme is that, if $|S|, |T| = \Theta(n)$, Algorithm 2.6 essentially partitions the entire graph and thus may be slower than directly estimating PPR. However, recall from Section 2.2 that our focus is $|S|, |T| = o(\sqrt{m}) = o(n)$, so this is not a concern. Indeed, for the Fig. 2.10 experiment, Algorithm 2.6 accounted for only 12% of runtime (averaged across graphs).

2.7 Conclusions and future directions

In this chapter, we analyzed the relationship between PPR estimation complexity and clustering by devising estimation algorithms for many node pairs and showing the complexity of these methods scales with quantities interpretable as clustering measures. To demonstrate the utility of these findings, we considered a distributed setting for which the clustering quantities computed *in situ* could be leveraged to reduce computation time. We believe this setting and the algorithms we designed for it are just one example of how our findings can be used; hence, an avenue for future work would be to further explore applications.

CHAPTER III

Personalized PageRank Dimensionality and Algorithmic Implications[1]

3.1 Introduction

In Chapter II, we devised algorithms to estimate submatrices of the Personalized PageRank (PPR) matrix Π while exploiting clustering in the underlying graph. In this chapter, we take a deeper and more holistic view and consider the structure of Π itself, and subsequent algorithmic implications. Our specific contributions are as follows:

1. In Section 3.4, we prove the dimensionality of Π scales sublinearly in n with high probability, for a certain class of random graphs and for a notion of dimensionality similar to rank (Theorem 3.1). Put differently, we argue that the effective dimension of this matrix is much less than n; this occurs despite the fact that Π is full rank (see Section 1.3.1.3). This class of graphs can be roughly described as the directed configuration model with sparse but heavy-tailed in-degrees. The notion of dimensionality we study is the smallest number of "hub" nodes such that the PPR vectors of all other nodes are close to linear combinations of the hub PPR vectors.
2. In Section 3.5, we show this notion of dimensionality relates closely to the complexity of estimating Π. Specifically, we use our dimensionality result to show that this matrix can be accurately estimated (in terms of the maximum l_1 error across rows) with complexity $O(n^{\bar{c}})$ for some $\bar{c} < 2$ for the same class of graphs (Theorem 3.2). Conceptually, this scheme leverages the low dimension of Π in a manner analogous to low-rank matrix approximation. To the best of our knowledge, our scheme improves upon all existing complexity bounds for this task, the most competitive of which are $O(n^2 \log n)$ in our setting. We also note that maximum l_1 error across rows is a natural accuracy objective, since each row of Π (i.e. each PPR vector) is a distribution over the nodes.

[1]This chapter is adapted from [45].

3. The estimation scheme we analyze is similar to those that were proposed (but not analyzed) by Jeh and Widom in [16] and Berkhin in [17]. Hence, we offer theoretical evidence for the empirical success of these algorithms.
4. While Theorems 3.1 and 3.2 apply to a class of random graphs, we show empirically in Section 3.6 that our dimensionality measure is small relative to n for real graphs. Hence, we argue that the dimension of Π is small more generally. This also suggests that the complexity of estimating Π becomes much smaller when one accounts for the dependencies among its rows that arise from the common underlying graph.
5. Additionally, we believe the class of random graphs considered contains realistic models of real-world networks. As an example, in Section 3.7.3 we provide a model for a graph like Twitter, which contains a few nodes with huge in-degrees – modeling celebrities with millions of Twitter followers – and many nodes with moderate in-degrees – modeling "normal" users with dozens or hundreds of followers. We also discuss various other aspects of our analysis throughout Section 3.7, including a geometric interpretation of our dimensionality result and a connection to Markov chain mixing times.

The chapter is organized as follows. We begin in Sections 3.2 and 3.3 with preliminaries and related work. Sections 3.4-3.7 follow the outline above. We close in Section 3.8.

3.2 Preliminaries

We begin by defining the main ingredients of the chapter. Most notation is standard or defined as needed, but we note the following is often used: for $x \in \mathbb{R}^n$ and $J \subset \{1, 2, \ldots, n\}$, we set $x(J) = \sum_{j \in J} x(j)$, we let $e_J \in \{0, 1\}^n$ satisfy $e_J(j) = 1(j \in J)$ (where $1(\cdot)$ is the indicator function), and we write $e_j = e_{\{j\}}$ for simplicity.

3.2.1 Directed configuration model (DCM)

We consider a random graph model called the directed configuration model (DCM). For the DCM, we are given realizations of random sequences $\mathbf{N}_n = \{N_v\}_{v \in V_n}$ and $\mathbf{D}_n = \{D_v\}_{v \in V_n}$ satisfying $N_v, D_v \in \mathbb{N} \; \forall \; v \in V_n$ and $\sum_{v \in V_n} N_v = \sum_{v \in V_n} D_v \triangleq L_n$ (here $V_n = \{1, 2, \ldots, n\}$).[2] We will refer to $(\mathbf{N}_n, \mathbf{D}_n)$ as the given *degree sequence*. Our goal is to construct a directed graph $G_n = (V_n, E_n)$, such that $v \in V_n$ has in- and out-degree N_v and D_v, respectively. Toward this end, we first assign N_v incoming half-edges and D_v outgoing half-edges to each $v \in V_n$; we call these half-edges *instubs* and *outstubs*, respectively. We then randomly pair half-edges to form edges in a breadth-first search fashion that proceeds as follows:

- Let $s \sim V_n$ uniformly. For each of the D_s outstubs assigned to s, sample an instub

[2]For example, in Section 3.7.3, we let $\mathbf{N}_n \sim f_{\text{in}}$ *i.i.d.* for a given distribution f_{in}; we then realize $\mathbf{D}_n$ conditional on $\sum_{v \in V_n} N_v$ in a manner that guarantees $\sum_{v \in V_n} D_v = \sum_{v \in V_n} N_v$. Alternatively, [18] proposes letting $\mathbf{N}_n \sim f_{\text{in}}$ *i.i.d.*, $\mathbf{D}_n \sim f_{\text{out}}$ *i.i.d.* for some $f_{\text{in}}, f_{\text{out}}$, then to modify $\mathbf{N}_n, \mathbf{D}_n$ to guarantee $\sum_{v \in V_n} D_v = \sum_{v \in V_n} N_v$.

uniformly from the set of all instubs (resampling if the sampled instub has already been paired), and pair the outstub and instub to form an edge out of s.

- Let $A_1 = \{v \in V_n \setminus \{s\} : \text{an outstub of } s \text{ was paired with an instub of } v\}$. For each $v \in A_1$, pair the D_v outstubs of v in the same manner s's outstubs were paired.
- Continue iteratively until all half-edges have been paired. Namely, during the $(m+1)$-th iteration we pair the outstubs of all $v \in A_m$, where A_m is the set of nodes at distance m from s (those $v \in V_n$ for which the shortest path from s to v has length m).

We define this procedure more formally in Appendix B.1.2. For now, the important points to remember are that the initial node s is chosen uniformly at random from V_n, and that, at the end of the m-th iteration, the m-step neighborhood out of s has been constructed. We emphasize the resulting graph will be a multi-graph in general, i.e. it will contain self-loops (edges $v \to v$ for $v \in V_n$) and multi-edges (more than one edge from $v \in V_n$ to $w \in V_n$).[3]

3.2.2 Personalized PageRank (PPR)

In this chapter, we use PPR notation similar to that used in Chapters I-II; however, there are a few key differences we mention here. First, we denote the adjacency matrix of G_n by M, i.e. $M(i,j) \in \{0, \dots, D_i\}$ is the number of directed edges from i to j, for each $i, j \in V_n$. From M, we define a row stochastic matrix P by $P(i,j) = M(i,j)/D_i \ \forall \ i, j \in V_n$. Note P describes the random walk on G_n for which we follow a uniform outgoing edge at each step (i.e. we account for the fact that G_n may be a multi-graph). For $v \in V_n$, we define the (primitive) PPR vector π_v as the stationary distribution of the chain with transition matrix $P_v = (1-\alpha_n)P + \alpha_n 1_n e_v^\mathsf{T}$ (as in Section 3.2.2). We treat each π_v as a row vector and define the PPR matrix Π_n as the matrix with rows $\{\pi_v\}_{v \in V_n}$ (again, as in Section 3.2.2). Thus, we explicitly denote the number of nodes n as a subscript of Π_n in this chapter.

As suggested by the notation above, we allow the restart probability α_n to vary with n in this chapter; in particular, we will let $\alpha_n = \Theta(1/\log n)$. We argue in Section 3.4.2 that this is appropriate when considering the asymptotic behavior of PPR on the DCM. We note a line of work by Boldi *et al.* [46, 47] analyzed the limit of PPR as $\alpha \to 0$ for a fixed graph G; in contrast, we fix a value of α_n for each G_n. Finally, we further motivate our choice of α_n in Section 3.7.4 in terms of the mixing time of the random walk discussed above.

3.2.3 Dimensionality measure

Our main focus is the dimensionality of the PPR matrix Π_n. A standard measure of this dimension is $\text{rank}(\Pi_n)$; however, $\text{rank}(\Pi_n) = n \ \forall \ n \in \mathbb{N}$ (see Section 1.3.1.1), so we will consider a different notion of dimensionality. This notion is motivated by the following obser-

[3] In [18], the authors provide conditions under which a simple graph results with positive probability as $n \to \infty$, but these are stronger than the conditions we require, so we will instead assume G_n is a multi-graph.

vation: the rank of the matrix with rows $\{x_i\}_{i\in I}$ (where I is some finite set) can be bounded by $|X' \cup X''|$, where $X' \subset \{x_i\}_{i\in I}$ and $X'' = \{x_i \notin X' : x_i \text{ is not a linear combination of } X'\}$. We will relax this slightly, by only including in X'' those $x_i \notin X'$ that are not "close" to a linear combination of X'. Specifically, for $\varepsilon > 0$ we define

$$\min_{K_n \subset V_n} \Delta(K_n, \varepsilon) = |K_n| + |\{v \in V_n \setminus K_n : B_v(K_n, \varepsilon) \text{ holds}\}|, \tag{3.1}$$

where $B_v(K_n, \varepsilon)$ is the event

$$\left\{ \inf_{\{\beta_v(k)\}_{k\in K_n} \subset \mathbb{R}} \left\| \pi_v - \left(\sum_{k\in K_n} \beta_v(k)\pi_k + \alpha_n e_v^{\mathsf{T}} \right) \right\|_1 \geq \varepsilon \right\}. \tag{3.2}$$

We offer several remarks on this definition. First, as will be discussed in Section 3.5, (3.1) suggests an algorithm for estimating the PPR matrix: we first estimate π_v for each $v \in K_n$ and $v \notin K_n$ such that $B_v(K_n, \varepsilon)$ holds; we then approximate π_v for other v as a linear combination of the $\{\pi_k\}_{k\in K_n}$ estimates. Under this scheme, (3.1) is the number of PPR vectors estimated directly (i.e., not as linear combinations); we will argue this direct estimation dominates the scheme's complexity, and thus the scheme's complexity scales with (3.1). Second, we note (3.1) differs slightly from the quantity $|X' \cup X''|$ defined in the previous paragraph, since in (3.2) the estimate of π_v is a linear combination *plus* the term $\alpha_n e_v^{\mathsf{T}}$. This latter term is included because it is a known component of π_v, independent of the graph structure encoded by P (by the power iteration (1.1)). Third, we note l_1 distance is a reasonable choice in (3.2) because PPR vectors are distributions over V_n, l_1 distance is twice total variation distance, and total variation is a standard distance for comparing distributions. Finally, we note (3.1) bears some resemblance to low-rank matrix approximation and nonnegative matrix factorization (NMF); we discuss this in Section 3.5.2.

Our dimensionality result (Theorem 3.1) provides conditions on the given degree sequence $(\mathbf{N}_n, \mathbf{D}_n)$ and the choice of K_n under which $\Delta(K_n, \varepsilon)$ scales sublinearly in n; hence, under these conditions, $\min_{K_n \subset V_n} \Delta(K_n, \varepsilon)$ is also sublinear. (Clearly, $|K_n|$ being sublinear is one such condition; proving that $|\{v \in V_n \setminus K_n : B_v(K_n, \varepsilon) \text{ holds}\}|$ is also sublinear is highly nontrivial.) We note that we currently lack a matching lower bound for (3.1).

For analytical tractability moving forward, we will set $K_n = \{v \in V_n : U_v = 0\}$, where $\mathbf{U}_n = \{U_v\}_{v\in V_n}$ is a random length-n binary sequence that may be correlated with the given degree sequence $(\mathbf{N}_n, \mathbf{D}_n)$. Additionally, we will assume the entire tuple $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{U}_n)$ is realized before the graph is constructed. In light of this, we emphasize that $\Delta(K_n, \varepsilon)$ is a random variable that depends on two sources of randomness: the random sequence $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{U}_n)$, and the random graph construction. Towards proving our dimensionality

result, intermediate results will be established with $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{U}_n)$ held fixed, after which expectation with respect to $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{U}_n)$ will be taken. This motivates the following definitions: we let $\mathbb{E}_n[\cdot] = \mathbb{E}[\cdot|\mathbf{N}_n, \mathbf{D}_n, \mathbf{U}_n]$ and $\mathbb{P}_n[\cdot] = \mathbb{P}[\cdot|\mathbf{N}_n, \mathbf{D}_n, \mathbf{U}_n]$ denote expectation and probability with the only source of randomness being the graph construction.

3.3 Related work

Before proceeding, we comment on relationships to prior work. We focus on [16], [17], and [48], the papers most closely related to this chapter. We will also return to discuss more related work – in particular, other PPR estimation schemes – in Section 3.5.2.

In [16], Jeh and Widom propose a scheme for estimating the PPR matrix. The scheme relies crucially on the Hubs Theorem in [16], which states that the PPR vector $\pi_v, v \in V_n \backslash K_n$, can be written as a linear combination of $\{\pi_k\}_{k \in K_n}$ and another vector. The Hubs Theorem is central to our results as well; an alternative formulation appears as Lemma B.1 here. Improving upon [16], Berkhin in [17] proposed a similar algorithm that uses sparse estimates of PPR vectors. We discuss these algorithms in more detail in Section 3.5.2.

Unfortunately, the authors of [16] and [17] present no complexity analysis. Namely, it is unclear how K_n should be chosen and how large it must be to guarantee accurate estimation. This chapter addresses this shortcoming. Specifically, as discussed briefly in the introduction, our dimensionality measure (3.1) relates to the complexity of a similar estimation scheme.

In [48], Chen, Litvak, and Olvera-Cravioto consider the limiting value of π_{σ_n} (the PPR vector with restart distribution σ_n, as in Section 1.2.2) as σ_n weakly converges to a probability distribution σ. Specifically, they show that the PPR value of a uniformly chosen node is given by the solution of a recursive distributional equation (RDE) [49]. They also show (roughly) that PPR values follow a power law when in-degrees follow a power law, establishing the "power law hypothesis" that had long been observed empirically. Similar results were proven for other graph families in [50]. On the other hand, [48] was preceded by [51], where the power law hypothesis was established for global PageRank; further back, the hypothesis was studied under more restrictive assumptions in [52, 53, 54].

While [51, 48, 50, 52, 53, 54] share a goal of understanding the power law behavior of PPR on random graphs, our goal is to instead understand the dimensionality of Π_n. As alluded to above, dimensionality carries with it algorithmic implications, so the current chapter is perhaps more useful from a practical perspective when compared to this body of work. However, the analytical approaches of these works will be useful to us. Specifically, we will use a modified version of Lemma 5.4 from [48]; see Appendix B.1.3.

In short, this chapter combines the strengths of [16, 17], which are entirely algorithmic, and [48], which is entirely analytical. Specifically, we leverage certain aspects of the analysis

from [48] to obtain guarantees on an algorithm similar to those proposed in [16, 17].

More broadly, other work studying PPR on random graphs includes [55], where it is shown that π_{σ_n} can be well-approximated as a convex combination of σ_n and the degree distribution for certain random graphs. The DCM was proposed and analyzed in [18, 56] as an extension of the (undirected) configuration model, the development of which began in [57, 58, 59]. The configuration model (and variants) have been studied in detail; for example, [60] considers graph diameter in this model, while [61] studies the emergence of a giant component.

3.4 Dimensionality result

We next turn to our dimensionality result. We define our assumptions and our choice of α_n in Sections 3.4.1 and 3.4.2, respectively, and then state the result in Section 3.4.3.

3.4.1 Assumptions

Our dimensionality result is a consequence of a key lemma, the proof of which requires Assumption 3.1. This assumption states that certain moments of $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{U}_n)$ exist with high probability, and furthermore, converge to limits at a certain rate.

Assumption 3.1. We have $\mathbb{P}[\Omega_n^C] = O(n^{-\delta})$ for some $\delta \in (0,1)$, where $\Omega_n = \cap_{i=1}^6 \Omega_{n,i}$ and for some constants $\gamma, p \in (0,1)$ and $\eta_i, \zeta^*, \lambda^* \in (0,\infty)$, all independent of n,

$$\Omega_{n,1} = \left\{ \left| \frac{\sum_{h=1}^n N_h}{n} - \eta_1 \right| \leq n^{-\gamma} \right\}, \qquad \Omega_{n,4} = \left\{ \left| \frac{\sum_{h=1}^n U_h D_h}{\sum_{h=1}^n U_h} - \zeta^* \right| \leq n^{-\gamma} \right\},$$
$$\Omega_{n,2} = \left\{ \left| \frac{\sum_{h=1}^n N_h D_h}{n} - \eta_2 \right| \leq n^{-\gamma} \right\}, \qquad \Omega_{n,5} = \left\{ \left| \frac{\sum_{h=1}^n U_h N_h}{\sum_{h=1}^n U_h} - \lambda^* \right| \leq n^{-\gamma} \right\},$$
$$\Omega_{n,3} = \left\{ \left| \frac{\sum_{h=1}^n U_h N_h^2}{n} - \eta_3 \right| \leq n^{-\gamma} \right\}, \qquad \Omega_{n,6} = \left\{ \left| \frac{\sum_{h=1}^n U_h N_h}{\sum_{h=1}^n N_h} - p \right| \leq n^{-\gamma} \right\}.$$

We also have $\zeta \triangleq \eta_2/\eta_1 > 1$ and define $\lambda = \eta_3/\eta_1$.

We note that the constants ζ and p appearing in Assumption 3.1 also appear in our dimensionality result, and both have simple interpretations: if v_n satisfies $\mathbb{P}[v_n = v] \propto N_v \ \forall \ v \in V_n, n \in \mathbb{N}$, then $\lim_{n\to\infty} \mathbb{E}[D_{v_n} | \Omega_n] = \zeta$ and $\lim_{n\to\infty} \mathbb{E}[U_{v_n} | \Omega_n] = p$, i.e. ζ and p give the limiting expected out-degree and the limiting probability of belonging to $V_n \setminus K_n$, respectively, for a node sampled with probability proportional to in-degree. (The other constants in Assumption 3.1 will not appear in our dimensionality result but have similar interpretations.) We also remark that $\zeta > 1$ in Assumption 3.1 is not necessary to establish our results but, given this interpretation, is the more interesting case; this also simplifies the statements and proofs of certain results (which otherwise would have to address the cases

$\zeta > 1$, $\zeta = 1$, and $\zeta < 1$ separately).

Our dimensionality result requires Assumption 3.2, which strengthens Assumption 3.1 by requiring $|K_n|$ to be sublinear (an obvious requirement for sublinearity of (3.1)).

Assumption 3.2. $\exists\ \kappa \in (0,1)$ independent of n s.t. $\mathbb{E}[|K_n|] = \mathbb{E}[\sum_{h=1}^{n}(1-U_h)] = O(n^{\kappa})$ and Assumption 3.1 holds.

While Assumption 3.2 may appear limiting, we provide an example $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{U}_n)$ in Section 3.7.3 that satisfies it, and that we believe is a reasonable model for certain graphs arising in the motivating applications of Section 1.2.3 (namely, Twitter). Additionally, we argue in Section 3.7.2 that several events $\Omega_{n,i}$ appearing in Assumption 3.1 are essentially implied by others, and are therefore not restrictive. Finally, we believe the most important condition for sublinearity of $\Delta(K_n, \varepsilon)$ is that K_n contains a vanishing fraction of nodes but a non-vanishing fraction of edges (i.e. $\mathbb{E}[|K_n|] = O(n^{\kappa})$ and $\Omega_{n,6}$ holds in Assumption 3.1). We discuss this further in Section 3.7.2 and provide empirical evidence that this holds when $\{N_v\}_{v \in V_n}$ follow a power law, a common model for real-world graphs.

3.4.2 Choice of restart probability

As mentioned in Section 3.2.2, we take $\alpha_n = \Theta(1/\log n)$ in this chapter. Having defined Assumption 3.1, we choose a specific value of α_n. This choice is motivated by the following proposition, which states that s's PPR concentrates in a small neighborhood surrounding s, and bounds the size of this neighborhood, for two choices of α_n.

Proposition 3.1. Let $\tau \in (0,1)$ and $\rho > 1$ be constants, and let $s \sim V_n \setminus K_n$ uniformly. For $l \in \mathbb{N}$, let $V_{n,s}(l)$ denote the l-step neighborhood out of s. Then the following hold:

- If $\alpha_n = \rho \log(1/\tau) \log(\zeta)/\log(n) = \Theta(1/\log n)$ and $l = \lceil \log(1/\tau)/\alpha_n \rceil$,

$$\liminf_{n\to\infty} \pi_s\left(V_{n,s}(l)\right) \geq 1-\tau \ a.s., \quad \mathbb{E}\left[|V_{n,s}(l)||\Omega_n\right] = O\left(n^{1/\rho}\right).$$

- If $\alpha_n = \alpha$ is a constant and $l = \lceil \log(\tau)/\log(1-\alpha) \rceil$,

$$\liminf_{n\to\infty} \pi_s\left(V_{n,s}(l)\right) \geq 1-\tau \ a.s., \quad \mathbb{E}\left[|V_{n,s}(l)||\Omega_n\right] = O(1).$$

Proof. See Appendix B.5. □

Loosely speaking, Proposition 3.1 states that, for both choices of α_n, all but τ of s's PPR concentrates on a small neighborhood surrounding s, for any $\tau > 0$. The difference is the size of this neighborhood: when $\alpha_n = \Theta(1/\log n)$, the neighborhood grows with the graph; when α_n is constant, the neighborhood has constant size. From the PPR interpretation of Section 1.2.2, this suggests that the number of nodes that are "similar" to s grows in the former

case but remains fixed in the latter case. We believe the former case is more appropriate. Additionally, the growth of this similar set of nodes remains sublinear in n in the former case; intuitively, this says that a vanishing fraction of all nodes are important to s, i.e. the PPR vector remains "personalized" to s. Later, we will further motivate this choice in terms of the mixing time of the simple random walk on G_n (see Section 3.7.4). We also remark that, since PPR concentrates on a small neighborhood for this choice of α_n, PPR vectors can be well-approximated by sparse estimates (with the sparsity precisely controlled by τ and ρ), which has implications in terms of both time and space complexity for algorithms we discuss in Section 3.5.1. We reiterate that for the remainder of the chapter, we set

$$\alpha_n = \frac{\rho \log(1/\tau) \log(\zeta)}{\log n} = \Theta\left(\frac{1}{\log n}\right). \tag{3.3}$$

3.4.3 Dimensionality result

Before presenting our dimensionality result, we state a tail bound for the event $B_s(K_n, \varepsilon)$ (recall this event, defined in (3.2), states that π_s is more than ε from a linear combination of $\{\pi_k\}_{k \in K_n}$). Our dimensionality result will follow almost immediately from this lemma. The bound is $n^{-\min\{c_1, c_2\varepsilon^2\}}$ for constants c_i depending only on the degree sequence and the choice of α_n; hence, ε only affects the bound when it is sufficiently small.

Lemma 3.1. Given Assumption 3.1, for $s \sim V_n$ uniformly and $\varepsilon > 0$ independent of n,

$$\mathbb{P}\left[B_s(K_n, \varepsilon) | U_s = 1\right] = O\left(n^{-c(\varepsilon)}\right),$$

where, with δ, p, ζ from Assumption 3.1, and with ρ, τ from (3.3),

$$c(\varepsilon) \triangleq \min\left\{\delta, \frac{\log(1/p)}{2\log(\zeta/p)}, \frac{((1-p)\varepsilon)^2}{2\rho \log(1/\tau) \log \zeta}\right\} > 0.$$

The proof of Lemma 3.1 is lengthy and occupies Appendices B.1 and B.2. For now, we note that the proof broadly requires four steps:

1. Show that, for a certain $\{\beta_s(k)\}_{k \in K_n}$, the error term $\|\pi_s - (\alpha_n e_s^{\mathsf{T}} + \sum_{k \in K_n} \beta_s(k)\pi_k)\|_1$ in $B_s(K_n, \varepsilon)$ can be bounded by only examining the m-step neighborhood out of s. (Here the choice of $\{\beta_s(k)\}_{k \in K_n}$ arises from an alternate form of the Hubs Theorem from [16], which requires a new proof; the error bound is new, to the best of our knowledge.)
2. Argue that, conditioned on certain events not occurring during the first m steps of the graph construction, this error bound follows the same distribution as a quantity defined in terms of the first m generations of a branching process. (Here we essentially argue that, before these events occur, a bijection exists between the subgraph that

determines the error bound and the tree resulting from the branching process.)

3. Bound the probability of these events occurring during the first m iterations. (Here we use a modification of Lemma 5.4 from [48]. Our modification weakens the assumptions of [48], allowing us to apply it to a wider class of degree sequences; see Section 3.7.3.)
4. Bound the probability of $B_s(K_n, \varepsilon)$, conditioned on these events not occurring, by analyzing the branching process quantity. (Here our analysis leverages the fact that the branching process quantity has a martingale-like structure.)

Before proceeding, we pause to state the choice of $\{\beta_s(k)\}_{k\in K_n}$ from Step 1, which will be used in Section 3.5. First, for any realization of the DCM and for $v \in V_n \setminus K_n$, we define

$$\tilde{P}(i,j) = U_i P(i,j), \quad \tilde{P}_v = (1-\alpha_n)\tilde{P} + (\alpha_n e_{V_n \setminus K_n} + e_{K_n})e_v^{\mathsf{T}}, . \tag{3.4}$$

Note $\tilde{P}_v$ corresponds to a Markov chain for which the random surfer from Section 1.2.2 restarts at v with probability 1 upon reaching K_n (instead of α_n). Letting $\tilde{\pi}_v$ denote the stationary distribution of this chain, one can show (see Appendix B.1.1)

$$\pi_v(w) = \frac{\alpha_n U_w \tilde{\pi}_v(w) + \sum_{k\in K_n} \tilde{\pi}_v(k)\pi_k(w)}{\alpha_n + (1-\alpha_n)\tilde{\pi}_v(K_n)} \ \forall \ w \in V_n. \tag{3.5}$$

With (3.5) in mind, we define

$$\beta_v(k) = \frac{\tilde{\pi}_v(k)}{\alpha_n + (1-\alpha_n)\tilde{\pi}_v(K_n)} \ \forall \ k \in K_n, \tag{3.6}$$

and we take $\{\beta_s(k)\}_{k\in K_n}$ as in (3.6) in Step 1 above.

We now turn to the dimensionality result, Theorem 3.1. Together with Assumption 3.2, it essentially states the following: when certain moments of the degree sequence exist, and when a sublinear number of nodes contains a constant fraction of instubs, the dimension of the PPR matrix scales sublinearly in n (with high probability).

Theorem 3.1. Given Assumption 3.2, for any $\varepsilon > 0$ independent of n,

$$\mathbb{E}[\Delta(K_n, \varepsilon)] = O\left(n^{\max\{\kappa, 1-c(\varepsilon)\}}\right),$$

where $c(\varepsilon)$ is from Lemma 3.1. Hence, $\forall\ \bar{c} \in (\max\{\kappa, 1-c(\varepsilon)\}, 1), C > 0$ independent of n,

$$\mathbb{P}\Big[\Delta(K_n, \varepsilon) \geq Cn^{\bar{c}}\Big] = O\left(n^{\max\{\kappa, 1-c(\varepsilon)\}-\bar{c}}\right) \xrightarrow[n\to\infty]{} 0.$$

Proof. See Appendix B.3. □

3.5 Algorithmic implications

Having stated Theorem 3.1, we next consider its algorithmic consequences. Broadly speaking, the main consequence is that only $\Delta(K_n, \varepsilon)$ PPR vectors – those corresponding to K_n and $\{v \in V_n \setminus K_n : B_v(K_n, \varepsilon) \text{ holds}\}$ – need be computed, after which the others can be estimated as linear combinations of $\{\pi_k\}_{k \in K_n}$ using the weights $\{\tilde{\pi}_v(k)\}_{k \in K_n}$. Because $\Delta(K_n, \varepsilon)$ scales sublinearly per Theorem 3.1, this may yield an order reduction when compared to the naive scheme of computing all n PPR vectors. However, computing $\Delta(K_n, \varepsilon)$ vectors and $n - \Delta(K_n, \varepsilon)$ sets of weights $\{\tilde{\pi}_v(k)\}_{k \in K_n}$ remains too costly, so each of these quantities will be estimated. This introduces a nontrivial challenge: the errors incurred by estimating $\{\pi_k\}_{k \in K_n}$ and $\{\tilde{\pi}_v(k)\}_{k \in K_n}$ will propagate through to the estimate of π_v, potentially rendering it highly inaccurate. In this section, we devise an algorithm that overcomes this challenge. We then discuss compare this algorithm to existing PPR estimators.

3.5.1 Algorithm to estimate the PPR matrix

At a high level, our algorithm proceeds as follows. First, estimate $\{\pi_k\}_{k \in K_n}$ as $\{\hat{\pi}_k\}_{k \in K_n}$. Next, for $v \in V_n \setminus K_n$, estimate π_v as

$$\alpha_n e_v^\mathsf{T} + \frac{\sum_{k \in K_n} \hat{\tilde{\pi}}_v(k) \hat{\pi}_k}{\alpha_n + (1 - \alpha_n) \hat{\tilde{\pi}}_v(K_n)}, \tag{3.7}$$

where $\hat{\tilde{\pi}}_v(k)$ is an estimate of $\tilde{\pi}_v(k)$. The basic idea behind this scheme is that, by (3.5), the estimate shown in (3.7) may be close to π_v. Throughout this section, we make this idea rigorous, developing an algorithm based on this idea and using Theorem 3.1 to analyze it.

The first step of our scheme is to estimate $\{\pi_k\}_{k \in K_n}$. Here we use a modified version of the `Approx-PageRank` algorithm [7] from Section 2.3. Our modification accounts for the fact that the DCM may be a multi-graph; we define it in Algorithm 3.2 and include a modified analysis in Appendix B.4. Specifically, in Appendix B.4 we show that for any realization of G_n and any $v \in V_n$, `Approx-PageRank`(v, ε_1) has complexity $O(n \log n / \varepsilon_1)$, assuming $L_n = O(n)$ (as in Assumption 3.1) and $\alpha_n = \Theta(1/\log n)$. Hence, running `Approx-PageRank`$(k, \varepsilon_1)$ $\forall\, k \in K_n$ will yield estimates of $\{\pi_k\}_{k \in K_n}$ with error guarantees in the l_1 norm.

We next consider estimation of $\{\tilde{\pi}_v(k)\}_{v \in V_n \setminus K_n, k \in K_n}$; here we desire an l_∞ error guarantee that, paired with the l_1 guarantee on $\hat{\pi}_k$, will yield an l_1 guarantee on (3.7). A natural algorithm to use is the `Approx-Contributions` algorithm [32] from Section 2.3. However, since we seek estimates of $\{\tilde{\pi}_v(k)\}_{v \in V_n \setminus K_n}$, not $\{\pi_v(k)\}_{v \in V_n \setminus K_n}$, the existing algorithm does not directly apply. For this reason, we consider a modified version of `Approx-Contributions`.

This is based on our analysis in Appendix B.2.2, which shows (see (B.34))

$$\tilde{\pi}_v(k) = \frac{\alpha_n \mu_v(k)}{1 - (1-\alpha_n)\mu_v(K_n)} \; \forall \; v \in V_n \setminus K_n, k \in K_n,$$

where $\mu_v(k)$ is the k-th element of $\mu_v = e_v^{\mathsf{T}}(I - (1-\alpha_n)\tilde{P})^{-1}$ (here $\tilde{P}$, defined in (3.4), is the row-normalized adjacency matrix with rows corresponding to K_n set to zero). Note that μ_v has nearly the same form as π_v, by (1.1); in particular, μ_v is a scaled PPR vector defined on a modified graph. Hence, we will use a modified version of `Approx-Contributions` to estimate $\{\mu_v(k)\}_{v \in V_n}$ for each $k \in K_n$, after which we may estimate $\{\tilde{\pi}_v(k)\}_{v \in V_n \setminus K_n}$ as

$$\hat{\tilde{\pi}}_v(k) = \frac{\alpha_n \hat{\mu}_v(k)}{1 - (1-\alpha_n)\hat{\mu}_v(K_n)} \; \forall \; v \in V_n \setminus K_n, k \in K_n.$$

This modification is necessary to account for the different scaling between π_v and μ_v, as well as the fact that our DCM may be a multi-graph. With this in mind, we define a modified version of `Approx-Contributions` in Algorithm 3.3; we include an analysis in Appendix B.4, based on [32], which includes our desired bound on $|\tilde{\pi}_v(k) - \hat{\tilde{\pi}}_v(k)|$.

After computing $\{\hat{\pi}_k\}_{k \in K_n}$ and $\{\hat{\tilde{\pi}}_v(k)\}_{v \in V_n \setminus K_n, k \in K_n}$ via Algorithms 3.2 and 3.3, we could immediately compute (3.7) and return this as our estimate of π_v for $v \in V_n \setminus K_n$. There are two drawbacks to this approach, both of which our algorithm will address. The first drawback is that, while Lemma 3.1 guarantees π_v is close to

$$\alpha_n e_v^{\mathsf{T}} + \frac{\sum_{k \in K_n} \tilde{\pi}_v(k)\pi_k}{\alpha_n + (1-\alpha_n)\tilde{\pi}_v(K_n)} \tag{3.8}$$

for most v (namely, all but a sublinear number), it could be far from (3.8) for some v; hence, the estimate given by (3.7) may be inaccurate for some v as well. The second drawback is that computing (3.7) for every $v \in V_n \setminus K_n, k \in K_n$ requires matrix multiplication that has complexity $O(n^{2+\kappa})$ when $|K_n| = O(n^{\kappa})$; as we discuss in Section 3.5.2, existing methods can estimate $\{\pi_v\}_{v \in V_n}$ in our setting with complexity $O(n^2 \log n)$ and with the same accuracy guarantee we will provide. Hence, we next address each of these drawbacks.

First, to address the accuracy concern, we show in Appendix B.4 that $\hat{\tilde{\pi}}_v(K_n)$ can be used at runtime to determine whether (3.7) will be sufficiently accurate, without actually computing (3.7) or knowing π_v itself. Specifically, we show that if $\hat{\tilde{\pi}}_v(K_n)$ exceeds

$$g_n(\varepsilon_1) = \frac{\alpha_n(1 - (\varepsilon_1 + \alpha_n))}{\varepsilon_1 + \alpha_n(2 - (\varepsilon_1 + \alpha_n))},$$

then (3.7) will be close to π_v in l_1. Note $\hat{\tilde{\pi}}_v(K_n) \geq g_n(\varepsilon_1)$ intuitively states that K_n is "close"

to v in the graph; hence, (3.7) is a good estimate of π_v whenever K_n is close to v.

Second, to address the matrix multiplication concern, we will "sparsify" the matrix $\{\hat{\pi}_k\}_{k\in K_n}$, i.e. set certain elements of this matrix to zero. We will do so in a manner that ensures (1) enough elements are set to zero to guarantee the resulting multiplication has complexity $o(n^2)$, and (2) not enough elements are set to zero to significantly alter the accuracy of the resulting estimates. In particular, we will set $\hat{\pi}_k(u)$ to zero whenever $u \notin V_{n,k}(l)$, where l is an input to our algorithm and $V_{n,k}(l)$ is the l-step neighborhood out of k (as in Proposition 3.1). Using an argument similar to the proof of Proposition 3.1, we will show that $|V_{n,k}(l)|$ scales sublinearly in n (which will address point (1) above) and that $\hat{\pi}_k(u)$ is small whenever $u \notin V_{n,k}(l)$ (which will address point (2) above).

Algorithm 3.1: $\{\hat{\pi}_v\}_{v\in V_n} = \texttt{Estimate-All-PPR}(\varepsilon_1, \varepsilon_2, l)$

for $k \in K_n$ **do**
  $\hat{\pi}_k = \{\hat{\pi}_k(u)\}_{u\in V_n} = \texttt{Approx-PageRank}(k, \varepsilon_1)$ (Algorithm 3.2)
  $\hat{\pi}_k^l(u) = \hat{\pi}_k(u)1(u \in V_{n,k}(l)) \ \forall \ u \in V_n$
  $\{\hat{\mu}_u(k)\}_{u\in V_n} = \texttt{Approx-Contributions}(k, \varepsilon_2)$ (Algorithm 3.3)
for $v \in V_n \setminus K_n$ **do**
  $\hat{\hat{\pi}}_v(k) = \alpha_n \hat{\mu}_v(k)/(1 - (1-\alpha_n)\hat{\mu}_v(K_n)) \ \forall \ k \in K_n$
  if $\hat{\hat{\pi}}_v(K_n) < g_n(\varepsilon_1)$ **then** $\hat{\pi}_v = \{\hat{\pi}_v(u)\}_{u\in V_n} = \texttt{Approx-PageRank}(v, \varepsilon_1)$
  else $\hat{\pi}_v = \alpha_n e_v^{\mathsf{T}} + \sum_{k\in K_n} \hat{\hat{\pi}}_v(k)\hat{\pi}_k^l/(\alpha_n + (1-\alpha_n)\hat{\hat{\pi}}_v(K_n))$

Algorithm 3.2: $\hat{\pi}_v = \{\hat{\pi}_v(u)\}_{u\in V_n} = \texttt{Approx-PageRank}(v, \varepsilon_1)$

$\hat{\pi}_v(u) = 0, r_v(u) = 1(u = v) \ \forall \ u \in V_n$
while $\max_{u\in V_n} r_v(u)/D_u > \varepsilon_1/L_n$ **do**
  $v^* \in \arg\max_{u\in V_n} r_v(u)/D_u$
  $r_v(u) \leftarrow r_v(u) + (1-\alpha_n) r_v(v^*) P(v^*, u) \ \forall \ u \neq v^*$
  $\hat{\pi}_v(v^*) \leftarrow \hat{\pi}_v(v^*) + \alpha_n r_v(v^*)$
  $r_v(v^*) \leftarrow (1-\alpha_n) r_v(v^*) P(v^*, v^*)$

Algorithm 3.3: $\{\hat{\mu}_u(v)\}_{u\in V_n} = \texttt{Approx-Contributions}(v, \varepsilon_2)$

$\hat{\mu}_u(v) = 0, r_v(u) = 1(u = v) \ \forall \ u \in V_n$
while $\max_{u\in V_n} r_v(u) > \varepsilon_2$ **do**
  $v^* \in \arg\max_{u\in V_n} r_v(u)$
  $r_v(u) \leftarrow r_v(u) + (1-\alpha_n) r_v(v^*) \tilde{P}(u, v^*) \ \forall \ u \neq v^*$
  $\hat{\mu}_{v^*}(v) \leftarrow \hat{\mu}_{v^*}(v) + r_v(v^*)$
  $r_v(v^*) \leftarrow (1-\alpha_n) r_v(v^*) \tilde{P}(v^*, v^*)$

Having described each step of our scheme, we provide a formal definition in Algorithm 3.1. Theorem 3.2 provides two guarantees for Algorithm 3.1, stated informally as follows:

1. For certain choices of τ in the definition of α_n, each estimate $\hat{\pi}_v$ is accurate in the l_1 norm *a.s.* In fact, this guarantee holds for any underlying graph G_n.
2. For certain choices of ρ in the definition of α_n, the complexity is $o(n^2)$ with high probability, assuming Assumption 3.2 holds and each $k \in K_n$ has $O(1)$ out-degree. (Note this strengthens the previous assumption of $O(1)$ *average* out-degree.)

We reemphasize that Algorithm 3.1 generates estimates accurate in the l_1 norm for any graph G_n. We also note this is a natural objective, since bounding l_1 distance is equivalent to bounding total variation distance, total variation is a standard distance for comparing distributions, and PPR vectors are distributions over V_n. Additionally, while the complexity guarantee pertains to a class of graphs, we suspect the algorithm will perform well for a wider class of graphs. For example, we believe that graphs with in-degrees following a power law satisfy the most crucial of our assumptions (see Section 3.7.2). Hence, while the entirety of Theorem 3.2 applies to a class of graphs, we believe Algorithm 3.1 is of wider value.

Theorem 3.2. Let $\varepsilon \in (0,1)$ be a constant, and set $\varepsilon_1 = \varepsilon/4$, $\varepsilon_2 = \alpha_n^2 g_n(\varepsilon/4)/(2|K_n|)$, and $l = \lceil \log(1/\tau)/\alpha_n \rceil$ in Algorithm 3.1. Then the following hold:

- (Accuracy) Assume $\tau \leq \varepsilon/4$ in the definition of α_n. Then for an arbitrary graph G_n,

$$\|\hat{\pi}_v - \pi_v\|_1 < \varepsilon \ \forall \ v \in V_n \ a.s.$$

- (Complexity) Assume G_n is the DCM, Assumption 3.2 holds, $\rho > \frac{1}{1-\kappa}$ in the definition of α_n (with κ from Assumption 3.2), and $\exists$ constants $\delta' > \kappa$ and $D_{\max} > 0$ s.t.

$$\mathbb{P}\left[\max_{k \in K_n} D_k > D_{\max} \middle| \Omega_n\right] = O\left(n^{-\delta'}\right).$$

  Then, letting $C_{Alg3.1}$ denote the complexity of Algorithm 3.1,

$$\mathbb{E}\left[C_{Alg3.1} | \Omega_n\right] = O\left(\max\left\{\mathbb{E}[\Delta(K_n, \varepsilon/14)] n (\log n)^3, n^{\max\{1+\kappa+1/\rho, 2+\kappa-\delta'\}}\right\}\right).$$

  Consequently, $\exists\ \bar{c} < 2$ such that

$$\lim_{n\to\infty} \mathbb{P}\left[C_{Alg3.1} \geq n^{\bar{c}} | \Omega_n\right] = 0.$$

Proof. See Appendix B.4. □

We note that the $\mathbb{E}[\Delta(K_n, \varepsilon/14)n(\log n)^3]$ term[4] in the complexity guarantee accounts for all but Line 8 of Algorithm 3.1. Roughly, the proof shows that `Approx-PageRank` need only be run $\Delta(K_n, \varepsilon/14)$ times, once for each node in K_n or $\{v \in V_n \setminus K_n : B_v(K_n, \varepsilon/14) \text{ holds}\}$; for other nodes v, π_v can be well-approximated as a linear combination of $\{\hat{\pi}_k\}_{k \in K_n}$. The proof also shows that each run of `Approx-PageRank` has complexity linear in n; hence, if instead $\Delta(K_n, \varepsilon/14)$ scaled linearly in n, Algorithm 3.1 would have n^2 complexity. Because of this, Theorem 3.2 should be viewed as a consequence of Theorem 3.1. On the other hand, the $n^{\max\{1+\kappa+1/\rho, 2+\kappa-\delta'\}}$ term in the complexity guarantee accounts for the matrix multiplication in Line 8 and requires the additional assumptions stated in Theorem 3.2.

One issue we have not addressed is how to optimally choose K_n; our analysis simply says *if* an appropriate K_n exists, *then* the algorithm has subquadratic complexity. We believe choosing nodes of highest in-degree as K_n is a good choice; see Section 3.6 for empirical results and Section 3.7.1 for some theoretical evidence. However, an important (though, we believe, a very difficult) question for future work is as follows: given $\varepsilon > 0$, how can one optimally choose $K_n = K_n(\varepsilon)$ to ensure ε-accuracy while minimizing complexity in Algorithm 3.1?

Finally, we note Algorithm 3.1 can be easily modified to obtain a variant that uses precomputation and that proceeds as follows:

- Offline stage: Run Lines 1-4 of Algorithm 3.1; store $\{\hat{\pi}_k^l\}_{k \in K_n}$ and $\{\hat{\mu}_k(k)\}_{u \in V_n, k \in K_n}$.
- Online stage: When an estimate of $\pi_k, k \in K_n$ is needed, return $\hat{\pi}_k^l$; when an estimate of $\pi_v, v \notin K_n$ is needed, run Lines 6-8 of Algorithm 3.1.

In Appendix B.4.3, we show that, under the assumptions of Theorem 3.2, the offline stage requires $O(n^{1+\kappa})$ storage, the estimate returned during the online stage has l_1 error bounded by ε, and for $s \sim V_n$ uniformly, the online stage has complexity

$$O\left(\max\left\{\Delta(K_n, \varepsilon/14)\log n, n^{\kappa+\max\{1/\rho, 1-\delta'\}}\right\}\right).$$

Note that this storage is subquadratic, strictly better than the n^2 storage required the store the PPR matrix itself. Additionally, with $\kappa < \max\{\delta', 1 - 1/\rho\}$ per the assumptions of Theorem 3.2, the online stage has sublinear complexity, strictly better than the existing approach of running `Approx-PageRank` for $s \sim V_n$ uniformly online ($O(n \log n)$ complexity).

3.5.2 Comparison to other algorithms

In the previous section, we showed Algorithm 3.1 estimates $\{\pi_v\}_{v \in V_n}$ with constant error in the l_1 norm for each vector and has complexity $o(n^2)$ on the DCM (under appropriate assumptions). To the best of our knowledge, this complexity bound is strictly

[4] $\Delta(K_n, \varepsilon/14)$ appears because the `if` statement in Algorithm 3.1 relies on an estimate of $\tilde{\pi}_v(K_n)$; if $\tilde{\pi}_v(K_n)$ were known, $\Delta(K_n, \varepsilon)$, a smaller quantity, would instead appear. The factor 14 has no particular significance.

better than any in the literature. In fact, the best existing algorithm is to simply run `Approx-PageRank`$(v, \varepsilon)\ \forall\ v \in V_n$, which, by Lemma B.11 in Appendix B.4, guarantees constant l_1 error and has complexity $O(n^2 \log n)$ when $L_n = O(n), \alpha_n = \Theta(1/\log n)$ (i.e. in the setting of Theorem 3.2). Alternatively, the original version of `Approx-Contributions` from [32] guarantees l_∞ error bounded by ε_n with complexity $O(L_n/(n\alpha_n\varepsilon_n))$ for a uniformly random node (see Theorem 2 in [62]). Hence, running this algorithm for all nodes, setting $\varepsilon_n = \varepsilon/n$ to obtain constant l_1 error, and taking $L_n = O(n), \alpha_n = \Theta(1/\log n)$ as in our setting, this complexity is also $O(n^2 \log n)$. Similar complexity bounds are provided in [63] for algorithms based on deterministic rounding and randomized sketching; namely, [63] shows all PPR vectors can be estimated accurately in l_∞ with $O(n \log n/\varepsilon_n)$ complexity, which again is $O(n^2 \log n)$ if we desire an l_1 guarantee. We do concede that these $O(n^2 \log n)$ bounds only require $L_n = O(n)$, while ours pertains to a class of random graphs; however, we believe this class contains reasonable models for many graphs of interest (see Section 3.7.3).

We also note we conditioned on Ω_n for the complexity guarantee of Theorem 3.2 because, if instead Ω_n^C holds, we could have $\Delta(K_n, \varepsilon/14) = O(n), L_n = O(n^2)$, in which case Algorithm 3.1 will have complexity $O(n^3 \log n)$, the same as the existing methods. However, if we assume $L_n = O(n)$ with probability 1, we can write

$$\mathbb{E}\left[C_{Alg3.1}\right] \leq \mathbb{E}\left[C_{Alg3.1}|\Omega_n\right] + O(n^2 \log n)\mathbb{P}\left[\Omega_n^C\right] = o(n^2),$$

where the first term is $o(n^2)$ by Theorem 3.2 and the second is $o(n^2)$ since $\mathbb{P}[\Omega_n^C] = O(n^{-\delta})$ by Assumption 3.1. Thus, when $L_n = O(n)$ with probability 1, Algorithm 3.1 is subquadratic and thus strictly better than existing methods *without* conditioning on Ω_n.

Another line of work worth mentioning includes [64, 65, 66, 67]. For example, [64] provides algorithms to estimate the solution of $Ax = b$ in nearly-linear time, which in principle could be run separately across nodes to estimate Π in nearly-quadratic time. However, it is unclear how to exploit dependencies across rows of Π in this scheme, and thus unclear if this can be improved to subquadratic time like our algorithm. Also, [64] bounds $\|\hat{x} - x\|_A$, where $\hat{x}$ is the estimated solution and $\|y\|_A = \sqrt{y^\intercal A y}$ for a vector y; this accuracy guarantee is somewhat unnatural for PPR vectors. In a related line of work, [68, 69] devise algorithms to estimate graph-related primitives including PPR, stationary distributions, and commute times in nearly-linear time; for PPR, l_2 accuracy guarantees are provided.

As mentioned in Sections 3.1 and 3.3, our algorithm is similar to those proposed in [16, 17]. Jeh and Widom's scheme from [16] follows the description from the beginning of Section 3.5.1: first, estimate the hub PPR vectors $\{\pi_k\}_{k \in K_n}$ and the weights $\{\tilde{\pi}_v(k)\}_{v \in V_n \setminus K_n, k \in K_n}$; second, for $v \notin K_n$, approximate π_v as a linear combination via (3.7). The estimation in

the first step uses dynamic programming (DP) algorithms similar in spirit to the methods `Approx-PageRank` and `Approx-Contributions` that we use. Berkhin in [17] similarly uses hub PPR vectors as a basis for estimating other PPR vectors, with dynamic programming algorithms again used for the primitive PPR estimation. The key difference is that Berkhin's scheme involves *bookmark-coloring vectors* (BCVs), which are essentially sparse estimates of PPR vectors; it is shown empirically in [17] that controlling the sparsity of these BCVs reduces the runtime of Jeh and Widom's scheme. However, in the case of both papers, no complexity analysis is provided for the primitive DP algorithms. Moreover, even if guarantees did exist for these DP algorithms, non-trivial subtleties arise when using such guarantees to analyze the overall estimation scheme, as was seen in Section 3.5.1; these subtleties were not addressed in [16] or [17] either. In contrast, our algorithm and its analysis provide the guarantees one would desire in practice: an accuracy guarantee for all estimates, a complexity guarantee for certain choices of K_n, and evidence for why choosing K_n with high in-degree is a good choice. It is also worth noting that Berkhin's idea to improve runtime by controlling sparsity is the same idea we use in this chapter; see Section 3.5.1.

Another relevant work is [36]. Here the authors provide an algorithm to estimate π_v s.t.

$$(1-\varepsilon')\pi_v(u) - \varepsilon_n \leq \hat{\pi}_v(u) \leq (1+\varepsilon')\pi_v(u) + \varepsilon_n \;\forall\; u \in V_n.$$

The complexity of this scheme is $O((\log n)^2 \log(1/\varepsilon_n)/(\varepsilon_n(\varepsilon')^2))$ per node. Hence, setting $\varepsilon_n = \varepsilon/n$ for some constant ε, setting ε' small and independent of n, and running this scheme for all nodes gives complexity $O(n^2(\log n)^3)$, similar to `Approx-PageRank` and `Approx-Contributions`. However, the multiplicative error term ε' cannot be avoided, so this is not quite an l_1 error guarantee. The analysis in [36] also shows via a tight lower bound that the complexity bound is within a polylogarithmic factor of the optimal. Naively using their lower bound independently for each node would lead to the erroneous claim that any algorithm for estimating the PPR matrix should have complexity at least $\Omega(n^2)$. Critically, though, properly accounting for the dependence of the PPR vectors based on the common underlying graph allows us to do better than this naive conjectured lower bound.

We also believe our analysis can used to tighten complexity bounds of other algorithms, similar to the analysis we conducted here for a modified version of the algorithm from [16]. For example, the algorithms from Chapter II and [15] estimate $\pi_v(u)$ with complexity $O(\sqrt{n}\log n)$ when $L_n = O(n), \alpha_n = \Theta(1/\log n)$, and with accuracy guarantee

$$|\hat{\pi}_v(u) - \pi_v(u)| \leq \begin{cases} \varepsilon\pi_v(u), & \pi_v(u) \geq 1/n \\ 2e/n, & \pi_v(u) < 1/n \end{cases}, \tag{3.9}$$

i.e. a relative error bound when $\pi_v(u)$ is large and an absolute error guarantee otherwise. If one desires this accuracy guarantee for the entire matrix $\{\pi_v(u)\}_{v,u\in V_n}$, the scheme can be run separately for every v, u pair at complexity $O(n^{2.5}\log n)$ (which is strictly better than computing the matrix via the inverse in (1.2)). However, the basic approach we have used here could also be used to reduce this complexity; namely, by first using the scheme to estimate $\{\pi_k\}_{k\in K_n}$ and $\{\tilde{\pi}_v(k)\}_{k\in K_n}$ and then using (3.7) to estimate $\{\pi_v(u)\}_{v,u\in V_n\setminus K_n}$. The challenge of designing such an algorithm would be similar to the challenge we encountered in this section. Specifically, we had to carefully design certain aspects of Algorithm 3.1 – "sparsifying" the matrix $\{\hat{\pi}_k\}_{k\in K_n}$ and checking if $\hat{\tilde{\pi}}_v(K_n)$ exceeds $g_n(\varepsilon_1)$ – to ensure that estimation errors pertaining to $\{\pi_k\}_{k\in K_n}$ and $\{\tilde{\pi}_v(k)\}_{k\in K_n}$ did not propagate through to estimates of $\{\pi_v(u)\}_{v,u\in V_n\setminus K_n}$ and render them highly inaccurate. Similarly, using our framework to estimate the PPR matrix using the algorithm of [15] would require a careful analysis of how the errors in (3.9) propagate through to later estimates. To summarize, we believe our basic approach can be used to design modified versions of other existing algorithms; however, we suspect this is nontrivial, suggesting an avenue for future work.

As mentioned in Section 3.2.3, our algorithm also bears resemblance to nonnegative matrix factorization (NMF), which, given $X \in \mathbb{R}^{n\times m}$, seeks $W \in \mathbb{R}^{n\times r}, H \in \mathbb{R}^{r\times m}$ such that $\|X - WH\|$ is small (where typically $r \ll m, n$). This is directly analogous to our algorithm. Indeed, Algorithm 3.2 generates $W \in \mathbb{R}^{n\times\Delta(K_n,\varepsilon/14)}, H \in \mathbb{R}^{\Delta(K_n,\varepsilon/14)\times n}$ satisfying

$$\|(\Pi_n - \alpha_n I) - WH\|_\infty < \varepsilon,$$

where $\|A\|_\infty = \max_i \|a_i\|_1$ for a matrix A with rows $\{a_i\}$. However, there are some key differences between NMF and our scheme. First, NMF assumes X is known, while in our algorithm $\Pi_n - \alpha_n I$ (which plays the role of X) is unknown. This means that standard NMF algorithms, which compute gradients dependent on X to iteratively update W, H (see e.g. [70]), do not apply. Additionally, computing these gradients requires the objective function $\|X - WH\|$ to be differentiable; in contrast, we use the non-differentiable norm $\|\cdot\|_\infty$, which further prohibits use of standard NMF algorithms. Finally, NMF chooses the dimensions of W, H *a priori*, while our algorithm adjusts r at runtime so as to minimize complexity (ultimately yielding $r = \Delta(K_n, \varepsilon/14)$ as above). In short, our algorithm can be viewed as a variant of NMF, tailored to the PPR setting in a manner that guarantees high accuracy and low complexity. We discuss this in more detail in Appendix B.8.1. We also that NMF with an objective function similar to ours has been studied in the contextual bandits literature. Namely, [71] analyzes a model for which $X_{i,j}$ gives the mean reward of pulling arm i given context j. The authors propose a regret-minimization algorithm to minimize

$\|X - WH\|_{\infty,\infty}$, where $\|A\|_{\infty,\infty}$ is the maximal element (in absolute value) of a matrix A. However, [71] assumes a small dimensionality by assuming a particular generative model for W, H; in contrast, this chapter develops conditions to prove a small dimensionality.

In light of this, we mention another matrix problem of relevance, low-rank approximation. For low-rank approximation, we are given a matrix $X \in \mathbb{R}^{n\times m}$ and aim to solve

$$\inf_{\hat{X}\in\mathbb{R}^{n\times m}} \|X - \hat{X}\| \text{ s.t. } \text{rank}(\hat{X}) \leq r,$$

for some $r \in \{1, \ldots, n\}$ (for example, when $\|\cdot\|$ is the spectral or Frobenius norm, the minimizing $\hat{X}$ is a truncated singular value decomposition , see e.g. Section 2.4 of [72]). A related problem, which can be viewed as the dual of low-rank approximation, is

$$\min_{\hat{X}\in\mathbb{R}^{n\times m}} \text{rank}(\hat{X}) \text{ s.t. } \|X - \hat{X}\| < \varepsilon. \tag{3.10}$$

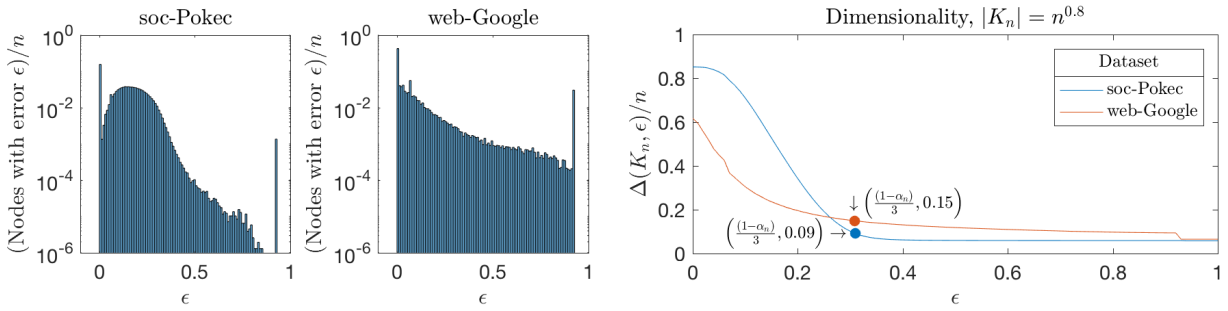
Note our dimensionality measure (3.1) is a variant of (3.10) in which the minimum is taken over a restricted class of matrices (those containing a subset of rows of the original matrix, in addition to linear combinations of this subset). Hence, as in the above discussion of NMF, our scheme can be viewed as a tailored version of low-rank approximation.

Finally, we note estimating the PPR matrix can be viewed as a special case of the algorithm from [73], which studies representation learning on graphs. However, we believe this algorithm offers worse performance than ours in the PPR setting; see Appendix B.8.2.

3.6 Experiments

In this section, we illustrate various aspects of our analysis with some empirical results. Our goal is to demonstrate that our theoretical findings – namely, that π_s can be well-approximated as a linear combination of $\{\pi_k\}_{k\in K_n}$ (Lemma 3.1) and that PPR dimensionality is small (Theorem 3.1) – may still hold when our assumptions fail. To this end, we will estimate our dimensionality measure for two real graphs, as well as showing that π_s is indeed well-approximated for a wider class of real graphs. Additionally, since our theoretical results are asymptotic statements, we will investigate the relevant quantities as n grows for the DCM. We note that, unless otherwise mentioned, we choose K_n as the n^κ nodes of highest in-degree, per the discussion of Section 3.7.1 (with κ specified for each experiment).

We first estimate $\Delta(K_n, \varepsilon)$ for a range of ε when $\kappa = 0.8$ (0.8 was chosen as we found it roughly balanced the two summands in (3.1)). For this, we use two datasets from the Stanford Network Analysis Platform (SNAP) [43] that align with the motivating applications of Section 1.2.3: soc-Pokec, a social network, and web-Google, a partial web graph (see



(a) Histograms of the error term (3.11)

(b) $\Delta(K_n, \varepsilon)$ is a small fraction of n when $\varepsilon = \frac{1-\alpha_n}{3}$ (ensuring estimation error is reduced by a factor of 3)

Figure 3.1: Dimensionality for social network soc-Pokec and partial web crawl web-Google.

Appendix B.7.1 for details). We set $\alpha_n = 1/\log n$, and, $\forall\ v \notin K_n$, compute a bound on

$$\left\| \pi_v - \left(\alpha_n e_v^\mathsf{T} + \frac{\sum_{k \in K_n} \tilde{\pi}_v(k) \pi_k}{\alpha_n + (1-\alpha_n)\tilde{\pi}_v(K_n)} \right) \right\|_1 \tag{3.11}$$

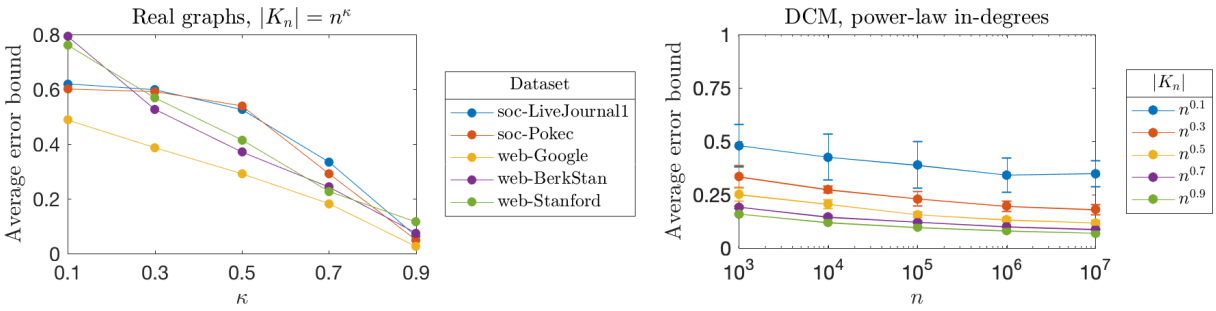
using a power iteration scheme described in Appendix B.7.2. Adding $n^{0.8}$ to the number of $v \in V_n \setminus K_n$ for which this bound exceeds ε then gives a bound on $\Delta(K_n, \varepsilon)$. Figure 3.1a shows histograms of the error bound, while Figure 3.1b shows our dimensionality measure. We highlight two points on Figure 3.1b, $((1-\alpha_n)/3, 0.09)$ for soc-Pokec and $((1-\alpha_n)/3, 0.15)$ for web-Google. We believe $(1-\alpha_n)/3$ is a reasonable choice of ε because (as proven in (B.31)) (3.11) is bounded by $1-\alpha_n$; hence, this choice of ε reduces the worst-case error term by a factor of 3. Note $\Delta(K_n, (1-\alpha_n)/3)$ is small for both datasets – 9% and 15% of nodes, respectively. This suggests that while Theorem 3.1 does not apply, the dimension of $\{\pi_v\}_{v \in V_n}$ appears small, and that while Theorem 3.2 does not apply, Algorithm 3.1 should be efficient.

We offer several other remarks on Figure 3.1. First, as proven in Appendix B.7.2, (3.11) is zero for v with no outgoing neighbors in $V_n \setminus K_n$, i.e. for any $v \in V_{n,0}$, where

$$V_{n,0} = \{v \notin K_n : \nexists\ (w, w') \in E_n \text{ s.t. } w = v, w' \notin K_n\}. \tag{3.12}$$

As a result, the "spikes" at $\varepsilon = 0$ in Figure 3.1a have height $|V_{n,0}|/n$, and $\Delta(K_n, 0) = n - |V_{n,0}|$ in Figure 3.1b. Next, the aforementioned claim that (3.11) is bounded by $1-\alpha_n$ explains the spikes at right in Figure 3.1a and the "dips" at right in Figure 3.1b (both of which occur at $\varepsilon = 1-\alpha_n$). Finally, we observe that, between the spikes at $\varepsilon = 0$ and $\varepsilon = 1-\alpha_n$, the soc-Pokec histogram quickly decays beyond $\varepsilon \approx 0.3$; this corresponds to the dimensionality being nearly flat beyond $\varepsilon \approx 0.3$ in Figure 3.1b.

Computing (3.11) for every $v \in V_n \setminus K_n$ requires significant computational time, but (as



(a) Average error decreases as $|K_n|$ grows for a variety of social networks and web graphs.

(b) For $s \sim V_n \setminus K_n$ uniformly on the DCM with power law in-degrees, error decreases as n grows.

Figure 3.2: Average error experiments for real and synthetic datasets.

described in Appendix B.7.2) we can also compute a bound on the average error

$$\frac{1}{|V_n \setminus K_n|} \sum_{v \in V_n \setminus K_n} \left\| \pi_v - \left(\alpha_n e_v^\mathsf{T} + \frac{\sum_{k \in K_n} \tilde{\pi}_v(k) \pi_k}{\alpha_n + (1-\alpha_n)\tilde{\pi}_v(K_n)} \right) \right\|_1 \tag{3.13}$$

much more efficiently. Hence, we show this average error bound for a wider variety of graphs in Figure 3.2a. Interestingly, the social networks soc-LiveJournal1 and soc-Pokec have similar behavior, as do the web graphs web-BerkStan and web-Stanford (web-Google is somewhat of an outlier; we believe this is in part because its $|V_{n,0}|$ is largest).

We next replicate this average error experiment for two synthetic graphs, which allows us to observe how the bound on (3.13) evolves as n grows. The first graph we consider is a DCM with power law in-degrees with exponent 2, i.e. $\mathbb{P}[N_v = i] \propto i^{-2}\ \forall\ v \in V_n, i \in \{1, 2, \ldots, n\}$, and out-degrees generated as in Algorithm 3.4 from Section 3.7.3 (which, in expectation, gives constant out-degree to each node). We note this in-degree model is a common one for many graphs observed in practice (see Section 3.7.2 for details); however, Lemma 3.1 does not apply, as the in-degree sequence does not satisfy all of our assumptions (for instance, the expected in-degree does not converge). Nevertheless, Figure 3.2b shows that the average error bound decays as n grows across all choices of κ. We suspect this is in part because, while the degree sequence does not satisfy all of our assumptions, empirical results show that it contains a vanishing fraction of nodes with a non-vanishing fraction of edges (see Section 3.7.2). We believe this to be the most important of our assumptions.

Finally, we replicate the average error experiment with the sequence $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{U}_n)$ generated via Algorithm 3.4 in Section 3.7.3; this sequence provably satisfies Assumption 3.2. We note that Algorithm 3.4 chooses $|K_n|$ in a manner that guarantees $\mathbb{E}[|K_n|] = n^\kappa$ (unlike previous experiments, for which $|K_n| = n^\kappa$ by design). Hence, in Figure 3.3a, we show the average error bound for a variety of $\mathbb{E}[|K_n|]$ choices. For choices of κ at or above 0.5, the

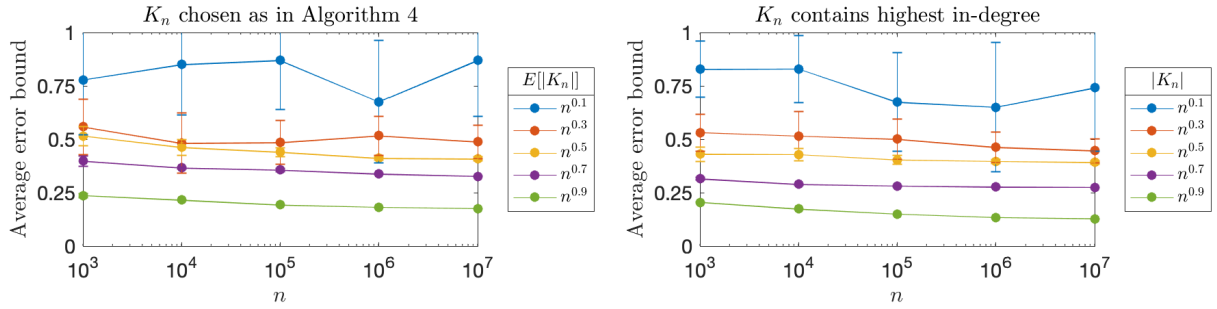


Figure 3.3: For $s \sim V_n \setminus K_n$ uniformly on the DCM with the degrees from Algorithm 3.4, error decreases as n grows.

average error bound slightly decays as n grows (though less notably than in Figure 3.2b). We also conduct an experiment for which $(\mathbf{N}_n, \mathbf{D}_n)$ is generated via Algorithm 3.4 but K_n is chosen as the nodes of highest in-degree; results are shown in Figure 3.3b. The average error bound is slightly smaller for each κ value than in Figure 3.3a; this again suggests that K_n being the nodes of highest in-degree is indeed a good choice.

3.7 Discussion

Before closing the chapter, we discuss several other aspects of our analysis, including the "optimal" choice of K_n, the restrictiveness of our assumptions, an example sequence $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{U}_n)$ that satisfies our assumptions, a connection between our result and recent work on mixing times, and a geometric interpretation of our dimensionality result.

3.7.1 Choice of hub nodes

A natural question is which choice of K_n gives the smallest exponent in Theorem 3.1. For this, first note the exponent grows with κ and p and decays with δ (ε, τ, ρ, and ζ also appear in the exponent, but all are independent of K_n). The growth with κ and p suggests a good choice of K_n is a small set of nodes (small κ) containing a large fraction of instubs (small p). In particular, this suggests choosing K_n to be the nodes with highest in-degree. We note the authors of [16] heuristically choose K_n to be the nodes with highest global PageRank, and we showed in Section 3.5 that the complexity of a similar algorithm relates to $\Delta(K_n, \varepsilon)$ on the DCM. Since global PageRank is suspected to correlate closely with in-degree for many graphs (see e.g. the aforementioned [51, 48, 50, 52, 53, 54] and the empirical works [74, 75]), our analysis appears to validate this heuristic. However, it is difficult to prove that choosing the highest in-degree nodes as K_n gives the smallest exponent, in part because exponent decays with δ, which interacts with K_n more subtly (see Assumption 3.1).

Choosing K_n as nodes of high in-degree can also be motivated using results concerning

the simple random walk on the DCM. In particular, it is known that the distribution of this walk is close to the stationary distribution after $\Theta(\log n)$ steps when starting from an arbitrary node (Theorem 1 in [56]), but is close to stationarity after just a constant number of steps when starting from the in-degree distribution (Theorem 3 in [56]).[5] In other words, the in-degree distribution is a good initial guess for the stationary distribution. This suggests that high in-degree nodes are reached quickly on random walks. On the other hand, our analysis states π_v is close to a linear combination of $\{\pi_k\}_{k\in K_n}$ when walks from v are likely to hit K_n (see Appendix B.1.1). In summary, choosing high in-degree nodes as K_n means walks are likely to reach K_n, which in turn means π_v is likely well-approximated by $\{\pi_k\}_{k\in K_n}$.

3.7.2 Comments on assumptions

At a high level, our assumptions fall into two groups: the events $\{\Omega_{n,i}\}_{i=1}^5$ in Assumption 3.1, which say that the degree sequence is sparse, and the event $\Omega_{n,6}$ in Assumption 3.1, which (in light of Assumption 3.2) says that a vanishing fraction of nodes contains a non-vanishing fraction of in-degrees. We discuss each of these in turn.

For the sparsity assumptions, we note $\{\Omega_{n,i}\}_{i=1}^3$ in Assumption 3.1 are fairly standard given our approach, which leverages the fact that the random graph is locally tree-like [76]. In fact, $\Omega_{n,3}$ is a weaker assumption than that required in e.g. [48]; see Appendix B.1.3 for details. Next, we argue $\Omega_{n,4}, \Omega_{n,5}$ in Assumption 3.1 are not restrictive. For this, first note that given $\Omega_{n,1}$ and $\Omega_{n,6}$ in Assumption 3.1, and since $\sum_{h=1}^n U_h = \Theta(n)$ by Assumption 3.2,

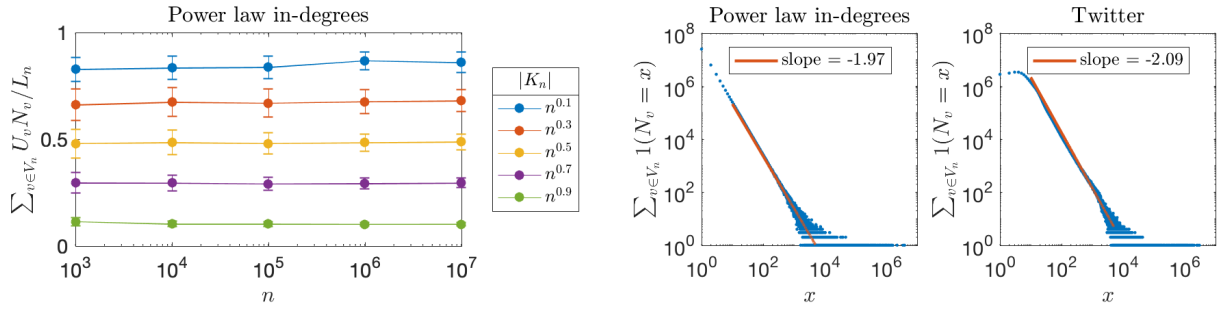
$$\lim_{n\to\infty} \frac{\sum_{h=1}^n U_h N_h}{\sum_{h=1}^n U_h} = \lim_{n\to\infty} \frac{\frac{\sum_{h=1}^n U_h N_h}{\sum_{h=1}^n N_h} \frac{\sum_{h=1}^n N_h}{n}}{\frac{1}{n}\sum_{h=1}^n U_h} = p\eta_1 < \infty,$$

i.e. $\sum_{h=1}^n U_h N_h / \sum_{h=1}^n U_h$ converging to a finite limit is implied by other assumptions; additionally, Assumption 3.2 implicitly requires $\lambda^* = p\eta_1$. (We have written $\Omega_{n,5}$ as its own assumption only out of convenience.) Similarly, $\Omega_{n,4}$ is essentially implied by $\sum_{h=1}^n U_h = \Theta(n)$ and $\Omega_{n,1}$, since then the fraction in $\Omega_{n,4}$ satisfies

$$\frac{\sum_{h=1}^n U_h D_h}{\sum_{h=1}^n U_h} \leq \frac{\frac{1}{n}\sum_{h=1}^n D_h}{\frac{1}{n}\sum_{h=1}^n U_h} \xrightarrow[n\to\infty]{} \eta_1 < \infty.$$

For the remaining assumption, we recall that $\Omega_{n,6}$ requires $\sum_{v\in V_n} U_v N_v / L_n$ to converge to $p < 1$ with $|K_n|$ sublinear by Assumption 3.2. We offer empirical evidence that this occurs for certain graphs of interest. Specifically, in Figure 3.4a, $\sum_{v\in V_n} U_v N_v / L_n$ remains constant

[5] These results require $\max\{\max_{v\in V_n} D_v, \max_{v\in V_n} N_v\} = O(1)$, which is a stronger sparsity condition than we have assumed in this chapter. In fact, we suspect that max in-degree is *not* $O(1)$ for many sequences to which our results apply (see Sections 3.7.2 and 3.7.3), so this discussion is not entirely rigorous.



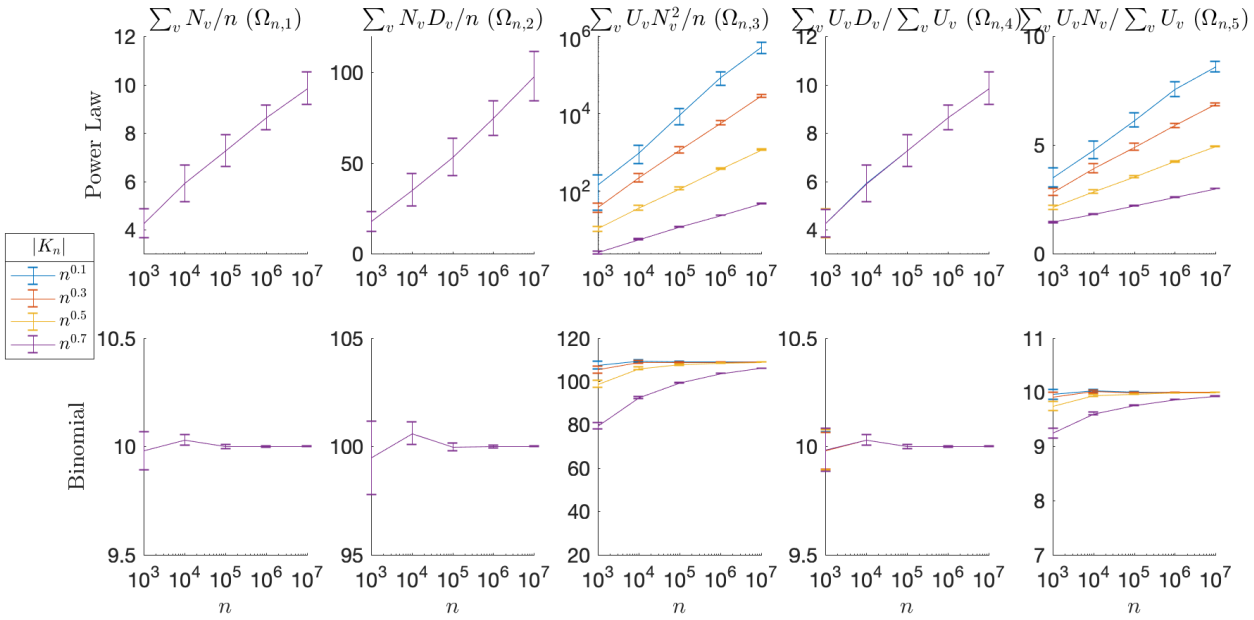
(a) K_n is sublinear and contains a constant fraction of instubs. (Here K_n are nodes of highest in-degree.)

(b) The in-degrees for Fig. 3.4a are similar to in-degrees for the Twitter graph from [77]. ($n \approx 4\times 10^7$.)

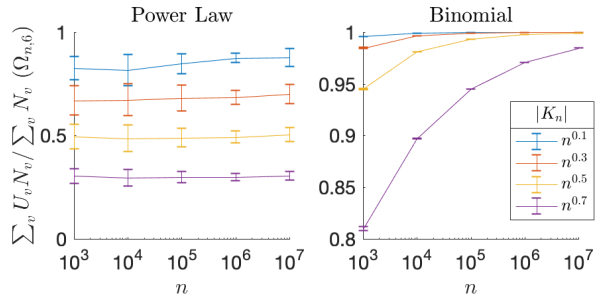
Figure 3.4: $\Omega_{n,6}$ is empirically satisfied for power law in-degrees similar to Twitter.

and strictly less than 1 as n grows, for a variety of sublinear $|K_n|$ choices (here we take K_n to be the nodes of highest in-degree, as in Section 3.6). For this plot, in-degrees were sampled from a power law with exponent 2, i.e. $\mathbb{P}[N_v = i] \propto i^{-2}$. This in-degree distribution is commonly seen in real graphs and has been studied extensively, see e.g. [78, 79]. As an example, Figure 3.4b compares the histogram of these in-degrees with the in-degrees of the Twitter graph (available at [77] from WebGraph [80]). Both histograms are linear with slopes ≈ -2 over $x \in [10, 5000]$. In short, a common model of in-degree distributions for graphs aligning with the applications of Section 1.2.3 empirically satisfies $\Omega_{n,6}$ with $|K_n|$ sublinear.

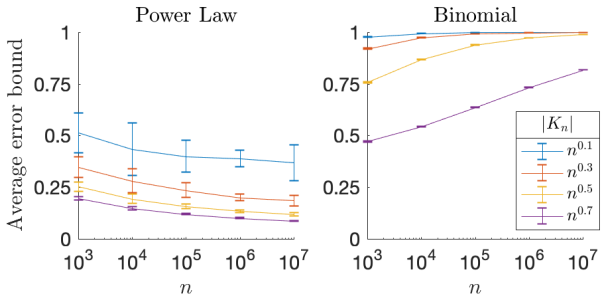
Ultimately, we believe this last assumption is fundamentally necessary, while the sparsity assumptions may be artifacts of our analysis. To illustrate this, we compare the same power law in-degree sequence to a sequence of binomial in-degrees with parameters n and $10/n$ (i.e. Poisson(10) in-degrees asymptotically). For both sequences, we first realize in-degrees independently and choose K_n to be the nodes of highest in-degree; we then generate out-degrees as in Section 3.7.3. In Figure 3.5a, we observe the moments appearing in $\{\Omega_{n,i}\}_{i=1}^5$ grow without bound as n grows for the power law case but converge to constants for the binomial case. On the other hand, Figure 3.5b shows the quantity appearing in $\Omega_{n,6}$ converges to $p < 1$ for the power law case but rapidly approaches 1 for the binomial case. In short, the sparsity assumptions fail while the remaining assumption holds for power law in-degrees; the opposite is true for binomial in-degrees. From Figure 3.5c, we observe the average error bound (3.13) (computed as in Section 3.6) decays to 0 for the power law case but grows to 1 for the binomial case. Hence, we ultimately conclude the following: when the sparsity assumptions fail but the remaining assumption holds, $\pi_v, v \notin K_n$ is typically well-approximated as a linear combination of $\{\pi_k\}_{k\in K_n}$, so our dimensionality measure should be small; when sparsity holds but the remaining assumption fails, the opposite is true. This suggests that the sparsity assumption is a less necessary assumption.



(a) Assumption 3.1 requires the quantities in $\{\Omega_{n,i}\}_{i=1}^5$ to converge to finite limits; this fails for the power law case (top row) but occurs for the binomial case (bottom row)



(b) Assumption 3.1 requires the quantity in $\Omega_{n,6}$ to converge to $p < 1$; this holds for the power law case (left) but fails for the binomial case (right)



(c) The average error bound (3.13) decays to 0 for the power law case (left) but increases to 1 for the binomial case (right)

Figure 3.5: Power law in-degrees satisfy only our most crucial assumption (Fig. 3.5a,3.5b), but average estimation error decreases, suggesting low dimensionality (Fig. 3.5c); the opposite is true for binomial in-degrees.

3.7.3 Example degree sequence

We next provide an example of a degree sequence satisfying Assumption 3.2. This is meant as a coarse model of a network like Twitter: roughly speaking, it will contain a small number of nodes with huge in-degrees (corresponding to celebrities on Twitter with millions of followers) and a large number of nodes with small in-degree (corresponding to "normal" users with tens or hundreds of followers); additionally, all nodes will have out-degrees that (in expectation) do not scale with n (a given Twitter user does not follow a sizeable portion of all users). Specifically, given $c_1, c_2 > 0$ and $\kappa, l_1, l_2 \in (0,1)$, we assign degrees and choose K_n via Algorithm 3.4. In words, we assign in-degrees as a mixture of two (truncated) power laws; after realizing in-degrees, each node initially receives one outstub, and the remaining $\sum_{w \in V_n}(N_w - 1)$ outstubs are each assigned uniformly. (Note that this guarantees $N_v, D_v \in \mathbb{N}\ \forall\ v \in V_n$ and $\sum_{v \in V_n} N_v = \sum_{v \in V_n} D_v$, as we have assumed throughout the chapter.) Proposition 3.2 states that the resulting sequence satisfies our assumptions.

Algorithm 3.4: Example degree sequence construction

1 $\forall\ v \in V_n$, let $U_v \sim \text{Bernoulli}(1 - n^{\kappa - 1})$ and sample N_v as

$$\mathbb{P}[N_v = i | U_v = 1] = \frac{i^{-c_1}}{\sum_{j=1}^{\lceil n^{l_1} \rceil} j^{-c_1}}\ \forall\ i \in \{1, \dots, \lceil n^{l_1} \rceil\}$$

$$\mathbb{P}[N_v = i | U_v = 0] = \frac{i^{-c_2}}{\sum_{j=1}^{\lceil n^{l_2} \rceil} j^{-c_2}}\ \forall\ i \in \{1, \dots, \lceil n^{l_2} \rceil\}$$

2 Sample $\nu_i \sim V_n$ uniformly $\forall\ i \in \{1, 2, \dots, \sum_{w \in V_n}(N_w - 1)\}$

3 $\forall\ v \in V_n$, set $D_v = 1 + \sum_{i=1}^{\sum_{w \in V_n}(N_w - 1)} 1(\nu_i = v)$

Proposition 3.2. Assume $c_1 \in (3,4)$, $c_2 \in (1,2)$, $l_1 \in (0, 1/(5 - c_1))$, $l_2 \in (0,1)$, and $\kappa = 1 - l_2(2 - c_2)$. Then $(\mathbf{N}_n, \mathbf{D}_n, \mathbf{U}_n)$ generated by Algorithm 3.4 satisfies Assumption 3.2.

Proof. The proof is tedious but elementary and can be found in [45, Appendix I]. □

As an example of parameter choices for Proposition 3.2, we can take $c_1 = 3.1$, $c_2 = 1.1$, $l_1 = 0.5$, $l_2 = 0.9$, and $\kappa = 0.19$. In this case, nodes belonging to $V_n \setminus K_n$ have maximum in-degree $n^{0.5}$, while nodes in K_n have maximum in-degree $n^{0.9}$; additionally, the in-degree distribution for K_n has a heavier tail (since $c_2 < c_1$). This is consistent with the discussion of Section 3.7.1, where we argued K_n should contain high in-degree nodes.

We note that, per Figure 3.4, a more appropriate model for the Twitter in-degree sequence would be a power law with exponent ≈ 2. However, Assumption 3.2 requires the second moment of N_v to converge for $v \in V_n \setminus K_n$; when N_v follows a power law for such v, this

requires the exponent to exceed 3 (hence the requirement $c_1 > 3$ in Proposition 3.2). On the other hand, recall that our analysis uses a result from [48] but with weaker assumptions. Specifically, we only require $\sum_v U_v N_v^2 = O(n)$, whereas [48] requires $\sum_v N_v^2 = O(n)$ (see Appendix B.1.3). Weakening the assumption in this manner allows for $N_v, v \in K_n$ to follow a power law with exponent $c_2 \in (1, 2)$ in Proposition 3.2. This is turn yields an in-degree distribution with a heavier tail than if *all* in-degrees were restricted to power law with exponent exceeding 3, which allows our model to more closely resemble the exponent ≈ 2 case. In particular, $c_1 \in (3, 4), c_2 \in (1, 2)$ yields an in-degree sequence with bounded mean but unbounded variance as $n \to \infty$, as does a power law with e.g. exponent 2.1.

Finally, we note $c_1 < 4$ is not necessary to prove Assumption 3.2 holds but allows us to avoid addressing separate cases in the proof; also, taking $c_1 > 4$ would yield a less accurate model of power law sequences observed in practice, which often have exponents ≈ 2.

3.7.4 Connection to mixing times

We can also motivate our choice of α_n in terms of the mixing time of the simple random walk on G_n. First, we let π denote the stationary distribution of this walk. For any $v \in \mathbb{N}$, and for a graph of n nodes, we let π_v be the uniform distribution on V_n for $n < v$, and we define π_v as in Section 3.2.2 for $n \geq v$. (This ensures π_v is well-defined in what follows.) We can then prove the following, which shows that π_v is (asymptotically) indistinguishable from π when $\alpha_n = o(1/\log n)$ and when a certain mixing condition is satisfied.

Proposition 3.3. Let $v \in \mathbb{N}$, $m = \Theta(\log n)$, $\alpha_n = o(1/\log n)$, and $\varepsilon > 0$. Then for n sufficiently large, we have for any G_n,

$$\|\pi_v - \pi\|_1 \leq 3 \max_{w \in V_n} \left\| e_w^\mathsf{T} P^m - \pi \right\|_1 + \varepsilon.$$

As a consequence, if

$$\max_{w \in V_n} \left\| e_w^\mathsf{T} P^m - \pi \right\|_1 \xrightarrow[n \to \infty]{\mathbb{P}} 0, \tag{3.14}$$

then $\|\pi_v - \pi\|_1 \xrightarrow[n \to \infty]{\mathbb{P}} 0$, where $\xrightarrow{\mathbb{P}}$ denotes convergence in probability.

Proof. See Appendix B.6. □

Proposition 3.3 states that when $\alpha_n \log n \to 0$ and the random walk on G_n mixes in $\log n$ steps (in the sense of (3.14)), π_v is close to π in l_1 (for large n and with high probability). Put differently, Π_n is close to the rank one matrix $1_n \pi$ in this case, suggesting a dimensionality of 1. This is in fact somewhat obvious: since the first restart at v on the Markov chain defining π_v occurs at time $1/\alpha_n$ (in expectation), the chain reaches stationarity after $\log n$ steps but does not restart at v until e.g. $(\log n)^2$ steps, so π_v should not depend on v.

We believe that the mixing condition (3.14) holds for our graph model. This belief is based on recent work by Bordenave, Caputo, and Salez, who prove (3.14) for a class of sparse, randomly-generated Markov chains (Theorem 1 in [81]). In particular, this class includes random walks on random graphs with a given degree sequence (i.e. the DCM). The key differences between this model and ours are (1) we permit multi-edges, while the model in [81] does not, and (2) $D_v > 1, D_v = O(1)\ \forall\ v \in V_n$ in the [81] model.[6] We note that the condition $D_v > 1, D_v = O(1)$ can be added to Assumption 3.1 without contradiction; Assumption 3.2 then implies a graph with a few huge in-degrees, mostly small in-degrees, and all small out-degrees, as in Section 3.7.3.

We have thus far argued the dimensionality of $\{\pi_v\}_{v\in V_n}$ grows as n^c for some $c \in (0,1)$ when $\alpha_n = \Theta(1/\log n)$ (Theorem 3.1) and is constant when $\alpha_n = o(1/\log n)$ (Proposition 3.3). A third case is $\alpha_n = \omega(1/\log n)$. For this case, we first note [81, 56] prove a matching lower bound to (3.14), i.e. they show for some $m' = \Theta(\log n)$,

$$\min_{w\in V_n} \|e_w^{\mathsf{T}} P^{m'} - \pi\|_{TV} \xrightarrow[n\to\infty]{\mathbb{P}} 1,$$

where $\|\cdot\|_{TV}$ denotes total variation distance. Hence, the number of restarts at v before mixing scales with $\alpha_n \log n$ in expectation, which is unbounded in the case $\alpha_n = \omega(1/\log n)$. In contrast, only a constant number of restarts at v occur before mixing in the $\alpha_n = \Theta(1/\log n)$ case. For this reason, we conjecture that $\Delta(K_n, \varepsilon)$ behaves fundamentally differently if $\alpha_n = \omega(1/\log n)$, perhaps dominating n^c for any $c \in (0,1)$ (e.g., $n/\log n$ or even n).

Ultimately, this discussion further explains our choice of α_n: if we set α_n much smaller, we obtain dimensionality 1; if we set α_n much larger, we expect to obtain a much larger dimensionality. Hence, our choice of α_n yields the strongest possible result before trivial behavior occurs. Finally, we note this "trichotomy" – $O(1)$, $O(n^c)$, and $\Omega(n/\log n)$ dimensionality if α_n is very small, moderate, or very large – is further explored in Chapter V.

3.7.5 Geometric interpretation

Before closing, we note Theorem 3.1 has a simple geometric interpretation. To see this, first recall that for all but a vanishing fraction of $v \in V_n \setminus K_n$, the theorem states

$$\left\| \pi_v - \frac{\sum_{k\in K_n} \tilde{\pi}_v(k)\pi_k}{\alpha_n + (1-\alpha_n)\tilde{\pi}_v(K_n)} \right\| < \varepsilon.$$

[6] For simplicity, we stated a slightly stronger condition than requiredin [81]; see Example 1 in [81] for details.

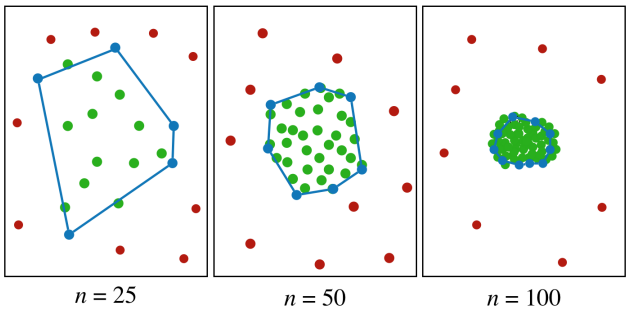


Figure 3.6: As n grows, most of $\{\pi_v\}_{v\in V_n\setminus K_n}$ (green dots) concentrate near the convex hull of $\{\pi_k\}_{k\in K_n}$ (blue dots/lines) (a few of $\{\pi_v\}_{v\in V_n\setminus K_n}$ (red dots) can be far away). Further, since $|K_n|$ shrinks relative to n, the convex hull of $\{\pi_k\}_{k\in K_n}$ shrinks relative to the n-dimensional simplex.

Furthermore, for such v, we have by (B.81) in Appendix B.4,

$$\frac{\tilde{\pi}_v(K_n)}{\alpha_n + (1-\alpha_n)\tilde{\pi}_v(K_n)} \geq \frac{1}{\frac{\varepsilon+\alpha_n(2-(\varepsilon+\alpha_n))}{1-(\varepsilon+\alpha_n)} + (1-\alpha_n)}. \tag{3.15}$$

Note the left side of (3.15) is upper bounded by 1, while the right side tends to $1-\varepsilon$ as $n \to \infty$. The previous two equations can then be interpreted as follows: setting ε arbitrarily small, and letting n grow large, π_v is arbitrarily close to a linear combination $\{\pi_k\}_{k\in K_n}$; furthermore, the weights for the linear combination are nonnegative and their sum is arbitrarily close to 1. Taken together, π_v is arbitrarily close to the convex hull of $\{\pi_k\}_{k\in K_n}$. Additionally, because $B_v(K_n, \varepsilon)$ fails with high probability, all but a vanishing fraction of $\{\pi_v\}_{v\in V_n}$ are arbitrarily close to this convex hull. Finally, because $|K_n|$ scales sublinearly in n, this convex hull is a low dimensional subset of the n-dimensional simplex to which $\{\pi_v\}_{v\in V_n}$ belong. Hence, beyond describing the dimensionality of the set of PPR vectors, our dimensionality result also describes the space in which most of these vectors reside. This interpretation is depicted graphically in Figure 3.6; we note this figure is simply an illustration of the preceding paragraph and was not generated using actual PPR vectors.[7]

3.8 Conclusions and future directions

In this chapter, we argued (analytically for the DCM and empirically for other graphs) that the dimensionality of the PPR matrix scales sublinearly in n. We also used our analysis to bound the complexity of an algorithm to compute all PPR vectors, which is similar to that found in [16]. Our analysis suggests several avenues for future work. First, the proof of

[7] Generating such a figure with actual PPR vectors is difficult because n-dimensional vectors must be projected into 2D space while roughly preserving l_1 distances, and such a projection is not well understood [82, 83]. Appendix B.7.4 includes a figure obtained from actual PPR vectors, but it is less illustrative.

Lemma 3.1 can be modified to analyze the tail of the l_∞ error (this would essentially involve replacing Lemma B.5 with a tail bound on a maximum instead of a sum). Hence, bounding *absolute* error for the estimate of $\pi_s(v)$ for any $v \in V_n$ is a straightforward extension; a less immediate variant would involve bounding *relative* error. Second, examining PPR dimensionality for other random graph models may be of interest. For example, several papers have analyzed PPR on preferential attachment models [84, 85]; we suspect a dimensionality analysis for such graphs would yield a message similar to this chapter (K_n should contain nodes with high in-degrees). A more interesting class of graphs would be from the stochastic block model; here it may be more beneficial to choose K_n such that each community contains a nonempty subset of K_n. Finally, as discussed in Section 3.5.2, we believe our analysis and our approach to analyzing the algorithm from [16] can be used to design improved versions of existing algorithms and derive tighter complexity bounds.

CHAPTER IV

Empirical Policy Evaluation with Supergraphs

Important remark on notation: In this chapter, we use notation from the reinforcement learning literature: π will denote a policy, not to be confused with the PageRank vector used in other chapters, and the roles of α and $1-\alpha$ are reversed compared to other chapters.

4.1 Introduction

Reinforcement learning (RL) is a machine learning paradigm with potential for impact in many applications. At its most basic level, RL studies autonomous agents interacting with uncertain environments, by taking actions and observing the effects of those actions, in hopes of achieving some goal. Mathematically, this is often cast in the following (finite, discrete-time) Markov decision process (MDP) model. Let $\mathcal{S}$ and $\mathcal{A}$ be finite sets called the *state space* and *action space*, respectively; for simplicity, we let $\mathcal{S} = \{1, \dots, S\}$ and $\mathcal{A} = \{1, \dots, A\}$ for some $S, A \in \mathbb{N}$. If the current state is $s \in \mathcal{S}$ and the agent takes action $a \in \mathcal{A}$, it incurs instantaneous cost $c(s,a) \in \mathbb{R}_+$ and transitions to state $s' \in \mathcal{S}$ with probability $Q(s'|s,a)$. Given *discount factor* $\alpha \in (0,1)$, the infinite horizon discounted cost problem is

$$\inf_{\pi} v_\pi(s) = \mathbb{E}_\pi \left[(1-\alpha) \sum_{t=0}^{\infty} \alpha^t c(S_t, A_t) \middle| S_0 = s \right], \tag{4.1}$$

where the infimum is over stationary, deterministic, Markov policies $\pi : \mathcal{S} \to \mathcal{A}$, i.e. mappings from the current state S_t to the current action A_t. It is well known that one can restrict to such policies without loss of optimality (see e.g. [86, Ch. 8]). Also, it is clear that each such policy π induces a transition matrix $Q_\pi(s,s') = Q(s'|s,\pi(s))$ and an instantaneous cost vector $c_\pi(s) = c(s,\pi(s))$ that depend only on the current state s. From this observation, one can use (4.1) to show that the vector $v_\pi = \{v_\pi(s)\}_{s\in\mathcal{S}}$ satisfies

$$v_\pi = (1-\alpha) \sum_{t=0}^{\infty} \alpha^t Q_\pi^t c_\pi = (1-\alpha)(I - \alpha Q_\pi)^{-1} c_\pi. \tag{4.2}$$

Assuming Q is known, several classical algorithms can be used to solve (4.1). For example, the *policy iteration* algorithm (see e.g. [86, Ch. 8]) solves (4.1) by iteratively computing (4.2) (the *policy evaluation* step) and greedily updating π (the *policy improvement* step). In the RL setting, however, one models the uncertainty of the environment by incomplete knowledge of Q. More specifically, Q is not known explicitly, but given state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, the agent can obtain samples from $Q(\cdot|s, a)$. The *empirical dynamic programming* approach thus adapts algorithms from the classical setting to the RL setting by replacing terms involving Q with empirical estimates (see e.g. [87, 88, 89, 90], and in particular [19] for the discounted cost problem). For instance, the policy evaluation step (4.2) becomes an *empirical policy evaluation* step, wherein (4.2) is estimated via samples from $Q_\pi(s, \cdot) = Q(\cdot|s, \pi(s))$.

In this chapter, we restrict attention to empirical policy evaluation. The policy π will thus be fixed for the remainder of the chapter, so we dispense with this subscript in (4.2) and (with slight abuse of notation) define our problem as follows. Let *discount factor* $\alpha \in (0, 1)$ and *cost vector* $c \in \mathbb{R}_+^S$ be given, and let Q be an unknown $S \times S$ row stochastic matrix. Our goal is to devise an algorithm to estimate the *value function*

$$v = (1 - \alpha) \sum_{t=0}^{\infty} \alpha^t Q^t c = (1 - \alpha)(I - \alpha Q)^{-1} c. \tag{4.3}$$

while accessing Q via samples, i.e. the algorithm has a subroutine which, given $s \in \mathcal{S}$, returns a sample from $Q(s, \cdot)$ (the s-th row of Q, a probability distribution over $\mathcal{S}$). Clearly, one can obtain an arbitrarily accurate estimate of v with enough samples. However, in RL applications this sampling requires costly interaction with the environment, and thus we face a trade-off between accuracy and sample complexity.

4.1.1 Conceptual motivation of our algorithms

To motivate our approach, we first discuss the existing approach from [19]. Let $\{W_t\}_{t=0}^{\infty}$ be a Markov chain with transition matrix Q, fix $s \in \mathcal{S}$ and $T \in \mathbb{N}$, and rewrite (4.3) as

$$v(s) = (1 - \alpha) \sum_{t=0}^{T-1} \alpha^t \mathbb{E}[c(W_t)|W_0 = s] + O\left(\|c\|_\infty \alpha^T\right). \tag{4.4}$$

Here the $O(\|c\|_\infty \alpha^T)$ bias can be made small if T is chosen large, and the first term can be estimated by simulating length-T trajectories. Specifically, let $\{W_t^{s,i}\}_{t=0}^{T-1}$ be a trajectory obtained as follows: set $W_0^{s,i} = s$ and, for $t \in \{1, \dots, T-1\}$, sample $W_t^{s,i}$ from $Q(W_{t-1}^{s,i}, \cdot)$. Letting $m \in \mathbb{N}$ and repeating this for each $i \in \{1, \dots, m\}$, we obtain an unbiased estimate of the first term in (4.4), namely $\frac{1}{m} \sum_{i=1}^{m} (1-\alpha) \sum_{t=0}^{T-1} \alpha^t c(W_t^{s,i})$. Repeating this across s yields an estimate of v, and for large m one can show $\|\hat{v} - v\|$ is small with high probability. We

discuss this more in Section 4.2.4; for now, we simply note that this approach *fundamentally requires* $\Omega(S)$ *samples*, as we must simulate trajectories beginning at each states.

At a high level, this approach explores the T-step outgoing neighborhood of each state s, i.e. those states reached with positive probability within T steps when the chain starts at s. We refer to this as *forward exploration*, as it is roughly analogous to conducting a T-step breadth-first-search forward (i.e. along outgoing edges in the graph induced by Q) from each state. Note that when Q is known, estimating v by computing the first T powers of Q involves a similar breadth-first-search. But when Q is known, it is more *computationally* efficient to use the power iteration: initialize $\hat{v}_0 = (1-\alpha)c$ and, given $\hat{v}_{t-1}$, set $\hat{v}_t = (1-\alpha)c + \alpha Q\hat{v}_{t-1}$, so that $\hat{v}_t = (1-\alpha)\sum_{\tau=0}^{t} \alpha^\tau Q^\tau c$. Conceptually, this is a *backward exploration* approach, i.e. exploration along incoming edges. This is most obvious when c has a single nonzero entry $c(s^*)$: we begin at s^* ($(1-\alpha)c$ term), then discover the incoming neighbors of s^* ($(1-\alpha)\alpha Qc$ term), then discover states two steps away from s^* ($(1-\alpha)\alpha^2 Q^2 c$ term), etc.

Motivated by the observation that backward exploration is more *computationally efficient* when Q is known, we will devise analogues for the case where Q is unknown, in hopes that these will be more *sample-efficient* than the forward exploration-based approach from [19]. Of course, computational complexity and sample complexity are in general very different creatures, but in our problem they are intuitively related owing to their conceptual connections to breadth-first-search discussed in the previous paragraph. We also note that several works – see e.g. [91, 92, 93, 94] – have recognized the advantages of the backward exploration approach, but these works have only studied the approach empirically. Thus, another motivation of this chapter is to add theoretical grounding to this line of work.

4.1.2 The supergraph

There is, however, a fundamental issue with our approach: backward exploration requires us to understand the *columns* of Q; in contrast, we can only sample from *rows* of Q. Of course, we can estimate all columns by estimating all rows, but then we incur $\Omega(S)$ sample complexity as in the existing approach. To overcome this issue, we will assume more is known about the underlying MDP: in addition to sampling from Q, we assume the algorithm is given a binary matrix A satisfying the "absolute continuity" condition

$$A(s,s') = 0 \Rightarrow Q(s,s') = 0 \ \forall \ s,s' \in \mathcal{S}. \tag{4.5}$$

Note we can view A as the adjacency matrix for a graph whose edges are a superset of those in the graph induced by Q; thus, we refer to this side information as the *supergraph*.

In words, the supergraph tells us that one-step transitions cannot occur between certain (ordered) pairs of states. For instance, if the MDP models a robot moving through an

environment, known physical limitations may prevent one-step transitions between states corresponding to significantly different locations, speeds, etc. As another example, if the MDP models a game, the game's rules may prevent transitions between certain pairs of states. Thus, in principle, domain knowledge can be used to construct such a supergraph. Hence, we believe our supergraph assumption is reasonable in many RL applications of interest. We emphasize that the reverse of the implication in (4.5) need not hold, i.e. we allow pairs s, s' for which $A(s, s') = 1$ but $Q(s, s') = 0$. Put differently, we do not require exact knowledge of the sparsity pattern of Q. Of course, there is a trade-off: our algorithms are more sample-efficient when A is sparser; thus, while one can always set $A(s, s') = 1 \; \forall \; s, s'$ to ensure that (4.5) holds, this will typically increase sample complexity.

In this chapter, we assume the supergraph is given, and we investigate how certain features of the supergraph (e.g. sparsity) impact the performance of our algorithms. An important practical consideration is how to actually obtain the supergraph; this problem is application-dependent and one we do not address. However, we do note that in applications like those of the previous paragraph, one can likely obtain policy-independent supergraphs: for instance, regardless of the action the robot takes from its current state (i.e. its policy), it cannot transition to a new state that corresponds to a significantly higher speed. Thus, in many applications, we believe one can construct a *single* supergraph, and then use it to evaluate *many* policies. For example, in the empirical policy iteration algorithm from [19] mentioned above, we would incur a one-time, offline expense to construct a supergraph that would then be used for the duration of the algorithm. In principle, this would be much more efficient than constructing a new supergraph at each policy improvement step.

4.1.3 Overview of chapter

Having motivated our approach, we give a brief summary of the chapter. As alluded to above, we devise algorithms for empirical policy evaluation (EPE), i.e. estimators for the value function (4.3) using the supergraph and samples from Q. Our first algorithm, `Backward-EPE`, is based on the idea of backward exploration and the power iteration. But in fact, `Backward-EPE` is more closely related to the `Approx-Contributions` algorithm [32] discussed in Section 2.3, which estimates (4.3) when Q is known and c has a single nonzero entry. We generalize this algorithm to the case $c \in \mathbb{R}_+^S$ and unknown Q; in the empirical dynamic programming spirit, we replace terms involving Q with empirical estimates.

Our analysis shows the *worst-case* sample complexity of `Backward-EPE` is $O(S \log S)$, which is the *best-case* complexity of the forward approach from [19] (assuming an l_∞ accuracy guarantee). We also show that when averaging over a certain class of cost vectors, `Backward-EPE` has expected complexity $O(\bar{d}(\|c\|_1/\|c\|_\infty) \log S\})$, where $\bar{d} = \frac{1}{S} \sum_{s,s'=1}^{S} A(s, s')$

is the average degree in the supergraph. Note $\bar{d}\|c\|_1/\|c\|_\infty = O(1)$ can occur, and thus *`Backward-EPE` can offer dramatic improvements over the forward approach, reducing sample complexity from $O(S \log S)$ to as low as $O(\log S)$.* In general, this average case result suggests `Backward-EPE` reduces sample complexity when the supergraph is sparse (so that $\bar{d}$ is small) and there are few high-cost states (so that $\|c\|_1/\|c\|_\infty$ is small).

Our second algorithm, `Bidirectional-EPE`, is inspired by the `Bidirectional-PPR` algorithm [15] discussed in Section 2.3; perhaps unsurprisingly, `Bidirectional-EPE` combines `Backward-EPE` with the forward exploration approach. This algorithm is less suited to the l_∞ guarantee that we establish for `Backward-EPE` and that is used in [19]; instead, we show `Backward-EPE` is conducive to a relative-plus-absolute error bound. Owing to this, it is more natural to compare `Bidirectional-EPE` to a plug-in estimator, wherein one estimates v by replacing Q with an empirical estimate in (4.3). Our analysis suggests `Bidirectional-EPE` is more sample-efficient than this plug-in estimator whenever the *average* degree in the supergraph is comparable to the *maximum* degree in the graph induced by Q.

Analytically, one of the main contributions of this chapter is an approach for analyzing power iteration variants like `Approx-Contributions` in the setting where Q is unknown. Analyzing such algorithms in this setting is difficult because the existing analysis relies on the invariant (2.2) from Section 2.3, which is in terms of PPR vectors defined on Q. Since we replace Q with empirical estimates in the algorithm, the invariant fails in the current setting, and with it the existing analysis. However, we show the invariant *does* hold if Q is replaced by any of a large set of matrices related to the estimate of Q generated during `Backward-EPE` (see Lemma 4.1). While we focus on two specific algorithms, we believe this analytical approach is applicable to other settings; Section 4.4 discusses some examples.

Notational conventions for the chapter: The following notation is often used in this chapter. For a matrix B and any $t \in \mathbb{N}$, we let $B^t(s, s')$, $B^t(s, \cdot)$, and $B^t(\cdot, s')$ denote the (s, s')-th entry, s-th row, and s'-th column of B^t, respectively. We write $0_{n\times m}$ and $1_{n\times m}$ for the $n \times m$ matrices of zeroes and ones, respectively. Matrix transpose is denoted by $^\intercal$. We use $1(\cdot)$ for the indicator function, i.e. $1(E) = 1$ if statement E is true and $1(E) = 0$ otherwise. For $s \in \mathcal{S}$, e_s is the length-S column vector with 1 in the s-th entry and 0 elsewhere, i.e. $e_s(s') = 1(s = s')$. Also for $s \in \mathcal{S}$, $N_{in}(s) = \{s' \in \mathcal{A} : A(s', s) = 1\}$ and $d_{in}(s) = |N_{in}(s)|$ are the incoming neighbors and in-degree of s in the supergraph. Average degree in the supergraph is denoted by $\bar{d} = \sum_{s,s'=1}^{S} A(s, s')/S = \sum_{s=1}^{S} d_{in}(s)/S$. We use the following (standard) notation for $\{a_n\}_{n\in\mathbb{N}}, \{b_n\}_{n\in\mathbb{N}} \subset [0, \infty)$: $a_n = O(b_n)$, $a_n = \Omega(b_n)$, $a_n = \Theta(b_n)$, and $a_n = o(b_n)$, resp., mean $\limsup_{n\to\infty} a_n/b_n < \infty$, $\liminf_{n\to\infty} a_n/b_n > 0$, $a_n = O(b_n)$ and $a_n = \Omega(b_n)$, and $\lim_{n\to\infty} a_n/b_n = 0$, resp. All random variables are defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with $\mathbb{E}[\cdot] = \int_\Omega \cdot \, d\mathbb{P}$ denoting expectation and *a.s.* meaning $\mathbb{P}$-almost surely.

4.2 Backward empirical policy evaluation

4.2.1 PageRank contributions and policy evaluation

We begin by restating some PPR ideas from Chapter II in the notation of this chapter and clarifying the connection between PageRank and policy evaluation. To avoid confusion with the policy notation of Section 4.1, we write $\mu_s = (1-\alpha)e_s^\mathsf{T}(I-\alpha Q)^{-1}$ for the s-th primitive PPR vector in this chapter (also recall the roles of α and $1-\alpha$ are reversed). Note that in the special case $c = e_{s^*}$ for some $s^* \in \mathcal{S}$, $v(s) = \mu_s c = \mu_s(s^*)$, so estimating the value function (4.3) amounts to estimating the s^*-th column of $(1-\alpha)(I-\alpha Q)^{-1}$. This is precisely the task accomplished by the `Approx-Contributions` algorithm. For the general case $c \in \mathbb{R}_+^S$, we can simply initialize the residual vector in `Approx-Contributions` as the cost vector; we define this scheme formally in Algorithm 4.1.

Algorithm 4.1: `Approx-Contributions` [32]

Input: *Transition matrix Q; cost c; discount factor α; termination parameter ε*
$k = 0$, $\hat{v}_k = 0_{S\times 1}$, $r_k = c$
while $\|r_k\|_\infty > \varepsilon$ **do**
  $k \leftarrow k+1$, $s_k \sim \arg\max_{s\in\mathcal{S}} r_{k-1}(s)$ uniformly
  for $s \in \mathcal{S}$ **do**
    if $s = s_k$ **then** $\hat{v}_k(s) = \hat{v}_{k-1}(s) + (1-\alpha)r_{k-1}(s)$, $r_k(s) = \alpha Q(s,s_k)r_{k-1}(s_k)$;
    else $\hat{v}_k(s) = \hat{v}_{k-1}(s)$, $r_k(s) = r_{k-1}(s) + \alpha Q(s,s_k)r_{k-1}(s_k)$;
Output: *Estimate $\hat{v}_k$ of $v = (1-\alpha)\sum_{t=0}^\infty \alpha^t Q^t c$*

In the case where Q is known, the accuracy analysis from [32] immediately extends to this general cost vector initialization. In particular, one can use the proof of [32] to show

$$\hat{v}_k(s) + \mu_s r_k = v(s) \;\forall\; k \in \{0,1,\ldots\}, s \in \mathcal{S} \tag{4.6}$$

(see also Remark 4.1 below). Thus, letting $k_* = \min\{k \in \mathbb{Z}_+ : \|r_k\|_\infty \le \varepsilon\}$ denote the iteration at which `Approx-Contributions` terminates, we have

$$|\hat{v}_{k_*}(s) - v(s)| \le \sum_{s'=1}^S \mu_s(s')r_{k_*}(s') \le \varepsilon \sum_{s'=1}^S \mu_s(s') = \varepsilon.$$

i.e. $\hat{v}_{k_*}(s)$ is an ε-accurate estimate of $v(s)$. Thus, the fact that `Approx-Contributions` produces an accurate estimate is fundamentally due to (4.6); since this equation relies on Q (through v and μ_s), one of our challenges will be to adapt this to the case of unknown Q.

At this point, it may not be clear why `Approx-Contributions` is preferable to the basic power iteration. To explain why it can be, first note that while we defined a *sequence* of vector

pairs $\{\hat{v}_k, r_k\}_{k=0}^{k_*}$ in Algorithm 4.1 for notational clarity, in practice one would iteratively update the *same* vector pair $\hat{v}, r$, i.e. one would overwrite the old values of $\hat{v}(s), r(s)$ with the new values per Lines 6-7 *if* these values change (for values that do not change, no update need occur). As an example, consider the problem instance

$$Q(s,s') = \begin{cases} 1, & s = 1, s' = S \text{ or } s > 1, s' = s-1 \\ 0, & \text{otherwise} \end{cases}, \quad c(s) = \begin{cases} 2\varepsilon, & s = 1 \\ (1-3\alpha)\varepsilon, & s \neq 1 \end{cases}.$$

Here we initialize $\hat{v} = 0_{S\times 1}$, $r = c$. At the first iteration, we choose $s_1 = 1$ and update $\hat{v}(1) = 2\varepsilon(1-\alpha)$, $r(1) = 0$, and $r(2) = (1-\alpha)\varepsilon$. After this update, $\|r\|_\infty = r(2) = (1-\alpha)\varepsilon < \varepsilon$, so the algorithm terminates. Note we only updated $\hat{v}(1)$, $r(1)$, and $r(2)$, and thus the computational complexity is $O(1)$. In contrast, a single power iteration, i.e. computation of Qc, incurs $\Omega(S)$ complexity. More generally, [32, Theorem 1] and [62, Theorem 2] provide computational complexity results; see Remark 4.4 below. For now, we only mention that, like the accuracy guarantee, these complexity results fundamentally rely on the invariant (4.6). Thus, our challenge will again be extending to the setting of unknown Q.

4.2.2 Algorithm

We now devise our first algorithm, `Backward-EPE`, by adapting `Approx-Contributions` to the EPE-with-supergraph setting. To explain our exact implementation, we begin with the first iteration of `Approx-Contributions`. Note here we only require knowledge of $Q(\cdot, s_1)$. Using the supergraph and samples from Q, we estimate this column as follows:

- $\forall\ s \in N_{in}(s_1)$, (4.5) ensures $Q(s, s_1) = 0$, so estimate such entries as zero.
- $\forall\ s \notin N_{in}(s_1)$, let $\{X_{s,i}\}_{i=1}^n \sim Q(s,\cdot)$ and estimate $Q(s, s_1)$ as $\sum_{i=1}^n 1(X_{s,i} = s_1)/n$.

Note this approach yields an unbiased estimate of $Q(\cdot, s_1)$ (and for large n this estimate will concentrate around $Q(\cdot, s_1)$). Now at future iterations, we could proceed in the exact same manner as the first iteration; however, this may be sample-inefficient. In particular, if $s \in N_{in}(s_1)$ and $s \in N_{in}(s_k)$ at some later iteration $k > 1$, we have already taken n samples from $Q(s,\cdot)$, so taking more samples from $Q(s,\cdot)$ to estimate $Q(s, s_k)$ is wasteful. Thus, we will instead reuse the samples taken at the first iteration to estimate $Q(s, s_k)$. (In Section 4.4.3, we discuss the relative merits of reusing samples versus resampling in more detail.)

To make this more concrete, we refer to the formal definition of `Backward-EPE`, Algorithm 4.2. In addition to the estimate and residual vectors from `Approx-Contributions`, we iteratively update a set U_k that tracks the states we have encountered up to and including iteration k and a matrix $\hat{Q}_k$ that represents our estimate of Q at iteration k, initialized to $U_0 = \emptyset, \hat{Q}_0 = 0_{S\times S}$. At the k-th iteration, we sample a high-residual state s_k as in `Approx-Contributions` and then use U_{k-1} and the supergraph to determine which

incoming neighbors of s_k have not yet been encountered (Line 4 of Algorithm 4.2), i.e. for which neighbors s we have not yet estimated $Q(s, \cdot)$. For such neighbors, we estimate $Q(s, \cdot)$ via samples from $Q(s, \cdot)$ (Line 7); for other states s, our estimate of $Q(s, \cdot)$ remains unchanged (Line 8). We then update the estimate and residual vectors in the same manner as `Approx-Contributions` but using the empirical estimate of Q. As in `Approx-Contributions`, this sequence of updates continues until $\|r_k\|_\infty \leq \varepsilon$. We re-emphasize that our initial estimate of Q is $\hat{Q}_0 = 0_{S \times S}$ and we gradually fill the rows of $\hat{Q}_k$ with estimates as we encounter more states; once we fill a row $\hat{Q}_k(s, \cdot)$ it remains unchanged for the remainder of the algorithm.

Algorithm 4.2: `Backward-EPE`

1 **Input:** *Sampler for transition matrix Q; cost vector c; discount factor α; supergraph in-neighbors $\{N_{in}(s)\}_{s=1}^S$; termination parameter ε; per-state sample count n*
2 $k = 0$, $\hat{v}_k = 0_{S \times 1}$, $r_k = c$, $U_k = \emptyset$, $\hat{Q}_k = 0_{S \times S}$
3 **while** $\|r_k\|_\infty > \varepsilon$ **do**
4 $\quad k \leftarrow k + 1$, $s_k \sim \arg\max_{s \in \mathcal{S}} r_{k-1}(s)$ uniformly, $U_k = U_{k-1} \cup N_{in}(s_k)$
5 $\quad$ // $\hat{Q}$ update loop
6 $\quad$ **for** $s \in \mathcal{S}$ **do**
7 $\quad\quad$ **if** $s \in N_{in}(s_k) \setminus U_{k-1}$ **then** $\{X_{s,i}\}_{i=1}^n \sim Q(s, \cdot)$, $\hat{Q}_k(s, \cdot) = \frac{1}{n} \sum_{i=1}^n 1(X_{s,i} = \cdot)$;
8 $\quad\quad$ **else** $\hat{Q}_k(v, \cdot) = \hat{Q}_{k-1}(v, \cdot)$;
9 $\quad$ // $\hat{v}, r$ update loop
10 $\quad$ **for** $s \in \mathcal{S}$ **do**
11 $\quad\quad$ **if** $s = s_k$ **then** $\hat{v}_k(s) = \hat{v}_{k-1}(s) + (1 - \alpha) r_{k-1}(s)$, $r_k(s) = \alpha \hat{Q}_k(s, s_k) r_{k-1}(s_k)$;
12 $\quad\quad$ **else** $\hat{v}_k(s) = \hat{v}_{k-1}(s)$, $r_k(s) = r_{k-1}(s) + \alpha \hat{Q}_k(s, s_k) r_{k-1}(s_k)$;
13 **Output:** *Estimate $\hat{v}_k$ of $v = (1 - \alpha) \sum_{t=0}^\infty \alpha^t Q^t c$*

4.2.3 Analysis

As discussed in Section 4.2.2, the main analytical difficulty is that $\hat{v}_k, r_k$ need not satisfy (4.6) when these vectors are generated via `Backward-EPE`. To overcome this issue, we begin with the key observation mentioned in Section 4.1.3: while the invariant need not hold for the *actual* transition matrix Q, it does hold for a class of matrices derived from the *estimated* transition matrix $\hat{Q}_{k_*}$. More specifically, we will show that (4.6) holds if we replace $\mu_s(s')$ and $v(s)$ with analogous quantities defined in terms any matrix P satisfying two key properties: P should contain the estimates of Q generated during `Backward-EPE`, and P should satisfy the absolute continuity condition (4.5). Note that if we encounter *all* states during the algorithm, i.e. if $U_{k_*} = \mathcal{S}$, then only $\hat{Q}_{k_*}$ satisfies these properties; however, if U_{k_*} is a strict subset of $\mathcal{S}$, many choices of P will satisfy these properties (i.e. we can fill the unestimated rows of $\hat{Q}_{k_*}$ with any entries satisfying (4.5)). This is formalized by Lemma 4.1.

Lemma 4.1. Let $\mathcal{P} = \{B \in \mathbb{R}_+^{S\times S} : \sum_{s'=1}^{S} B(s,s') = 1 \ \forall \ s \in \mathcal{S}\}$ denote the set of $S \times S$ row stochastic matrices, and let P be a $\mathcal{P}$-valued random matrix satisfying the following:

$$P(s,\cdot) = \hat{Q}_{k_*}(s,\cdot) \ \forall \ s \in U_{k_*} \ a.s., \quad A(s,s') = 0 \Rightarrow P(s,s') = 0 \ \forall \ s,s' \in \mathcal{S} \ a.s.$$

For each $s \in \mathcal{S}$, let $\nu_s = (1-\alpha)e_s^{\mathsf{T}}(I-\alpha P)^{-1}$ and $u(s) = \nu_s c$ denote the PPR vector and value function on P. Then the vectors $\{\hat{v}_k, r_k\}_{k=0}^{k_*}$ from Algorithm 4.2 satisfy the following:

$$\hat{v}_k(s) + \nu_s r_k = u(s) \ \forall \ k \in \{0,\ldots,k_*\}, s \in \mathcal{S} \ a.s. \tag{4.7}$$

Proof. Fix $s \in \mathcal{S}$. We prove (4.7) by induction. For $k = 0$, (4.7) is immediate, since $\hat{v}_0 = 0_{S\times 1}$, $r_0 = c$ in Algorithm 4.2. For $k \in [k_*]$, Lines 11-12 of Algorithm 7 imply $(a.s.)$

$$\begin{aligned}\hat{v}_k(s) + \nu_s r_k &= \hat{v}_{k-1}(s) + (1-\alpha)r_{k-1}(s_k)1(s = s_k) \\ &\quad + \sum_{s'=1}^{S} \nu_s(s')(r_{k-1}(s')1(s' \neq s_k) + \alpha\hat{Q}_k(s',s_k)r_{k-1}(s_k)) \\ &= \hat{v}_{k-1}(s) + \nu_s r_{k-1} + r_{k-1}(s_k)(-\nu_s(s_k) + (1-\alpha)1(s=s_k) + \alpha\nu_s\hat{Q}(\cdot,s_k)),\end{aligned} \tag{4.8}$$

where for the second equality we added and subtracted $\mu_s(s_k)r_{k-1}(s_k)$. Now since $\hat{v}_{k-1}(s) + \nu_s r_{k-1} = u(s)$ $a.s.$ by the inductive hypothesis, and since

$$\nu_s(s_k) - (1-\alpha)1(s=s_k) = \alpha(1-\alpha)\sum_{t=0}^{\infty}\alpha^t P^t(s,\cdot)P(\cdot,s_k) = \alpha\nu_s P(\cdot,s_k), \tag{4.9}$$

it suffices to show $\hat{Q}_k(s',s_k) = P(s',s_k) \ \forall \ s' \in \mathcal{S}$ $a.s.$ (since then the term in parentheses in (4.8) will be zero). Toward this end, we fix $s' \in \mathcal{S}$ and consider two cases:

- If $s' \in U_k$, Algorithm 4.2 implies $\hat{Q}_k(s',s_k) = \hat{Q}_{k_*}(s',s_k)$. Moreover, $U_k \subset U_{k_*}$ in Algorithm 4.2, so $s' \in U_{k_*}$, and thus $P(s',s_k) = \hat{Q}_{k_*}(s',s_k)$ $a.s.$ by assumption on P. Taken together, $\hat{Q}_k(s',s_k) = P(s',s_k)$ $a.s.$
- If $s' \notin U_k$, Algorithm 4.2 implies $Q_k(s',s_k) = 0$. On the other hand, $N_{in}(s_k) \subset U_k$, (Line 4 of Algorithm 4.2) so $s' \notin N_{in}(s_k)$ and $A(s',s_k) = 0$ by definition of $N_{in}(s_k)$. Hence, by assumption on P, we have $P(s',s_k) = 0$ $a.s.$ as well.

Thus, $\hat{Q}_k(s',s_k) = P(s',s_k)$ $a.s.$ in both cases, completing the proof. □

Remark 4.1. The `Approx-Contributions` invariant (4.6) is proven in a similar (but simpler) manner: assuming (4.6) holds for $k-1$, one proves it holds for k using the approach of (4.8) and (4.9) (replacing ν_s with μ_s and both $P, \hat{Q}_k$ with Q).

Lemma 4.1 allows us to apply the invariant (4.7) to (potentially) many matrices P. In this chapter, we only use two (somewhat obvious) choices of P. For the first choice, we fill rows of $\hat{Q}_{k_*}$ that were *not* estimated during the algorithm with independent estimates generated offline. More precisely, for each $s \in \mathcal{S}$ and each $i \in [n]$, let $Y_{s,i} \sim Q(s, \cdot)$, independent across s and i, and independent of all the random variables generated by Algorithm 4.2. From $\{Y_{s,i}\}_{s \in \mathcal{S}, i \in [n]}$, define an offline estimate $\tilde{Q}$ of Q row-wise by

$$\tilde{Q}(s, \cdot) = \frac{1}{n} \sum_{i=1}^{n} 1(Y_{s,i} = \cdot). \tag{4.10}$$

From $\tilde{Q}$, we define our first choice of P and the corresponding value function by

$$\overline{Q}(s, \cdot) = \begin{cases} \hat{Q}_{k_*}(s, \cdot), & s \in U_{k_*} \\ \tilde{Q}(s, \cdot), & s \in \mathcal{S} \setminus U_{k_*} \end{cases}, \overline{\mu_s} = (1-\alpha) e_s^\mathsf{T} (I - \alpha \overline{Q})^{-1}, \overline{v}(s) = \overline{\mu_s} c \ \forall \ s \in \mathcal{S}. \tag{4.11}$$

For the second choice, we fill unestimated rows by the actual rows of Q, i.e. we let

$$\underline{Q}(s, \cdot) = \begin{cases} \hat{Q}_{k_*}(s, \cdot), & s \in U_{k_*} \\ Q(s, \cdot), & s \in \mathcal{S} \setminus U_{k_*} \end{cases}, \underline{\mu_s} = (1-\alpha) e_s^\mathsf{T} (I - \alpha \underline{Q})^{-1}), \underline{v}(s) = \underline{\mu_s} c \ \forall \ s \in \mathcal{S}. \tag{4.12}$$

Note $\overline{Q} = \underline{Q} = \hat{Q}_{k_*}$ if $U_{k_*} = \mathcal{S}$. Also note $\overline{Q}$ and $\underline{Q}$ satisfy the assumptions of Lemma 4.1, so

$$\hat{v}_k(s) + \overline{\mu_s} r_k = \overline{v}(s), \quad \hat{v}_k(s) + \underline{\mu_s} r_k = \underline{v}(s) \quad \forall \ k \in \{0, \ldots, k_*\}, s \in \mathcal{S} \ a.s. \tag{4.13}$$

We will refer to the identities in (4.13) as the $\overline{Q}$-*invariant* and the $\underline{Q}$-*invariant*, respectively. For clarity, we hereafter refer to (4.6) as the Q-*invariant*. The $\overline{Q}$- and $\underline{Q}$-invariants will be crucial tools in our proofs; however, typically these proofs will only work when using one of the two invariants (see Remarks 4.2, 4.3, and 4.5).

Equipped with Lemma 4.1, we can prove our first main result concerning `Backward-EPE`, Theorem 4.1. The theorem provides a lower bound on the per-state sample count n to ensure the ultimate estimate $\hat{v}_{k_*}$ is 2ε-close to v in the l_∞ norm with high probability. Here 2ε arises because we have two sources of error: the ε-bounded residual and the fact that the Q-invariant fails. Of course, we could bound both errors by $\varepsilon/2$ with only worse constants for the per-state sample count; however, for later analysis, it will be more convenient to have residual error bounded by ε instead of $\varepsilon/2$ (i.e. to not carry the $1/2$ factor).

Theorem 4.1. Fix $\varepsilon, \delta > 0$ and define

$$n^*(\varepsilon, \delta) = \frac{2\|c\|_\infty^2 \alpha^2}{\varepsilon^2 (1-\alpha)^2} \log\left(\frac{2S}{\delta}\left\lceil \frac{\log(4\|c\|_\infty/\varepsilon)}{1-\alpha}\right\rceil\right).$$

Then assuming $n \geq n^*(\varepsilon, \delta)$ in Algorithm 4.2, we have

$$\mathbb{P}(\|\hat{v}_{k_*} - v\|_\infty \geq 2\varepsilon) \leq \delta. \tag{4.14}$$

Proof sketch. The full proof is deferred to Appendix C.1 but we sketch it here. First, by the $\overline{Q}$-invariant (4.13), the triangle inequality, and the termination criteria of `Backward-EPE`,

$$\|\hat{v}_{k_*} - v\|_\infty \leq \|\hat{v}_{k_*} - \overline{v}\|_\infty + \|\overline{v} - v\|_\infty \leq \varepsilon + \|\overline{v} - v\|_\infty,$$

so our task is reduced to showing $\|\overline{v} - v\|_\infty \leq \varepsilon$ with high probability, i.e. that $\overline{v}$ concentrates around v. It is reasonable to expect this to hold for large n, since $\overline{v}$ and v are defined in terms of $\overline{Q}$ and Q, respectively, and since $\overline{Q} \approx Q$ when n is large. However, this concentration is not immediate, in part because $\overline{v}$ need not be an unbiased estimate of v. Thus, most of the proof involves estimating $\|\overline{v} - v\|_\infty$ by an upper bound that is more amenable to concentration inequalities, i.e. the deviation of an empirical average from its true mean. Here the key steps are as follows. First, it is straightforward to show that for large enough T,

$$\|\overline{v} - v\|_\infty \leq (1-\alpha) \sum_{t=1}^{T-1} \alpha^t \|(\overline{Q}^t - Q^t)c\|_\infty + \frac{\varepsilon}{2}. \tag{4.15}$$

Second, using convexity of $\|\cdot\|_\infty$ and row stochasticity of $\overline{Q}$, a simple calculation yields

$$\|(\overline{Q}^t - Q^t)c\|_\infty \leq \|(\overline{Q}^{t-1} - Q^{t-1})c\|_\infty + \|(\overline{Q} - Q)Q^{t-1}c\|_\infty.$$

Iterating this inequality and substituting into (4.15) gives a bound on $\|\overline{v} - v\|_\infty$ in terms of $\|(\overline{Q} - Q)Q^{t-1}c\|_\infty$. Furthermore, by definition this latter quantity has the same distribution as $\|(\tilde{Q} - Q)Q^{t-1}c\|_\infty$, so we can bound $\|\overline{v} - v\|_\infty$ in terms of $\|(\tilde{Q} - Q)Q^{t-1}c\|_\infty$ (see Remark 4.2). Finally, defining $d_{t-1} = Q^{t-1}c$, the s-th entry of $\tilde{Q}Q^{t-1}c$ is

$$\sum_{s'=1}^{S} \tilde{Q}(s, s') d_{t-1}(s') = \sum_{s'=1}^{S} \left(\frac{1}{n}\sum_{i=1}^{n} 1(Y_{s,i} = s')\right) d_{t-1}(s') = \frac{1}{n}\sum_{i=1}^{n} d_{t-1}(Y_{s,i}),$$

and similarly, the s-th entry of $QQ^{t-1}c$ is $\mathbb{E}d_{t-1}(Y_{s,i})$. Therefore,

$$\|(\tilde{Q}-Q)Q^{t-1}c\|_\infty = \max_{s\in\mathcal{S}} \left| \frac{1}{n}\sum_{i=1}^{n}(d_{t-1}(Y_{s,i}) - \mathbb{E}d_{t-1}(Y_{s,i}))\right|,$$

so that $\|(\tilde{Q}-Q)Q^{t-1}c\|_\infty$ is the deviation of an empirical average around its mean, as desired. The proof is completed using standard Chernoff bounds. □

Remark 4.2. It may seem wasteful that we use the $\overline{Q}$-invariant instead of the $\underline{Q}$-invariant for Theorem 4.1, since $\underline{Q}$ fills unestimated rows of $\hat{Q}_{k_*}$ with the actual rows of Q, and thus $\underline{v}$ should be a better estimate of v. We explain this choice as follows. First note that by the arguments in the proof sketch, bounding $\|\underline{v}-v\|_\infty$ amounts to bounding $\|(\underline{Q}-Q)Q^{t-1}c\|_\infty$. It is tempting to use the union bound to bound such terms as

$$\mathbb{P}\left(\|(\underline{Q}-Q)Q^{t-1}c\|_\infty \geq \eta \middle| U_{k_*}\right) \leq \sum_{s\in U_{k_*}} \mathbb{P}\left(\frac{|\sum_{i=1}^{n}(d_{t-1}(X_{s,i}) - \mathbb{E}d_{t-1}(X_{s,i}))|}{n} \geq \eta \middle| U_{k_*}\right).$$

The issue with this approach is that there is a complicated dependence between $\{X_{s,i}\}_{i=1}^n$ and U_{k_*} in Algorithm 4.2. We also note that we replace $\|(\overline{Q}-Q)Q^{t-1}c\|_\infty$ by $\|(\tilde{Q}-Q)Q^{t-1}c\|_\infty$ in the proof of Theorem 4.1 owing to a similar issue.

Theorem 4.1 says that if we take $n \geq n^*(\varepsilon,\delta)$ samples from $Q(s,\cdot)$ in Line 7 of Algorithm 4.2, the ultimate estimate $\hat{v}_{k_*}$ will be 2ε-accurate. Hence, the total number of samples needed to ensure 2ε-accuracy is $n^*(\varepsilon,\delta)$ multiplied by the number of times Line 7 is reached; by definition, this latter quantity is $|U_{k_*}|$. Our next goal is thus to bound $|U_{k_*}|$, in order to bound the overall sample complexity of `Backward-EPE`. Before presenting our result, we develop some intuition regarding the behavior of $|U_{k_*}|$. First, it is clear that $|U_{k_*}| \leq S$. Moreover, this upper bound can be attained in certain cases, for example:

- Suppose $\min_{s\in\mathcal{S}} c(s) \geq \varepsilon$. Then $\forall\ s \in \mathcal{S}$, we have $s_k = s$ for some $k \in [k_*]$ (else, $r_{k_*}(s) \geq r_0(s) = c(s) \geq \varepsilon$, a contradiction). Thus, each $s \in \mathcal{S}$ will belong to $N_{in}(s_k)$ at some k, so $U_{k_*} = \cup_{k=1}^{k_*} N_{in}(s_k) = \mathcal{S}$.
- Suppose $A(s,s') = 1\ \forall\ s,s' \in \mathcal{S}$. Then $N_{in}(s_1) = \mathcal{S}$ by definition, so $U_{k_*} = \mathcal{S}$ as well.

While these examples are extreme cases, they suggest $|U_{k_*}|$ will be large if there are too many high-cost states or too many edges in the supergraph. Put differently, it seems $|U_{k_*}|$ *may be small if there are sufficiently few high-cost states and sufficiently few edges in the supergraph.* But even when both of these occur, one can construct adversarial examples for which $U_{k_*} = \mathcal{S}$. For instance, suppose we restrict to c having a single high-cost state and the supergraph to having the minimal number of edges possible, S. Then taking $c = [1\ 0\ \ldots\ 0]$,

$A = 1_{S\times 1}e_1^\mathsf{T}$ will satisfy this restriction, but will yield $U_{k_*} = \mathcal{S}$ (assuming $\varepsilon < 1$). The key issue in this example (and, we suspect, in most adversarial examples) is the interaction between the cost vector and the supergraph; in particular, if high-cost states have high in-degrees, $|U_{k_*}|$ will be large (even if there are few high-cost states and edges overall).

In summary, the sample complexity of `Backward-EPE` scales with $|U_{k_*}|$, which is intuitively small when there are few high-cost states and supergraph edges; however, even when both of these quantities are minimal, $|U_{k_*}|$ is maximal in the worst-case. Given this, our best hope for bounding $|U_{k_*}|$ is an average-case analysis; in particular, bounding $\mathbb{E}|U_{k_*}|$ while randomizing over the inputs of `Backward-EPE`. As it turns out, we only need to randomize over the cost vector (not Q). Roughly, we will consider a random cost vector C for which $\mathbb{E}C(s) = O(\mathbb{E}\|C\|_1/S)\ \forall\ s \in \mathcal{S}$, i.e. the expected cost of any given state does not dominate the average expected cost. For such cost vectors, the interaction between cost and in-degree discussed in the previous paragraph will "average out", and consequently the adversarial examples will not dominate in expectation. This is formalized in the following theorem.

Theorem 4.2. Let C be an $\mathbb{R}_+^S$-valued random vector s.t. $\mathbb{E}\|C\|_1 < \infty$ and $\max_{s\in\mathcal{S}} \mathbb{E}C(s) \leq \beta\mathbb{E}\|C\|_1/S =: \bar{c}$ for some constant $\beta \in [1, \infty)$. Then if Algorithm 4.2 is initialized with C,

$$\mathbb{E}|U_{k_*}| \leq \frac{S\bar{c}\bar{d}}{\varepsilon(1-\alpha)},$$

where the expectation is with respect to C and the randomness in Algorithm 4.2.

Proof. We use the $\overline{Q}$-invariant (4.13) (note we proved Lemma 4.1 for fixed c but the same arguments hold for random C owing to their almost-sure nature). First, for for any $s \in \mathcal{S}$,

$$\overline{v}(s) \geq \hat{v}_{k_*}(s) = (1-\alpha)\sum_{k=1}^{k_*} r_{k-1}(s)1(s = s_k) \geq \varepsilon(1-\alpha)\sum_{k=1}^{k_*} 1(s = s_k), \tag{4.16}$$

where the first inequality holds by the $\overline{Q}$-invariant (4.13), the equality by Lines 11-12 of Algorithm 4.2, and the second inequality by definition of k_*. On the other hand, we have

$$|U_{k_*}| = |\cup_{s=1}^{k_*} N_{in}(s_k)| \leq \sum_{k=1}^{k_*} d_{in}(s_k) = \sum_{k=1}^{k_*}\sum_{s=1}^{S} d_{in}(s)1(s = s_k) = \sum_{s=1}^{S} d_{in}(s)\sum_{k=1}^{k_*} 1(s = s_k).$$

Combining the previous two inequalities and taking expectation, we have therefore shown

$$\mathbb{E}|U_{k_*}| \leq \frac{1}{\varepsilon(1-\alpha)}\sum_{s=1}^{S} d_{in}(s)\mathbb{E}\overline{v}(s). \tag{4.17}$$

Now consider $\mathbb{E}\overline{v}(s)$. By definition (4.11),

$$\mathbb{E}\overline{v}(s) = \mathbb{E}\overline{\mu_s}C = (1-\alpha)\sum_{t=0}^{\infty}\alpha^t\mathbb{E}\overline{Q}^t(s,\cdot)C = (1-\alpha)\sum_{t=0}^{\infty}\alpha^t\mathbb{E}[\mathbb{E}[\overline{Q}^t(s,\cdot)|C]C].$$

Now after realizing C, we fill some rows of $\overline{Q}$ with samples generated during the algorithm and other rows with samples generated offline; in contrast, all rows of $\tilde{Q}$ are filled with offline samples. But in either case, these samples have the same distribution, so we can replace $\overline{Q}$ by $\tilde{Q}$ in the previous equation. Moreover, $\tilde{Q}$ is independent of the random variables in Algorithm 4.2, including $r_0 = C$. In summary,

$$\mathbb{E}[\overline{Q}^t(s,\cdot)|C] = \mathbb{E}[\tilde{Q}^t(s,\cdot)|C] = \mathbb{E}[\tilde{Q}^t(s,\cdot)]. \tag{4.18}$$

Combining the previous two equations and using the assumption on C, we obtain

$$\mathbb{E}\overline{v}(s) = (1-\alpha)\sum_{t=0}^{\infty}\alpha^t\mathbb{E}[\tilde{Q}^t(s,\cdot)]\mathbb{E}[C] \leq (1-\alpha)\sum_{t=0}^{\infty}\alpha^t\mathbb{E}[\tilde{Q}^t(s,\cdot)]\bar{c}1_{S\times 1} = \bar{c},$$

where we also used row stochasticity of $\tilde{Q}$. Substituting into (4.17) completes the proof. □

Remark 4.3. Note this approach fails if we use the $\underline{Q}$-invariant. In particular, we cannot express $\mathbb{E}[\underline{Q}^t(s,\cdot)|C]$ as deterministic in (4.18), since C influences which states are encountered during the algorithm and thus influences which rows of $\underline{Q}$ are estimates and which are exact. This illustrates the utility of the $\overline{Q}$-invariant: it allows us to "decorrelate" the estimated transition matrix from the cost vector, i.e. to obtain $\mathbb{E}[\overline{Q}^t(s,\cdot)C] = \mathbb{E}[\tilde{Q}^t(s,\cdot)]\mathbb{E}[C]$. In the current chapter, this is our only use of this decorrelation trick, but it may useful in analyses of algorithms like `Backward-EPE` (for example, those discussed in Section 4.4).

Remark 4.4. The preceding proof is similar that of Theorem 2 in [62], which considers the expected computational complexity of `Approx-Contributions` when $C \sim \{e_s\}_{s=1}^S$ uniformly. In fact, [62] uses the Q-invariant but otherwise follows the logic leading to (4.17); since μ_s is deterministic in the Q-invariant, one immediately obtains $\mathbb{E}v(s) = \mu_s\mathbb{E}C = \mu_s 1_{S\times 1}/S = 1/S$ in this case. Similarly, [32, Theorem 1] provides a bound on k_* for fixed c of the form $c = e_{s^*}$; the proof uses the Q-invariant and the logic of (4.16) to obtain $v(s) \geq \varepsilon(1-\alpha)\sum_{k=1}^{k_*} 1(s = s_k)$, then sums over s to obtain $k_* \leq \|v\|_1/(\varepsilon(1-\alpha))$. We note similar arguments are used in our analysis for Chapters II (see Appendix A.1-A.2) and III (see Appendix B.4).

4.2.4 Discussion

We now return to interpret our results for `Backward-EPE` and specifically to derive the algorithm's overall sample complexity, which by definition is $n^*(\varepsilon,\delta)|U_{k_*}|$. In the worst case,

$|U_{k*}| = \Omega(S)$, and thus the worst-case sample complexity for fixed c is $Sn^*(\varepsilon, \delta)$. Neglecting all $\log\log$ factors and all absolute constants, ignoring log terms for quantities that have polynomial scaling (e.g. writing $\log(1/(1-\alpha))/(1-\alpha)^2$ as simply $1/(1-\alpha)^2$), and assuming α is either constant or grows to 1, Theorem 4.1 implies

$$Sn^*(\varepsilon, \delta) = O\left(S \log(S/\delta) \|c\|_\infty^2 \varepsilon^{-2} (1-\alpha)^{-2}\right).$$

For comparison, the sample complexity of the approach from [19] in *any case* is

$$O\left(S \log(S/\delta) \|c\|_\infty^2 \varepsilon^{-2} (1-\alpha)^{-3}\right) \tag{4.19}$$

(see Appendix C.4). Thus, in the worst case `Backward-EPE` has similar complexity to the best case of the approach from [19], with a slightly improved dependence on the discount factor α. (The extra $(1-\alpha)$ factor in (4.19) arises since $O(1/(1-\alpha))$-length trajectories must be sampled to make the bias in (4.4) small.)

In the average case, the sample complexity of `Backward-EPE` can be dramatically better. In particular, by Theorem 4.2, we can bound the average-case sample complexity as

$$\mathbb{E}[|U_{k*}|] \times n^*(\varepsilon, \delta) = O\left(\frac{S\bar{c}\bar{d}}{\varepsilon(1-\alpha)} \times \frac{\log(S/\delta)\|C\|_\infty^2}{\varepsilon^2(1-\alpha)^2}\right) = O\left(\frac{\|C\|_1 \bar{d} \log(S/\delta)\|C\|_\infty^2}{\varepsilon^3(1-\alpha)^3}\right).$$

(This argument is not precise, since $\|C\|_\infty$ is random in Theorem 4.2; we return to address this shortly.) Thus, assuming α, δ, and $\|C\|_\infty/\varepsilon$ are constants, the expected complexity is

$$O\left((\|C\|_1/\|C\|_\infty) \times \bar{d} \times \log S\right). \tag{4.20}$$

Interestingly, (4.20) exactly captures the intuition discussed in Section 4.2.3: $\|C\|_1/\|C\|_\infty$ quantifies the intuition that `Backward-EPE` has low complexity if there are few high-cost states; $\bar{d}$ quantifies the intuition of low complexity if there are sufficiently few edges in the supergraph. We also note that when α, δ, and $\|C\|_\infty/\varepsilon$ are constants, the existing approach's complexity (4.19) becomes $O(S \log S)$. In the extreme case, $\|C\|_1/\|C\|_\infty$ and $\bar{d}$ are both $O(1)$, and thus `Backward-EPE` offers a dramatic reduction in sample complexity.

Though this average-case argument is not entirely precise, we can make it precise with further assumptions on C. For example, the following corollary considers random binary cost vectors with H nonzero entries. Such cost vectors could arise, for example, in simple MDP models of games, where states corresponding to losing configurations of the game have unit cost and other states have zero cost.

Corollary 4.1. Let $H \in \mathcal{S}$ and $\mathcal{C}_H = \{\sum_{s=1}^S a_s e_s : a_s \in \{0,1\} \ \forall \ s \in \mathcal{S}, \sum_{s=1}^S a_s = H\}$.

Assume the cost vector C is chosen uniformly at random from $\mathcal{C}_H$ and $\alpha, \delta, \varepsilon$ are constants. Then to guarantee an accurate estimate in the sense of (4.14), `Backward-EPE` requires $O(\min\{H\bar{d}, S\} \log S)$ samples in expectation.

Proof. Though we stated Theorem 4.1 in the case of a deterministic cost vector c, it also holds for C if the lower bound on n holds almost surely (see Remark C.1). Moreover, by assumption on C, $\|C\|_\infty = 1$ pointwise and thus $n^*(\varepsilon, \delta)$ is deterministic; paired with the assumption on $\alpha, \delta, \varepsilon$, we have $n^*(\varepsilon, \delta) = O(\log S)$. Thus, the expected sample complexity of `Backward-EPE` is $\mathbb{E}[|U_{k_*}| n^*(\varepsilon, \delta)] = O(\mathbb{E}[|U_{k_*}|] \log S)$. Again using the assumption on C, $\mathbb{E}C(s) = H/S \ \forall \ s \in \mathcal{S}$, so we can apply Theorem 4.2 with $\bar{c} = H/S$ to obtain $\mathbb{E}|U_{k_*}| = O(H\bar{d})$. Finally, since $U_{k_*} \subset \mathcal{S}$, we can sharpen this to obtain $\mathbb{E}|U_{k_*}| = O(\min\{H\bar{d}, S\})$. □

4.3 Bidirectional empirical policy evaluation

4.3.1 Algorithm

We next explain our second algorithm, which is derived from `Backward-EPE` in much the same way `FW-BW-MCMC` and `Bidirectional-PPR` are derived from `Approx-Contributions` (see Section 2.3). The main wrinkle is that the Q-invariant fails, so we will instead use the $\underline{Q}$-invariant. In particular, similar to Theorem 4.1, we can make $|\underline{v}(s) - v(s)|$ small if we take enough samples during `Backward-EPE`; when this holds, we have

$$v(s) \approx \underline{v}(s) = \hat{v}_{k_*}(s) + \underline{\mu_s} r_{k_*}. \tag{4.21}$$

Now since $\underline{\mu_s}$ is a probability distribution over $\mathcal{S}$, the residual term in (4.21) satisfies

$$\underline{\mu_s} r_{k_*} = \mathbb{E}_{Z_s \sim \underline{\mu_s}} r_{k_*}(Z_s) \approx \frac{1}{n_F} \sum_{i=1}^{n_F} r_{k_*}(Z_{s,i}),$$

where $\{Z_{s,i}\}_{i=1}^{n_F}$ are i.i.d. samples from $\underline{\mu_s}$ and n_F is large. Hence, by (4.21),

$$v(s) \approx \hat{v}_{k_*}(s) + \frac{1}{n_F} \sum_{i=1}^{n_F} r_{k_*}(Z_{s,i}). \tag{4.22}$$

Intuitively, the right side of (4.22) is a more accurate estimate of $v(s)$ than $\hat{v}_{k_*}(s)$ alone; the only remaining question is how to generate $\{Z_{s,i}\}_{i=1}^{n_F}$. For this, we exploit the perfect sampling property (1.4), restated in our current notation as $\mathbb{P}^{\underline{Q}}(Z_{s,i} = s') = \underline{\mu_s}(s')$, where $\mathbb{P}^{\underline{Q}}$ means probability conditioned on $\underline{Q}$ and $Z_{s,i}$ is the endpoint of a Geometric$(1-\alpha)$-length trajectory on $\underline{Q}$ beginning on s. Note we can indeed simulate trajectories on $\underline{Q}$, since sampling from $\underline{Q}(s, \cdot)$ amounts to sampling either from $Q(s, \cdot)$ (as in `Backward-EPE`) or from

$\hat{Q}_{k_*}(s,\cdot)$ (which is known after running `Backward-EPE`). Put differently, to generate $Z_{s,i}$ we sample from $Q(s,\cdot)$ *unless we have already sampled from* $Q(s,\cdot)$ during `Backward-EPE`, in which case we sample from the empirical estimate $\hat{Q}_{k_*}(s,\cdot)$ from `Backward-EPE`.

This procedure is formalized in Algorithm 4.3. As above, write n_F for the per-state forward trajectory count; we also write n_B for the per-state sample count in the `Backward-EPE` subrountine. We denote the ultimate estimate of v by $\hat{v}_{BD}$. Other than these changes, the notation is identical to that used for `Backward-EPE`.

Algorithm 4.3: `Bidirectional-EPE`

Input: *Sampler for transition matrix Q; cost vector c; discount factor α; supergraph in-neighbors $\{N_{in}(s)\}_{s=1}^{S}$; termination parameter ε; per-state backward, forward sample counts n_B, n_F*

// Backward exploration stage

Run `Backward-EPE` with inputs Q sampler, c, α, $\{N_{in}(s)\}_{s=1}^{S}$, ε, n_B

// Forward exploration stage

for $s \in \mathcal{S}$ **do**

  for $i = 1$ **to** n_F **do**

    // Generate sample $Z_{s,i}$ from $\underline{\mu_s}$

    $L_{s,i} \sim \text{Geometric}(1-\alpha)$, $Z_{s,i}^0 = s$

    for $t = 1$ **to** $L_{s,i}$ **do**

      $Z_{s,i}^t \sim \underline{Q}(Z_{s,i}^{t-1}, \cdot)$

    $Z_{s,i} = Z_{s,i}^{L_{s,i}}$

Output: *Estimate* $\hat{v}_{BD}(s) = \hat{v}_{k_*}(s) + \frac{1}{n_F}\sum_{i=1}^{n_F} r_{k_*}(Z_{s,i})$ *for each* $s \in \mathcal{S}$

4.3.2 Analysis

Since `Bidirectional-EPE` samples trajectories forward from each $s \in \mathcal{S}$ and thus incurs $\Omega(S)$ sample complexity, it cannot asymptotically dominate the approach of [19] when both algorithms are subject to an l_∞ error guarantee (unlike `Backward-EPE`). Instead, we will show in this section that `Bidirectional-EPE` is conductive to a different error guarantee, and we will argue in Section 4.3.3 that this guarantee is stronger than the l_∞ guarantee of `Backward-EPE` for certain problem instances. Thus, `Bidirectional-EPE` can be viewed as a more accurate but less sample efficient variant of `Backward-EPE`.

The aforementioned error guarantee is formalized in the following theorem, which states that with high probability, the estimate $\hat{v}_{BD}$ satisfies the following (uniformly in s):

$$(1-\varepsilon_{rel})v(s) - \varepsilon_{abs} \le \hat{v}_{BD}(s) \le (1+\varepsilon_{rel})v(s) + \varepsilon_{abs}.$$

Here $\varepsilon_{rel} \in (0,1)$ denotes a relative error tolerance while $\varepsilon_{abs} > 0$ denotes an absolute error

tolerance. Put differently (with a change of constants to n_F and n_B) the estimate will satisfy

$$|\hat{v}_{BD}(s) - v(s)| \leq \frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2} \leq \max\{\varepsilon_{rel} v(s), \varepsilon_{abs}\} = \begin{cases} \varepsilon_{rel} v(s), & v(s) \geq \varepsilon_{abs}/\varepsilon_{rel} \\ \varepsilon_{abs}, & v(s) \leq \varepsilon_{abs}/\varepsilon_{rel} \end{cases}$$

with high probability. Hence, Theorem 4.3 provides a relative error guarantee for high-value states ($v(s) \geq \varepsilon_{abs}/\varepsilon_{rel}$) and an absolute error guarantee otherwise, similar to the guarantee found in [15] for the analogous bidirectional PPR estimator (see Appendices A.1-A.2).

Theorem 4.3. Fix $\varepsilon_{rel} \in (0,1)$ and $\varepsilon_{abs}, \delta > 0$, and define

$$n_F^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta) = \frac{324\varepsilon \log(4S/\delta)}{\varepsilon_{rel}^2 \varepsilon_{abs}},$$
$$n_B^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta) = \frac{3\log(4S^2/\delta)}{(\log(1+\varepsilon_{rel}/2))^2 \min_{i,j \in \mathcal{S}: Q(i,j)>0} Q(i,j)} \left\lceil \frac{\log(2\|c\|_\infty/\varepsilon_{abs})}{(1-\alpha)} \right\rceil^2.$$

Then assuming $n_F \geq n_F^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta)$ and $n_B \geq n_B^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta)$ in Algorithm 4.3, we have

$$\mathbb{P}(\cup_{s=1}^S \{|\hat{v}_{BD}(s) - v(s)| > \varepsilon_{rel} v(s) + \varepsilon_{abs}\}) \leq \delta. \tag{4.23}$$

Proof sketch. The proof separately treats errors from the backward and forward exploration stages. We briefly describe each stage here; the full proof is in Appendix C.2. For the backward stage, Lemma C.1 in Appendix C.2 shows that if $n_B \geq n_B^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta)$, then

$$\mathbb{P}\left(\cup_{s=1}^S \left\{|\underline{v}(s) - v(s)| > \frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2}\right\}\right) \leq \frac{\delta}{2}, \tag{4.24}$$

with $\underline{v}$ defined as in (4.12). We prove (4.24) in two steps. First, we show that if $n_B \geq n_B^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta)$, then $|\underline{Q}(s,s') - Q(s,s')| \leq \lambda Q(s,s')$ for some $\lambda \in (0,1)$ and all $s, s' \in \mathcal{S}$. This follows from standard Chernoff bounds after replacing $\underline{Q}$ by $\tilde{Q}$ (which is necessary for similar reasons as those discussed in Remark 4.2). Second, we show that if $|\underline{Q}(s,s') - Q(s,s')| \leq \lambda Q(s,s')$, then $\underline{v}$ is close to v (in the sense of (4.24)). To illustrate the second step, note we can use the upper bound on $\underline{Q}$ to write

$$(1-\alpha)\sum_{t=0}^{\bar{T}} \alpha^t \underline{Q}^t(s,\cdot)c \leq (1-\alpha)\sum_{t=0}^{\bar{T}} \alpha^t (1+\lambda)^t Q^t(s,\cdot)c \leq (1+\lambda)^{\bar{T}}(1-\alpha)\sum_{t=0}^{\bar{T}} \alpha^t Q^t(s,\cdot)c.$$

For large enough $\bar{T}$, the left and right sides are within $\varepsilon_{abs}/2$-additive errors of $\underline{v}(s)$ and $(1+\lambda)^{\bar{T}} v(s)$, respectively; having chosen $\bar{T}$, we can choose λ to ensure $(1+\lambda)^{\bar{T}} \leq 1 + \varepsilon_{rel}/2$. It is from here that the relative-plus-additive guarantee of Theorem 4.3 arises.

For the forward exploration stage, we let $\mathcal{G}$ denote the σ-algebra generated by the random variables from the `Backward-EPE` subroutine and show that when $n_F \geq n_F^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta)$,

$$\mathbb{P}\left(|\hat{v}_{BD}(s) - \underline{v}(s)| \geq \frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2} \middle| \mathcal{G}\right) 1\left(|\underline{v}(s) - v(s)| \leq \frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2}\right) \leq \frac{\delta}{2S}. \quad (4.25)$$

Here we use probability conditioned on $\mathcal{G}$ so that the only randomness in $|\hat{v}_{BD}(s) - \underline{v}(s)|$ is that from the forward exploration. Moreover, we can bound this term by exploiting the $\underline{Q}$-invariant to write $|\hat{v}_{BD}(s) - \underline{v}(s)|$ as the deviation of an empirical average from its mean (as discussed in Section 4.3.1) and then use standard Chernoff bounds. Roughly speaking, this requires us to use the indicator function in (4.25) to replace $v(s)$ by $\underline{v}(s)$ in the probability term; we then separately address the cases of large $\underline{v}(s)$ and small $\underline{v}(s)$ using a modification of the approach for the analogous bidirectional PPR estimators. □

Remark 4.5. While the choice of invariant for Theorems 4.1-4.2 was subtle (see Remarks 4.2-4.3), using the $\underline{Q}$-invariant for Theorem 4.3 is obvious, since $\underline{Q}$ appears in Algorithm 4.3.

4.3.3 Discussion

We next discuss Theorem 4.3. To simplify notation, we restrict to the setting of Corollary 4.1; however, the key insights extend to the more general setting of Theorem 4.2. Also, we assume the relative error tolerance ε_{rel}, the discount factor α, and inaccuracy probability δ are constants independent of S. Finally, we note Theorem 4.3 holds for random C assuming the lower bound on n_B holds almost surely; see Remark C.2.

We begin by deriving expressions for the sample complexity of `Bidirectional-EPE` in the setting of Corollary 4.1. For the backward stage (i.e. the `Backward-EPE` subroutine), we require per-state sample complexity $n_B^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta)$; note this is deterministic since $\|C\|_\infty = 1$ pointwise in Corollary 4.1. Thus, the average-case sample complexity is (by Corollary 4.1),

$$\mathbb{E} n_B^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta) \times |U_{k_*}| = O\left(\frac{\log(S)\log(1/\varepsilon_{abs})}{\min_{i,j \in \mathcal{S}: Q(i,j)>0} Q(i,j)} \times \frac{H\bar{d}}{\varepsilon}\right). \quad (4.26)$$

For the forward stage, we require $n_F^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta) = O(\varepsilon \log(S)/\varepsilon_{abs})$ trajectories of expected length $\alpha/(1-\alpha)$ for each of S states. We are assuming α is a constant, and thus the expected forward complexity is simply $O(\varepsilon S \log S/\varepsilon_{abs})$. Combined with (4.26), and writing K_{BD} for the overall expected sample of `Bidirectional-EPE` in the setting of Corollary 4.1, we obtain

$$K_{BD} = O\left(\frac{H\bar{d}\log(S)\log(1/\varepsilon_{abs})}{\varepsilon \min_{i,j \in \mathcal{S}: Q(i,j)>0} Q(i,j)} + \frac{\varepsilon S \log S}{\varepsilon_{abs}}\right).$$

Here the termination parameter ε for the `Backward-EPE` subroutine is a free parameter that

can be chosen to minimize the overall sample complexity. For example,

$$\varepsilon = \Theta\left(\sqrt{\frac{H\bar{d}\varepsilon_{abs}}{S\min_{i,j\in\mathcal{S}:Q(i,j)>0}Q(i,j)}}\right) \Rightarrow K_{BD} = O\left(\sqrt{\frac{SH\bar{d}}{\varepsilon_{abs}\min_{i,j\in\mathcal{S}:Q(i,j)>0}Q(i,j)}}\log S\right), \tag{4.27}$$

where for simplicity we wrote $\log(1/\varepsilon_{abs})/\sqrt{\varepsilon_{abs}}$ as simply $1/\sqrt{\varepsilon_{abs}}$. Now to better understand (4.27), we will consider a specific choice of ε_{abs} (similar in spirit to the choice of the analogous parameter the PPR setting; see Appendix A.1). To motivate this, we first observe that in the setting of Corollary 4.1,

$$\mathbb{E}v = (1-\alpha)\sum_{t=0}\alpha^t Q^t \times \mathbb{E}C = (1-\alpha)\sum_{t=0}\alpha^t Q^t \times \frac{H}{S}1_{S\times 1} = \frac{H}{S}1_{S\times 1},$$

i.e. the "typical" value in the setting of Corollary 4.1 is H/S. It is thus sensible to choose $\varepsilon_{abs} = \Theta(H/S)$, so that we obtain a relative guarantee for above-typical values and settle for the absolute guarantee for below-typical values. Substituting into (4.27), we conclude

$$K_{BD} = O\left(\sqrt{\frac{\bar{d}}{\min_{i,j\in\mathcal{S}:Q(i,j)>0}Q(i,j)}}S\log S\right) \tag{4.28}$$

samples are required to guarantee (4.23) in the setting of Corollary 4.1.

It is interesting to compare `Bidirectional-EPE` to a certain plug-in estimator that lends itself to the same accuracy guarantee. For this plug-in estimator, we simply estimate v as $(1-\alpha)\sum_{t=0}\alpha^t\tilde{Q}^tC$, where $\tilde{Q}(s,\cdot) = \frac{1}{n}\sum_{i=1}^n 1(Y_{s,i}=\cdot)$ with $Y_{s,i}\sim Q(s,\cdot)$ as in (4.10). Then by the same argument following (C.17) in the proof of Theorem 4.3, the plug-in estimate will satisfy the guarantee (4.23) whenever $n \geq n_B^*(\varepsilon_{rel},\varepsilon_{abs},\delta)$. Consequently, the sample complexity of the plug-in estimator is, under the assumptions leading to (4.28),

$$Sn_B^*(\varepsilon_{rel},\varepsilon_{abs},\delta) = O\left(\frac{S\log S}{\min_{i,j\in\mathcal{S}:Q(i,j)>0}Q(i,j)}\right). \tag{4.29}$$

Comparing (4.28) and (4.29), we see `Bidirectional-EPE` is more efficient than the plug-in estimator whenever $\bar{d} \leq 1/\min_{i,j\in\mathcal{S}:Q(i,j)>0}Q(i,j)$. To interpret this inequality, first suppose the supergraph is precisely the graph induced by Q, i.e. $A(s,s') = 0 \Leftrightarrow Q(s,s') = 0$. Then

$$\sum_{s'\in\mathcal{S}}A(s,s') = \sum_{s'\in\mathcal{S}:Q(s,s')>0}\frac{Q(s,s')}{Q(s,s')} \leq \frac{\sum_{s'\in\mathcal{S}:Q(s,s')>0}Q(s,s')}{\min_{i,j\in\mathcal{S}:Q(i,j)>0}Q(i,j)} = \frac{1}{\min_{i,j\in\mathcal{S}:Q(i,j)>0}Q(i,j)},$$

so $\bar{d} \leq 1/\min_{i,j\in\mathcal{S}:Q(i,j)>0}Q(i,j)$ holds. More generally, this shows that the complexity of

`Bidirectional-EPE` is order-wise equivalent to that of the plug-in method whenever degrees in the supergraph and induced graph are order-wise equivalent. If also most positive transition probabilities dominate the minimum probability, then $\bar{d} = o(1/\min_{i,j \in \mathcal{S}: Q(i,j)>0} Q(i,j))$, in which case `Bidirectional-EPE` is strictly better.

Generally, it is difficult to compare the sample complexity (4.28) to the bounds derived in Section 4.2, owing to the different error guarantees. However, we note the l_∞ guarantee for the estimators Section 4.2 implies the guarantee of Theorem 4.3 if we choose $\varepsilon_{abs} = H/S$ for the l_∞ error tolerance; this choice gives $O(S^3 \log S/H^2)$ sample complexity for the Section 4.2 estimators. In certain cases, `Bidirectional-EPE` is thus dramatically more efficient: for instance, if $\bar{d} = O(1)$, $\min_{i,j \in \mathcal{S}: Q(i,j)>0} Q(i,j) = \Omega(1)$, and $H = O(1)$, then $K_{BD} = O(S \log S)$ but $O(S^3 \log S/H^2) = O(S^3 \log S)$, i.e. ***`Bidirectional-EPE`*** *reduces the sample complexity of the Section 4.2 estimators by a factor of* S^2. This illustrates that `Bidirectional-EPE` is more sample efficient than the approach from [19] when a highly accurate estimate is desired.

4.4 Conclusions and future directions

4.4.1 Adaptation of other PageRank algorithms

In this chapter, we adapted `Approx-Contributions` and `Bidirectional-PPR` to the EPE setting. As discussed in Sections 2.2-2.3, many related algorithms exist (e.g. the forward exploration analogue of `Approx-Contributions` from [7], our algorithm `FW-BW-MCMC` from Section 2.3, etc.). Each of these algorithms relies on an invariant analogous to (4.6), so each could (in principle) be adapted to the current setting using our analytical approach. Thus, while we have focused on two specific algorithms in this chapter, our analysis should be viewed as an example of how to extend a family of algorithms to the setting of EPE.

4.4.2 Finite horizon empirical policy evaluation

We studied the discounted cost problem in this chapter, where the value function is given by (4.3). Another problem in the MDP literature is the finite horizon problem, wherein one aims to minimize the total cost over a finite time horizon T. Here the value function is

$$v(s) = \mathbb{E}\left[\sum_{t=0}^{T} c(Z_t) \middle| Z_0 = s\right] = \sum_{t=0}^{T} Q^t(s,\cdot)c,$$

so one aims to estimate multi-step transition distributions of the form $Q^t(s,\cdot)$. Though our algorithms do not immediately apply, relevant analogues of `Approx-Contributions` exist in the case where Q is known. In particular, [95] provides an algorithm to estimate $Q^t(s,\cdot)$ when Q is known. The algorithm is analogous to `Approx-Contributions` in that it explores backward while only pushing residual mass from a single high-residual state at each iteration.

Moreover, the authors of [95] provide a bidirectional variant. Both of these algorithms could be adapted to the current setting; this would yield analogues of `Backward-EPE` and `Bidirectional-EPE` for the finite horizon problem. As in Section 4.4.1, the analysis in [95] relies on a Q-invariant analogous to (4.6) and thus our analytical approach may be useful.

4.4.3 Reusing samples versus resampling

As an alternative to `Backward-EPE`, we can take independent samples from $Q(s, \cdot)$ for each $s \in N_{in}(s_k)$ and at each iteration k, rather than only sampling from $Q(s, \cdot)$ when we first encounter s as in `Backward-EPE`. This alternative scheme is formally defined in Appendix C.3. An interesting property of this alternative is that, while the Q-invariant (4.6) need not hold, the corresponding error process $e_k(s) = \hat{v}_k(s) + \mu_s r_k - v(s)$ is a zero-mean martingale (see Appendix C.3 for a proof), so the Q-invariant holds in expectation. Analytically, this is an advantage over `Backward-EPE`, where the $\overline{Q}$- and $\underline{Q}$-invariants hold but the corresponding value functions $\overline{v}$ and $\underline{v}$ are biased. The disadvantage of this scheme is that it may sample many times from each row of Q, and thus the overall sample complexity may exceed that of the approach from [19]. Put differently, `Backward-EPE` is conservative in the sense that it performs no worse than the existing approach in the worst case (see Section 4.2.4), but it sacrifices desirable properties that could perhaps improve performance in other cases. A useful avenue for future work would thus be to investigate this tradeoff.

4.4.4 PageRank estimation with limited knowledge

A problem that has received little attention in the PageRank/PPR literature is PageRank/PPR estimation when the estimator has limited knowledge of Q. For instance, consider a third party who wishes to pay influential Twitter users to promote their products. Since PageRank serves as a measure of influence or centrality in networks, the third party may wish to identify high PageRank users, but the Twitter graph (as encoded by Q) is not publicly available, and thus existing PageRank estimators do not apply. However, Twitter does allow limited data requests [96], which may allow the third party to partially recover relevant entries of Q. This setting could be abstracted as the follows: devise an algorithm that estimates PageRank while only sampling from Q and while minimizing sample complexity. This is similar to the problem we considered in this chapter, with one major difference: we justified the existence of a supergraph based on, for example, physical limitations that prevent transitions between states; if Q represents Twitter, states (i.e. Twitter users) can be connected arbitrarily. Thus, we could perhaps replace knowledge of the supergraph with sampling of incoming neighbors, i.e. given a Twitter user, we can sample a random follower via data request. This would serve a similar purpose as the supergraph (allowing us to understand columns of Q), and we suscept many of our ideas could be recycled.

CHAPTER V

Restart Perturbations for Lazy, Reversible Markov Chains[1]

5.1 Introduction

In this chapter, we apply analytical ideas from the PPR literature to study the robustness of Markov models. Our motivation is the basic question of how modeling inaccuracies affect a chain's steady-state behavior, i.e. how changes to the transition matrix affect the stationary distribution. Mathematically, we formalize this as follows. Let P_n be the transition matrix of a Markov chain with n states and stationary distribution π_n. Denote by $\tilde{P}_n$ the transition matrix and $\tilde{\pi}_n$ the stationary distribution of another chain, obtained by perturbing each row of P_n by at most $\alpha_n \in (0, 1)$ (in total variation). Then the main question we study is as follows: how does the perturbation magnitude α_n relate to the error magnitude $\|\pi_n - \tilde{\pi}_n\|$ (where $\|\cdot\|$ denotes total variation) as the number of states n grows?

Before previewing our results, we outline two basic notions. The first is a class of PPR-like perturbations we call *restart perturbations* in this chapter. Here we obtain $\tilde{P}_n$ from P_n by "restarting" at a state distributed as some auxiliary distribution σ_n with probability α_n at each step. A second important notion is that of mixing times and cutoff. Roughly, the *ε-mixing time* $t_{\text{mix}}^{(n)}(\varepsilon)$ is the number of steps the chain with transition matrix P_n must take before its distribution is ε-close to π_n (see (5.3)). Certain chains exhibit *cutoff*, meaning

$$\lim_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon) = 1 \;\forall\; \varepsilon \in (0, 1/2). \tag{5.1}$$

Intuitively, (5.1) says the chain is far from stationarity for many steps, then abruptly becomes

[1]This chapter is adapted from [97].

close to stationarity. A weaker condition is *pre-cutoff*, which only requires

$$\sup_{\varepsilon\in(0,1/2)} \limsup_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon) < \infty. \tag{5.2}$$

We now preview the two main results of this chapter. The first, Theorem 5.1, says that the relative asymptotics of α_n and $t_{\text{mix}}^{(n)}(\varepsilon)$ fully characterize the asymptotics of $\|\pi_n - \tilde{\pi}_n\|$ in the case of restart perturbations. More specifically, we establish the following trichotomy:

- If $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) = 0$, then $\lim_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| = 0$ for all restart perturbations.
- If $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) = \infty$, then $\lim_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| = 1$ for some restart perturbation.
- If $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) = c \in (0,\infty)$, then $\limsup_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| \leq 1 - e^{-c}$ for all restart perturbations, and some restart perturbation attains the bound.

We note Theorem 5.1 holds assuming the original chain is lazy ($P_n(i,i) \geq 1/2 \; \forall \; i$), reversible ($\pi_n(i)P_n(i,j) = \pi_n(j)P_n(j,i) \; \forall \; i,j$), and exhibits cutoff. The laziness and reversibility assumptions are inherited from [22], which contains an inequality used in our lower bounds (see Section 5.3). Hence, we suspect these assumptions may be artifacts of our analysis. In contrast, we believe some notion of cutoff is fundamentally necessary (as will be discussed shortly). Also, parts of our analysis hold more generally; see Lemmas 5.1 and 5.2.

Interestingly, Theorem 5.1 says that a threshold phenomena for the original chain – cutoff – translates into a different threshold phenomena for the perturbed chain – the trichotomy above. Another point of interest is that similar trichotomies have been established in several recent works. For example, [21] shows that the restart perturbation adopts the cutoff behavior of the original chain if $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to 0$, has a distinct convergence to stationarity if $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$, and exhibits an intermediate behavior if $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to (0,\infty)$, assuming the original chain is the random walk on the directed configuration model from Chapter III. Similar results were obtained in [20, 98] for random walks on dynamic versions of the DCM. Finally, in Chapter III we showed the PPR matrix has dimension $O(1)$ if $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to 0$, conjectured the dimension is $\Omega(n/\log n)$ if $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$, and proved the dimension is $O(n^{f(c)})$ for some $f(c) \in (0,1)$ if $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to c \in (0,\infty)$, when the original chain is generated as in [21]. See Section 5.6 for details of these related results.

Ultimately, this chapter, [21, 20, 98], and Chapter III all study different questions, but the similarities speak to a much deeper phenomena: some aspect of the original chain is unaffected when $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to 0$, this aspect is significantly altered when $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$, and an intermediate behavior occurs when $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to (0,\infty)$. However, in contrast to [21], [20], and Chapter III, we work directly with the stationary distribution in this chapter, which is arguably the most fundamental such aspect one would hope to understand. Additionally, we do not assume a generative model for the original chain in this chapter; in this sense, the

results of this chapter are more general, while demonstrating a similar idea.

Our second result concerns pre-cutoff. As alluded to above, we believe some notion of cutoff is fundamental for the lower bounds of Theorem 5.1. Indeed, in Theorem 5.2 we show that for lazy and reversible chains, pre-cutoff (5.2) implies a perturbation condition, and

$$\sup_{\varepsilon\in(0,1/2)} \liminf_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon) = \infty$$

implies the negation of the perturbation condition. Roughly, this condition is as follows: for certain $\{\alpha_{n,\varepsilon}\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)} \subset (0,1)$ and all $\varepsilon \in (0,1/2)$, there exists a sequence of restart perturbations with restart probabilities $\{\alpha_{n,\varepsilon}\}_{n\in\mathbb{N}}$ and stationary distributions $\{\tilde{\pi}_{n,\varepsilon}\}_{n\in\mathbb{N}}$ s.t. $\|\pi_n - \tilde{\pi}_{n,\varepsilon}\| \to 1$. Hence, Theorem 5.2 says that chains with pre-cutoff are sensitive to perturbation, in the sense that certain perturbations maximally change the stationary distribution, and the converse (almost) holds. The only gap in our logic involves the case

$$\sup_{\varepsilon\in(0,1/2)} \liminf_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon) < \infty = \sup_{\varepsilon\in(0,1/2)} \limsup_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon),$$

which only occurs for a class of chains of little interest (see Section 5.4). Thus, for all intents and purposes, Theorem 5.2 is an equivalence between pre-cutoff and sensitivity.

The utility of Theorem 5.2 is that, while different notions of cutoff have been proven for different chains, there is little general theory. In fact, only recently was a condition equivalent to cutoff determined in [22] (a certain notion of "hitting time cutoff"). Additionally, while Theorem 5.2 relies on an inequality from [22], we believe it is more than a corollary of this inequality. Instead, we believe our result complements [22], since we consider pre-cutoff instead of cutoff, and since our equivalent notion is different. See Section 5.6 for details.

In short, this chapter contributes to two lines of work. First, we add to the growing collection of "trichotomy" results; unlike existing results, however, we study the stationary distribution directly and do not assume a generative model. Second, we add to the general theory of cutoff in the manner of [22], but for a different notion of cutoff.

The remainder of the chapter is organized as follows. We begin in Section 5.2 with definitions. Sections 5.3 and 5.4 contain the two theorems described above. We present examples in Section 5.5. Finally, Section 5.6 discusses the related results mentioned above, and Section 5.7 discusses conclusions and future directions.

5.2 Preliminaries

We first define the notation used in this chapter. Let $\mathbb{Z}_+ = \{0, 1, \ldots\}$, and let $\{X_n(t)\}_{t\in\mathbb{Z}_+}$ be a time-homogeneous, irreducible, aperiodic Markov chain with states $[n] = \{1, \ldots, n\}$.

We denote by P_n the transition matrix of this chain, i.e. the matrix with (i,j)-th entry

$$P_n(i,j) = \mathbb{P}(X_n(t+1) = j | X_n(t) = i) \ \forall \ i,j \in [n], t \in \mathbb{Z}_+.$$

It is a standard result that this chain has a unique stationary distribution π_n, i.e. a unique vector π_n satisfying $\pi_n = \pi_n P_n$ and $\sum_{i=1}^n \pi_n(i) = 1$. Here and for the remainder of the chapter, we treat all vectors as row vectors. For $i \in [n]$, we let e_i denote the length-n vector with 1 in the i-th coordinate and zeros elsewhere. Also, we let Δ_{n-1} denote the set of distributions over $[n]$, so that (for example) $\pi_n \in \Delta_{n-1}$. Finally, we let $\mathcal{E}_n$ denote the set of transition matrices for time-homogeneous, irreducible, and aperiodic Markov chains with state space $[n]$, so that (for example) $P_n \in \mathcal{E}_n$.

Some of our results will only apply to a strict subset of $\mathcal{E}_n$. In particular, we at times require the chain to be lazy, meaning $P_n(i,i) \geq 1/2 \ \forall \ i \in [n]$, and reversible, meaning $\pi_n(i)P_n(i,j) = \pi_n(j)P_n(j,i) \ \forall \ i,j \in [n]$. Note any chain can be made lazy without changing its stationary distribution, by considering $(P_n + I_n)/2$ instead of P_n, where I_n is the $n \times n$ identity matrix. In this sense, reversibility is our most restrictive assumption. However, this is a common restriction in the mixing times literature, as it guarantees the eigenvalues of P_n are real and allows one to use certain linear algebraic techniques (see e.g. [99, Chapter 12]).

As discussed in Section 5.1, mixing times will play a pivotal role. To define mixing times, we first define the distance between the t-step distribution and stationarity as

$$d_n(t) = \max_{i \in [n]} \|e_i P_n^t - \pi_n\| \ \forall \ t \in \mathbb{Z}_+,$$

where $\|\cdot\|$ denotes total variation distance, $\|\mu - \nu\| = \max_{A \subset [n]} |\mu(A) - \nu(A)|$ for $\mu, \nu \in \Delta_{n-1}$. For $\varepsilon \in (0,1)$, we can now define the ε-mixing time as

$$t_{\text{mix}}^{(n)}(\varepsilon) = \min\{t \in \mathbb{Z}_+ : d_n(t) \leq \varepsilon\}. \tag{5.3}$$

As is convention in the literature, we set $t_{\text{mix}}^{(n)} = t_{\text{mix}}^{(n)}(1/4)$. We also note the following monotocity property follows immediately, but we record it here as it will be used often:

$$\forall \ \varepsilon, \delta \in (0,1) \text{ s.t. } \varepsilon \leq \delta, \ t_{\text{mix}}^{(n)}(\varepsilon) \geq t_{\text{mix}}^{(n)}(\delta). \tag{5.4}$$

Having defined mixing times, cutoff (5.1) and pre-cutoff (5.2) are now clearly defined[2]. We

[2]Note $\|e_i - \pi_n\| \geq 1 - \pi_n(i) \ \forall \ i \in [n]$, so $d_n(0) \geq 1 - \min_{i \in [n]} \pi_n(i) \geq 1 - 1/n > 1 - \varepsilon$ for fixed ε and n large. Thus, $t_{\text{mix}}^{(n)}(1-\varepsilon) > 0$ for such n, so the fractions in (5.1) and (5.2) is well-defined for large n. Along these lines, we at times assume $t_{\text{mix}}^{(n)}(1-\varepsilon) \geq 1$, with the implicit understanding that this holds for large n.

note a basic result (see e.g. Section 18.1 of [99]) says that cutoff occurs if and only if

$$s < 1 \Rightarrow \lim_{n\to\infty} d_n(st_{\text{mix}}^{(n)}) = 1, \quad s > 1 \Rightarrow \lim_{n\to\infty} d_n(st_{\text{mix}}^{(n)}) = 0. \tag{5.5}$$

Thus, cutoff means the graph of $d_n(t)$ approaches a step function as $n \to \infty$, when the t-axis is normalized by $t_{\text{mix}}^{(n)}$. Put differently, the chain is far from stationarity at time e.g. $0.99t_{\text{mix}}^{(n)}$, then reaches stationarity at time e.g. $1.01t_{\text{mix}}^{(n)}$. Pre-cutoff has weaker but similar intuition.

For the perturbation analysis described in the introduction, it will be convenient to introduce some additional notation. First, given $P_n \in \mathcal{E}_n$ and $\alpha \in (0, 1)$, we define

$$B(P_n, \alpha) = \left\{ \tilde{P}_n \in \mathcal{E}_n : \max_{i\in[n]} \|e_i P_n - e_i \tilde{P}_n\| \leq \alpha \right\}. \tag{5.6}$$

In words, $B(P_n, \alpha)$ is the set of transition matrices for time-homogeneous, irreducible, and aperiodic chains whose rows differ from the rows of P_n by at most α. We will denote the unique stationary distribution of $\tilde{P}_n \in B(P_n, \alpha)$ by $\tilde{\pi}_n$. A particular subset of $B(P_n, \alpha)$ is the class of restart perturbations discussed above. Such perturbations have the same form as PPR, i.e. $(1 - \alpha)P_n + \alpha 1_n^{\mathsf{T}} \sigma_n \in B(P_n, \alpha)$ for some $\alpha \in (0, 1)$ and $\sigma_n \in \Delta_{n-1}$, where 1_n is the length-n row vector of ones. For clarity, we use the notation

$$P_{\alpha,\sigma_n} = (1 - \alpha)P_n + \alpha 1_n^{\mathsf{T}} \sigma_n$$

to define restart perturbations. We denote the corresponding stationary distribution by π_{α,σ_n}. Moving forward, α will typically depend on n, so we write P_{α_n,σ_n} and π_{α_n,σ_n}.

Finally, the following (standard) notation for $\{a_n\}_{n\in\mathbb{N}}, \{b_n\}_{n\in\mathbb{N}} \subset [0, \infty)$ will be used: we write $a_n = O(b_n)$, $a_n = \Omega(b_n)$, $a_n = \Theta(b_n)$, and $a_n = o(b_n)$, resp., if $\limsup_{n\to\infty} a_n/b_n < \infty$, $\liminf_{n\to\infty} a_n/b_n > 0$, $a_n = O(b_n)$ and $a_n = \Omega(b_n)$, and $\lim_{n\to\infty} a_n/b_n = 0$, resp.

5.3 Trichotomy

In this section, we formulate our first main result, the trichotomy described in Section 5.1. For transparency, we begin with two lemmas, parts of which require weaker assumptions. We then collect these results under our strongest assumptions in Theorem 5.1.

The first lemma concerns the case $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \{0, \infty\}$. The lemma states that if the perturbation magnitude α_n is dominated by the inverse mixing time, no perturbation can change the stationary distribution. On the other hand, if α_n dominates the inverse mixing time, one can find a perturbation that maximally changes this distribution. Note the former case holds for all bounded perturbations (not just the restart variety). Also, while the latter case requires laziness and reversibility, it does not require cutoff (only pre-cutoff). Hence,

Lemma 5.1 contains stronger results for the case $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \{0, \infty\}$ than Theorem 5.1.

Lemma 5.1. Let $P_n \in \mathcal{E}_n, \alpha_n \in (0,1)\ \forall\ n \in \mathbb{N}$, and let $\varepsilon \in (0,1)$ be independent of n. Assume $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) = c \in \{0, \infty\}$. Then the following hold:

- If $c = 0$ and $\varepsilon < 1/2$, then $\forall\ \{\tilde{P}_n\}_{n\in\mathbb{N}}$ s.t. $\tilde{P}_n \in B(P_n, \alpha_n)\ \forall\ n \in \mathbb{N}$,

$$\lim_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| = 0. \tag{5.7}$$

- If $c = \infty$, $\{P_n\}_{n\in\mathbb{N}}$ exhibits pre-cutoff, and each P_n is lazy and reversible, then $\exists\ \{\tilde{P}_n\}_{n\in\mathbb{N}}$ s.t. $\tilde{P}_n \in B(P_n, \alpha_n)\ \forall\ n \in \mathbb{N}$ and

$$\lim_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| = 1. \tag{5.8}$$

  In particular, $\forall\ n \in \mathbb{N}$, $\tilde{P}_n$ is a restart perturbation, i.e. $\tilde{P}_n = P_{\alpha_n,\sigma_n}$ for some $\sigma_n \in \Delta_{n-1}$.

Proof. See Appendix D.2 □

We briefly discuss the proof. The case $c = 0$ is simpler and relies on standard mixing time results. In particular, we use the fact that distance to stationarity decays exponentially after it reaches $1/2$ ($d_n(kt_{\text{mix}}^{(n)}(\varepsilon)) \le (2\varepsilon)^k\ \forall\ k \in \mathbb{N}$), hence the additional assumption $\varepsilon < 1/2$ in this case. The case $c = \infty$ is more involved. The key step is to establish a weaker version of (5.8): namely, $\forall\ \delta > 0$ s.t. $\alpha_n t_{\text{mix}}^{(n)}(\delta) \to \infty$, $\exists\ \tilde{P}_n \in B(P_n, \alpha_n)$ s.t.

$$\liminf_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| \ge 1 - 3\delta. \tag{5.9}$$

After proving (5.9), we define a vanishing sequence $\{\delta_k\}_{k\in\mathbb{N}}$ and apply (5.9) to each $k \in \mathbb{N}$ to reach the stronger conclusion of (5.8). (The extension to (5.8) is not as immediate because the left side of (5.9) has a dependence on δ; however, it is still reasonably simple.)

Before proceeding, we discuss further the key step from the $c = \infty$ case, i.e. the proof of (5.9). This proof involves a construction of $\tilde{P}_n$ that relies on a result from the aforementioned [22]. Roughly speaking, this result shows that one can find a state $x_n \in [n]$, a subset of states $A_n \subset [n]$, and some $t_n \in \mathbb{Z}_+$, such that $\{X_n(t)\}_{t\in\mathbb{Z}_+}$ is unlikely to reach A_n within t_n steps when started from $X_n(0) = x_n$. Furthermore, in the case of pre-cutoff, $\pi_n(A_n)$ is large and t_n is comparable to $t_{\text{mix}}^{(n)}(\delta)$. In summary, the chain started from x_n makes its first visit to a "large" set A_n just before $t_{\text{mix}}^{(n)}(\delta)$.

This argument suggests a good construction for the perturbed chain: set $\tilde{P}_n = P_{\alpha_n, e_{x_n}}$, i.e. perturb the chain by restarting at x_n with probability α_n at each step. On this perturbed chain, the number of steps between restarts at x_n is (in expectation) $1/\alpha_n$; hence, when $\alpha_n t_{\text{mix}}^{(n)}(\delta) \to \infty$, restarts occur at intervals typically much shorter than $t_{\text{mix}}^{(n)}(\delta)$. In other

words, the perturbed chain rarely wanders $t_{\text{mix}}^{(n)}(\delta)$ steps from x_n. But, per the previous paragraph, the chain started from x_n requires $t_{\text{mix}}^{(n)}(\delta)$ steps to reach A_n. Hence, the perturbed chain rarely visits A_n and thus assigns a small stationary measure to A_n. Finally, since $\pi_n(A_n)$ is large, the definition of total variation ensures $\|\pi_n - \tilde{\pi}_n\| \geq \pi_n(A_n) - \tilde{\pi}_n(A_n)$ is also large. This intuition is the key idea behind (5.9).

We turn to the second lemma, which considers the case $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to (0, \infty)$. This lemma contains two bounds; one analogous to the upper bound (5.7) and one analogous to the lower bound (5.8). Here we require stronger assumptions than Lemma 5.1. For the upper bound, we restrict to restart perturbations and we assume $t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$ as $n \to \infty$. This latter assumption is minor, since typically one studies the growth rate of $t_{\text{mix}}^{(n)}(\varepsilon)$, and thus chains that mix in constant time are of less interest. For the lower bound, we again assume laziness and reversibility, as well as strengthening the pre-cutoff assumption to cutoff. The proof is similar to that of Lemma 5.1, but the stronger assumptions allow for a tighter analysis.

Lemma 5.2. Let $P_n \in \mathcal{E}_n, \alpha_n \in (0,1)\ \forall\ n \in \mathbb{N}$, and let $\varepsilon \in (0,1)$ be independent of n. Assume $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) = c \in (0, \infty)$. Then the following hold:

- If $\lim_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon) = \infty$, then $\forall\ \{\sigma_n\}_{n\in\mathbb{N}}$ s.t. $\sigma_n \in \Delta_{n-1}\ \forall\ n \in \mathbb{N}$,

$$\limsup_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| \leq \begin{cases} 1 - (1-\varepsilon)e^{-c}, & \varepsilon \in [1/2, 1) \\ \min\{1 - (1-\varepsilon)e^{-c}, (1-e^{-c})/(1-2\varepsilon e^{-c})\}, & \varepsilon \in (0, 1/2) \end{cases}. \tag{5.10}$$

- If $\{P_n\}_{n\in\mathbb{N}}$ exhibits cutoff and each P_n is lazy and reversible, then $\exists\ \{\sigma_n\}_{n\in\mathbb{N}}$ s.t. $\sigma_n \in \Delta_{n-1}\ \forall\ n \in \mathbb{N}$ and $\liminf_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| \geq 1 - e^{-c}$.

Proof. See Appendix D.3. □

Before proceeding, we comment on the upper bound in the case $\varepsilon \in (0, 1/2)$, which (we note) includes the usual case of interest $\varepsilon = 1/4$. Here one can verify

$$\min\{1 - (1-\varepsilon)e^{-c}, (1-e^{-c})/(1-2\varepsilon e^{-c})\} = \begin{cases} 1 - (1-\varepsilon)e^{-c}, & c \geq \log(2(1-\varepsilon)) \\ (1-e^{-c})/(1-2\varepsilon e^{-c}), & c \leq \log(2(1-\varepsilon)) \end{cases}.$$

Hence, for smaller c, the upper bound in Lemma 5.2 is $(1-e^{-c})/(1-2\varepsilon e^{-c})$, while for larger c, the bound is $1-(1-\varepsilon)e^{-c}$. Note the former bound approaches 0 as $c \to 0$, and thus approaches the $c = 0$ case of Lemma 5.1. Furthermore, the latter bound approaches 1 and thus becomes trivial as $c \to \infty$; this is expected due to the $c = \infty$ case of Lemma 5.1.

Combining Lemmas 5.1 and 5.2, we arrive at our first main result. Theorem 5.1 collects the results of the lemmas under our strongest assumptions: the chain is lazy, reversible,

and exhibits cutoff, and the perturbation is restricted to the restart variety. Under these assumptions, we can fully characterize perturbation behavior. Note these assumptions are stronger than those required for the upper bounds in the lemmas, which allows us to discard the $\varepsilon < 1/2$ assumption of Lemma 5.1 and the $t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$ assumption of Lemma 5.2.

Theorem 5.1. Let $P_n \in \mathcal{E}_n, \alpha_n \in (0,1)\ \forall\ n \in \mathbb{N}$, and let $\varepsilon \in (0,1)$ be independent of n. Assume $\{P_n\}_{n\in\mathbb{N}}$ exhibits cutoff, each P_n is lazy and reversible, and $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) = c \in [0,\infty]$. Then the following hold:

- If $c = 0$, then $\forall\ \{\sigma_n\}_{n\in\mathbb{N}}$ s.t. $\sigma_n \in \Delta_{n-1}\ \forall\ n \in \mathbb{N}$,

$$\lim_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| = 0. \tag{5.11}$$

- If $c \in (0,\infty)$, then $\forall\ \{\sigma_n\}_{n\in\mathbb{N}}$ s.t. $\sigma_n \in \Delta_{n-1}\ \forall\ n \in \mathbb{N}$,

$$\limsup_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| \le 1 - e^{-c}. \tag{5.12}$$

  Furthermore, (5.12) is tight, i.e. $\exists\ \{\sigma_n\}_{n\in\mathbb{N}}$ s.t. $\sigma_n \in \Delta_{n-1}\ \forall\ n \in \mathbb{N}$ and

$$\liminf_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| \ge 1 - e^{-c}. \tag{5.13}$$

- If $c = \infty$, then $\exists\ \{\sigma_n\}_{n\in\mathbb{N}}$ s.t. $\sigma_n \in \Delta_{n-1}\ \forall\ n \in \mathbb{N}$ and

$$\lim_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| = 1. \tag{5.14}$$

Proof. See Appendix D.4. □

5.4 Pre-cutoff equivalence

We next turn to Theorem 5.2. As discussed in the introduction, the theorem provides a near-equivalence between pre-cutoff and a perturbation condition. More specifically, we will show that pre-cutoff implies a perturbation condition, and that this condition fails if

$$\sup_{\varepsilon\in(0,1/2)} \liminf_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon) = \infty. \tag{5.15}$$

The caveat of Theorem 5.2 being a near-equivalence arises because (5.15) is stronger than the negation of pre-cutoff. Indeed, one can construct sequences of chains for which pre-cutoff and (5.15) both fail. For instance, in Section 5.5 we provide two example sequences with

drastically different cutoff behaviors; if we oscillate between these two, we obtain

$$\liminf_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon) = 1, \quad \limsup_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon) = \infty, \quad \forall\ \varepsilon \in (0,1/2).$$

However, this oscillating sequence is pathological; the literature typically considers chains defined in the same manner for each n. Thus, the "near-equivalence" caveat is a small one.

Before presenting Theorem 5.2, we must define the perturbation condition. However, this condition is somewhat mysterious, so we first discuss the difficulty in deriving it, in hopes of making it less opaque. We begin with an obvious candidate, the condition from Lemma 5.1:

$$\forall\ \{\alpha_n\}_{n\in\mathbb{N}} \text{ s.t. } \lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) = \infty,\ \exists\ \{\tilde{P}_n\}_{n\in\mathbb{N}} \text{ s.t. } \lim_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| = 1. \tag{5.16}$$

Indeed, we have already proven that pre-cutoff implies (5.16) (assuming laziness and reversibility). The difficulty is showing that (5.16) fails whenever (5.15) holds. The most obvious approach is as follows. When (5.15) holds, it is *possible* that for a fixed $\varepsilon \in (0,1/2)$,

$$\lim_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon) = \infty, \tag{5.17}$$

which suggests setting $\alpha_n = c/t_{\text{mix}}^{(n)}(1-\varepsilon)$ for some c independent of n, since then

$$\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) = c \lim_{n\to\infty} t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon) = \infty.$$

Our task would then be reduced to upper bounding $\|\pi_n - \tilde{\pi}_n\|$ (perhaps via techniques used for upper bounds above). Unfortunately, we are not guaranteed that (5.17) holds.

While this attempt fails, it illustrates the dissonance at hand: (5.16) considers sequences $\{\alpha_n\}_{n\in\mathbb{N}}$ depending only on n, while the sequence $\{1/t_{\text{mix}}^{(n)}(1-\varepsilon)\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)}$ in (5.15) depends on both n and ε. Hence, we could modify (5.16) to involve a sequence $\{\alpha_{n,\varepsilon}\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)}$ depending on both n and ε. However, if (5.16) is modified in this manner, it is no longer implied by pre-cutoff via Lemma 5.1, so this direction of the proof may become difficult.

It turns out this issue can be resolved by placing appropriate restrictions on the set of sequences of restart probabilities appearing in the perturbation condition. In particular, we will say $\{\alpha_{n,\varepsilon}\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)} \subset (0,1)$ *coincides with* the mixing times $\{t_{\text{mix}}^{(n)}(\varepsilon)\}_{n\in\mathbb{N},\varepsilon\in(0,1)}$ if[3]

$$\sup_{\varepsilon\in(0,1/2)} \liminf_{n\to\infty} \alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(\varepsilon) = \infty, \frac{\alpha_{n,\varepsilon}}{\alpha_{n,\delta}} \in \left[\frac{t_{\text{mix}}^{(n)}(1-\delta)}{t_{\text{mix}}^{(n)}(1-\varepsilon)}, 1\right]\ \forall\ \varepsilon,\delta \in (0,1/2) \text{ s.t. } \varepsilon \geq \delta, \forall\ n \in \mathbb{N}, \tag{5.18}$$

[3]As shown in the proof of Theorem 5.2, such sequences always exist under the assumption of laziness.

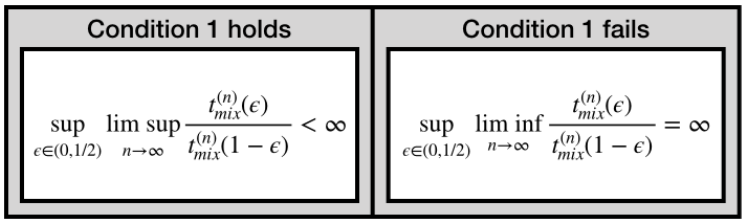


Figure 5.1: Partition of lazy/reversible sequences of chains induced by Condition 5.1. Theorem 5.2 says chains satisfying pre-cutoff and (5.15), respectively, are contained in the subsets for which Condition 5.1 holds and fails, respectively. The gray subset contains e.g. the pathological example from Section 5.4.

and we will restrict to sequences that coincide with the mixing times. More specifically, we define the following perturbation condition for use in our second main result.

Condition 5.1. For any $\{\alpha_{n,\varepsilon}\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)} \subset (0,1)$ that coincides with the mixing times $\{t_{\text{mix}}^{(n)}(\varepsilon)\}_{n\in\mathbb{N},\varepsilon\in(0,1)}$, there exists $\{\sigma_{n,\varepsilon}\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)}$ such that

$$\sigma_{n,\varepsilon} \in \Delta_{n-1} \; \forall \; n \in \mathbb{N}, \varepsilon \in (0,1/2), \quad \lim_{n\to\infty} \|\pi_n - \pi_{\alpha_{n,\varepsilon},\sigma_{n,\varepsilon}}\| = 1 \; \forall \; \varepsilon \in (0,1/2).$$

The definition of "coincides with" yields a crucial property: when pre-cutoff holds and $\{\alpha_{n,\varepsilon}\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)}$ coincides with $\{t_{\text{mix}}^{(n)}(\varepsilon)\}_{n\in\mathbb{N},\varepsilon\in(0,1)}$, $\alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(\varepsilon) \to \infty \; \forall \, \varepsilon \in (0,1/2)$. In words, not only is the sup in (5.18) infinite, the lim inf in (5.18) is infinite, for every $\varepsilon \in (0,1/2)$. This allows us to prove (via Lemma 5.1) that Condition 5.1 is implied by pre-cutoff, while also proving that Condition 5.1 fails (via the approach discussed above) if (5.15) holds.

With Condition 5.1 in place, we present Theorem 5.2; see Figure 5.1 for an illustration.

Theorem 5.2. Let $\{P_n\}_{n\in\mathbb{N}}$ be s.t. $P_n \in \mathcal{E}_n$ is lazy and reversible $\forall \; n \in \mathbb{N}$. If $\{P_n\}_{n\in\mathbb{N}}$ exhibits pre-cutoff, Condition 5.1 holds; if $\{P_n\}_{n\in\mathbb{N}}$ satisfies (5.15), Condition 5.1 fails.

Proof. See Appendix D.5. □

5.5 Illustrative examples

Our results suggest a deep connection between some notion of cutoff and some notion of perturbation sensitivity. Here we illustrate this with two example chains called the *winning streak reversal* (WSR) and the *complete graph bijection* (CGB). We define each in turn.

The winning streak reversal (WSR) is taken from [99]. As its name suggests, this chain is the time reversal of the so-called *winning streak* chain. The winning streak chain is shown at left in Figure 5.2a and has the following interpretation. At each step, one plays a fair game. If the game is won, the winning streak is increased, meaning the state is increased by 1 (unless the current state is n, in which case the state remains n); if the game is lost,

the winning streak ends, meaning the state returns to its lowest value.[4] The reversal of this chain, which we analyze, is shown at right in Figure 5.2a. For general n, the transition matrix and stationary distribution for the WSR are (see Section 4.6 of [99] for details)

$$P_n(i,j) = \begin{cases} 2^{-j}, & i = 1, j \in \{1, \dots, n-1\} \\ 2^{-n+1}, & i = 1, j = n \\ 1, & i \in \{2, \dots, n-1\}, j = i-1 \\ 2^{-1}, & i = n, j \in \{n-1, n\} \\ 0, & \text{otherwise} \end{cases}, \pi_n(i) = \begin{cases} 2^{-i}, & i \in \{1, \dots, n-1\} \\ 2^{-n+1}, & i = n \end{cases} \tag{5.19}$$

Note that $P_n(1,i) = \pi_n(i) \; \forall \; i \in [n]$; hence, the chain started from state 1 reaches stationarity (exactly) after 1 step. Furthermore, the chain starting from $i \in \{2, \dots, n-1\}$ deterministically transitions to state 1 in $i-1$ steps and thus reaches stationarity (again, exactly) after i steps. As will be seen, this implies a particularly strong form of cutoff.

We next define the complete graph bijection (CGB). As suggested by the name, for even n we first construct complete graphs on nodes $\{1, \dots, n/2\}$ and $\{1 + n/2, \dots, n\}$; we then add edges between i and $i+n/2$ for each $i \in [n/2]$, corresponding to the bijection $i \mapsto i+n/2$. For n odd, we first construct this graph for $n-1$; we then add an auxiliary node n, along with an edge between n and every $i \in [n-1]$. Figure 5.2b shows these graphs for $n = 6$ and $n = 7$. We consider the lazy random walks on these graphs. The transition matrices are

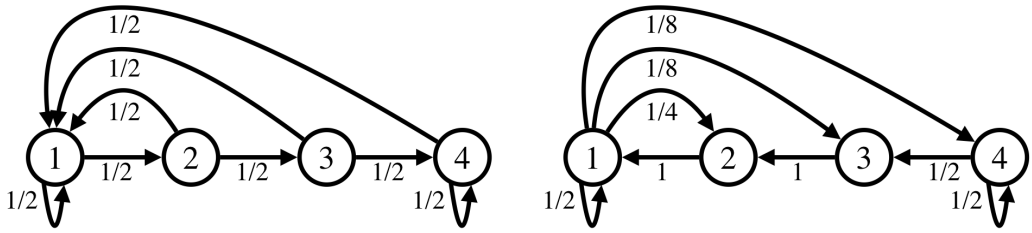
$$P_n = \frac{I_n}{2} + \frac{1}{n} \begin{bmatrix} 1_{\frac{n}{2}} 1_{\frac{n}{2}}^\intercal - I_{\frac{n}{2}} & I_{\frac{n}{2}} \\ I_{\frac{n}{2}} & 1_{\frac{n}{2}} 1_{\frac{n}{2}}^\intercal - I_{\frac{n}{2}} \end{bmatrix} \; \forall \; n \text{ even}, \tag{5.20}$$

$$P_n = \frac{1}{2} I_n + \frac{1}{n+1} \begin{bmatrix} 1_{\frac{n-1}{2}} 1_{\frac{n-1}{2}}^\intercal - I_{\frac{n-1}{2}} & I_{\frac{n-1}{2}} & 1_{\frac{n-1}{2}} \\ I_{\frac{n-1}{2}} & 1_{\frac{n-1}{2}} 1_{\frac{n-1}{2}}^\intercal - I_{\frac{n-1}{2}} & 1_{\frac{n-1}{2}} \\ \frac{n+1}{2(n-1)} 1_{\frac{n-1}{2}}^\intercal & \frac{n+1}{2(n-1)} 1_{\frac{n-1}{2}}^\intercal & 0 \end{bmatrix} \; \forall \; n \text{ odd}.$$

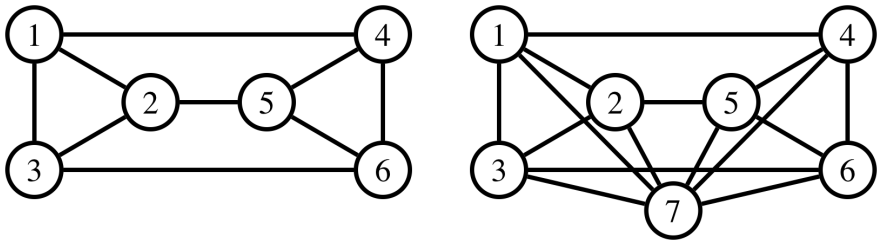
It is a standard result that the degree distribution is stationary for random walks on undirected graphs; this also holds for P_n since laziness does not change the stationary distribution. From this, one can easily verify the stationary distributions for the CGB are

$$\pi_n(i) = \frac{1}{n} \; \forall \; i \in [n], n \text{ even}, \quad \pi_n(i) = \begin{cases} \frac{n+1}{(n+3)(n-1)}, & i \in [n-1] \\ \frac{2}{n+3}, & i = n \end{cases} \; \forall \; n \text{ odd}. \tag{5.21}$$

[4] Given this interpretation, it is more sensible to use state space $\{0, \dots, n-1\}$, so that the winning streak is zero after a loss. However, for consistency with the rest of this chapter, we use state space $\{1, \dots, n\}$.



(a) Winning streak chain (left) and its reversal (right) for $n = 4$



(b) Complete graph bijection for $n = 6$ (left) and $n = 7$ (right).

Figure 5.2: Depicition of example chains.

We next state a proposition that estimates the mixing times of these chains. The proposition contains several results. First, we show both chains have $\Theta(n)$ ε-mixing time, for any fixed $\varepsilon \in (0, 1/2)$. Furthermore, the proposition says that for the WSR and for any such ε,

$$1 \leq t_{\text{mix}}^{(n)}(\varepsilon)/t_{\text{mix}}^{(n)}(1-\varepsilon) \leq 1 + \Theta\left(n^{-1}\right). \tag{5.22}$$

Hence, the ratios in (5.22) converge to 1 at rate n^{-1}, a particularly strong notion of cutoff (the standard definition of cutoff, (5.1), imposes no rate of convergence). In contrast, for the CGB, the proposition shows that these ratios are $\Theta(n)$, the maximum (up to constants) among all chains with $\Theta(n)$ ε-mixing times. In summary, while both chains have equivalent ε-mixing times, their cutoff behaviors are at opposite extremes among such chains.

Proposition 5.1. Let $\varepsilon \in (0, 1/2)$ be independent of n. Then the following hold:

- Suppose $\{P_n\}_{n\in\mathbb{N}}$ is the WSR. Then

$$(n-1) - \log_2(1/\varepsilon) < t_{\text{mix}}^{(n)}(1-\varepsilon) \leq n-1, \quad t_{\text{mix}}^{(n)}(\varepsilon) = n-1 \quad \forall\, n \in \mathbb{N}.$$

- Suppose $\{P_n\}_{n\in\mathbb{N}}$ is the CGB. Then

$$t_{\text{mix}}^{(n)}(1-\varepsilon) = 1 \;\forall\; n \in \mathbb{N} \text{ sufficiently large}, \quad t_{\text{mix}}^{(n)}(\varepsilon) = \Theta(n).$$

Proof. See Appendix D.6. (For the WSR, much of the analysis is taken from [99].) □

The next proposition shows that these polarized cutoff behaviors translate into polarized

perturbation behaviors. First, for the WSR, note we cannot invoke lower bounds from our earlier analysis, since we lack laziness. However, we can prove a stronger result: namely, we can identify an uncountable class of restart perturbations such that $\|\pi_n - \pi_{\alpha_n,\sigma_n}\| \to 1$ (this is a stronger result than previous lower bounds, which only guaranteed one such perturbation). On the other hand, for the CGB, we show that the conclusion of Lemma 5.1 fails, despite all assumptions except pre-cutoff holding. In particular, we have the following:

Proposition 5.2. Let $\varepsilon \in (0, 1/2)$ be independent of n. Then the following hold:

- Suppose $\{P_n\}_{n\in\mathbb{N}}$ is the WSR and $\{\alpha_n\}_{n\in\mathbb{N}} \subset (0,1)$ satisfies $\alpha_n = \Theta(n^{-c_1})$ for some $c_1 \in (0,1)$ independent of n; note $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$ by Proposition 5.1. Furthermore, let $\{\sigma_n\}_{n\in\mathbb{N}}$ satisfy $\sigma_n \in \Delta_{n-1}\ \forall\ n \in \mathbb{N}$, and, for some $c_2 > 1, c_3 > 0$ independent of n,

$$\lim_{n\to\infty} \sum_{i=1}^{\lfloor c_3 \alpha_n^{-c_2} \rfloor} \sigma_n(i) = 0. \tag{5.23}$$

  Then $\lim_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| = 1$.
- Suppose $\{P_n\}_{n\in\mathbb{N}}$ is the CGB and $\{\alpha_n\}_{n\in\mathbb{N}} \subset (0,1)$ satisfies $\lim_{n\to\infty} \alpha_n n = \infty$ and $\limsup_{n\to\infty} \alpha_n = \bar{\alpha} < 1/2$; note $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$ by Proposition 5.1. Then $\forall\ \{\tilde{P}_n\}_{n\in\mathbb{N}}$ s.t. $\tilde{P}_n \in B(P_n, \alpha_n)\ \forall\ n \in \mathbb{N}$, $\limsup_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| \le \bar{\alpha} + 1/2 < 1$.

Proof. See Appendix D.7. □

We have stated Proposition 5.2 in some generality, so it is useful to consider an example. Namely, let $\varepsilon \in (0, 1/2)$ and $\alpha_n = 1/\sqrt{n}\ \forall\ n \in \mathbb{N}$, so that $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$ for both example chains. Then for the WSR, many sequences $\{\sigma_n\}_{n\in\mathbb{N}}$ yield restart perturbations satisfying $\|\pi_n - \pi_{\alpha_n,\sigma_n}\| \to 1$. Some examples (easily verified to satisfy (5.23)) are as follows:

- Uniform restart, i.e. $\sigma_n(i) = 1/n\ \forall\ i \in [n]$.
- "Flipped" stationary restart, i.e. $\sigma_n(i) = \pi_n(n - i + 1)\ \forall\ i \in [n]$.
- Deterministic restart on $\Omega(n^{3/4})$, i.e. $\sigma_n = e_{i_n}$ for some $i_n = \Omega(n^{3/4})$.

In contrast, for this choice of α_n and any perturbation of the CGB, Proposition 5.2 implies that $\limsup_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| \le 1/2$. Thus, while many restart perturbations maximally perturb the WSR, no perturbation (restart or otherwise) can maximally perturb the CGB.

Finally, we summarize the discussion of this section graphically. At left in Figure 5.3, we plot $d_n(t)$ versus t for $n = 2^5$. Note the WSR exhibits a clear cutoff behavior, dropping suddenly from $d_n(n-3) \approx 1$ to $d_n(n-1) = 0$. In contrast, the CGB initially falls from $d_n(0) \approx 1$ to $d_n(1) < 1/2$, after which point $d_n(t)$ decays gradually in t. Hence, roughly speaking, the WSR "makes no progress" towards stationarity until step $n-1$; in contrast, the CGB "makes half its progress" towards stationarity after a single step. At right in Figure

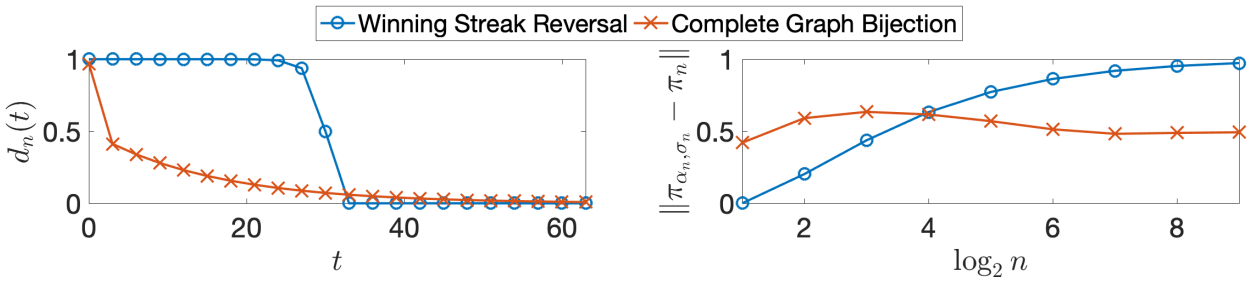


Figure 5.3: Convergence if $n = 2^5$ (left) and perturbation error (right) for WSR and CGB.

5.3, we show the error $\|\pi_n - \pi_{\alpha_n,\sigma_n}\|$ for a certain[5] σ_n and for $\alpha_n = 1/\sqrt{n}$. Note $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$ for both chains, and that restarts occur every $1/\alpha_n = \sqrt{n}$ steps (in expectation). For the WSR, error rapidly increases from ≈ 0 to ≈ 1; for the CGB, error approaches $1/2$.

In short, we can (roughly) say the following to illustrate the intuition of this chapter:

- The WSR requires $n-1$ steps to make any progress to stationarity. Thus, with the perturbed chain restarting every $\sqrt{n}$ steps, it never approaches the original stationary distribution. Consequently, the perturbed chain wanders far from this distribution.
- The CGB makes half its progress to stationarity at time 1. Hence, one step after each restart, the perturbed chain comes close to the original stationary distribution. Consequently, the perturbed chain cannot wander too far from this distribution.

Ultimately, while the cutoff/perturbation connection is perhaps obvious for these chains, this is because their cutoff behaviors lie at opposite extremes among chains with $t_{\text{mix}}^{(n)}(\varepsilon) = \Theta(n)$. The main contribution of this chapter is to extend this connection to a wider class of chains (lazy and reversible), for which it is far less obvious.

5.6 Related work

We now return to discuss the trichotomy results mentioned in the introduction. All of these results concern the directed configuration model (DCM) discussed in Chapter III. It was recently shown that for random walks on the DCM, cutoff occurs at $\Theta(\log n)$ steps [81, 56]. More precisely, [81, 56] prove an analogue of (5.5), namely

$$s < 1 \Rightarrow d_n(st_{\text{ent}}^{(n)}) \xrightarrow[n\to\infty]{\mathbb{P}} 1, \quad s > 1 \Rightarrow d_n(st_{\text{ent}}^{(n)}) \xrightarrow[n\to\infty]{\mathbb{P}} 0, \tag{5.24}$$

where $t_{\text{ent}}^{(n)} = \Theta(\log n)$ is defined in terms of the given degrees and $\xrightarrow{\mathbb{P}}$ denotes convergence in probability. Using these results, Theorem 2 in [21] states that for certain sequences $\{\sigma_n\}_{n\in\mathbb{N}}$, the distance to stationarity $d_{\alpha_n,\sigma_n}(\cdot)$ corresponding to P_{α_n,σ_n} satisfies the following:

[5]Intuitively, one should choose σ_n "far from" π_n. Thus, in Figure 5.3 we let σ_n be uniform for the WSR (since π_n is highly non-uniform) and set $\sigma_n = e_n$ for the CGB (since π_n is roughly uniform).

- If $\alpha_n t_{\text{ent}}^{(n)} \to 0$, (5.24) holds with $d_n(\cdot)$ replaced by $d_{\alpha_n,\sigma_n}(\cdot)$.
- If $\alpha_n t_{\text{ent}}^{(n)} \to \infty$, $d_{\alpha_n,\sigma_n}(s/\alpha_n) \xrightarrow[n\to\infty]{\mathbb{P}} e^{-s}\ \forall\ s > 0$, i.e. $d_{\alpha_n,\sigma_n}(t)$ decays exponentially.
- If $\alpha_n t_{\text{ent}}^{(n)} \to (0,\infty)$, the behavior is intermediate: for $t < t_{\text{ent}}^{(n)}$, $d_{\alpha_n,\sigma_n}(t)$ decays exponentially; for $t > t_{\text{ent}}^{(n)}$, $d_{\alpha_n,\sigma_n}(t) = 0$.

In [20], the authors study a dynamic version of the DCM for which an α_n fraction of edges are randomly sampled and re-paired at each time step. The main result (Theorem 1.4) says the distance to stationarity of the non-backtracking random walk on this dynamic DCM follows a trichotomy similar to the one from [21]. Similar results were also obtained in [98] for a dynamic DCM in which the entire graph is regenerated at random intervals distributed as Geometric(α_n). Finally, in Chapter III we showed the dimensionality of Π_n (the matrix rows $\{\pi_{\alpha_n,e_i}\}_{i\in[n]}$ in the present notation) scales like $O(n^{f(c,\varepsilon)})$ for some $f(c,\varepsilon) \in (0,1)$ if $\alpha_n = c/t_{\text{ent}}^{(n)}$ but scales like $O(1)$ if $\alpha_n t_{\text{ent}}^{(n)} \to 0$; we also conjectured this dimensionality is significantly larger if $\alpha_n t_{\text{ent}}^{(n)} \to \infty$ (see Section 3.7.4).

Ultimately, these results all echo Theorem 5.1 and hint at a deeper phenomena. However, prior to this chapter, one may have (erroneously) suspected that such results rely crucially on some property of the DCM, since [21, 20, 98] and Chapter III all study this generative model. In contrast, the present chapter suggests that cutoff is the crucial property. Accordingly, it is unsurprising that the existing trichotomy results rely on the cutoff results from [81, 56].

Theorem 5.2 relates closely to the aforementioned [22]. Here it is shown that mixing cutoff (5.1) is equivalent to "hitting time cutoff". Namely, Theorem 3 in [22] shows that for lazy, reversible, and irreducible chains, (5.1) is equivalent to each of the following:

$$\exists\ \eta \in (0,1/2] \text{ s.t. } t_{\text{hit}}^{(n)}(\eta,\varepsilon) - t_{\text{hit}}^{(n)}(\eta,1-\varepsilon) = o(t_{\text{hit}}^{(n)}(\eta,1/4))\ \forall\ \varepsilon \in (0,1/4), \tag{5.25}$$
$$\exists\ \eta \in (1/2,1) \text{ s.t. } t_{\text{hit}}^{(n)}(\eta,\varepsilon) - t_{\text{hit}}^{(n)}(\eta,1-\varepsilon) = o(t_{\text{hit}}^{(n)}(\eta,1/4))\ \forall\ \varepsilon \in (0,1/4), t_{\text{rel}}^{(n)} = o(t_{\text{mix}}^{(n)}).$$

Here $t_{\text{rel}}^{(n)}$ is the inverse spectral gap of P_n (see (D.1) in Appendix D.1), and $t_{\text{hit}}^{(n)}(\eta,\varepsilon)$ is the first time the chain has visited all sets of stationary measure at least η with probability at least $1-\varepsilon$, from any starting state (see (D.2) in Appendix D.1). Hence, (5.25) roughly says that shortly after "large" sets are reached at all, they are reached with high probability. As discussed in Section 5.1 in Theorem 5.2 nicely complements this result, since both the cutoff notion and the equivalent notion differ.

Finally, we mention some prior work with less immediate connections to our own. First, we note that the basic connection between mixing times and perturbation bounds has been previously been explored; for instance, the line of work [100, 101] derives upper bounds for perturbation error in terms of mixing times. However, the more difficult lower bounds and the precise asymptotic characterization in Theorem 5.1 are (to the best of our knowledge)

new. In the PageRank literature, another relevant paper is [55], which estimates π_{α_n,σ_n} as a mixture of σ_n and the degree distribution; in this sense, the results in [55] are more precise than ours, but they are restricted to a certain class of P_n.

5.7 Conclusions and future directions

In this chapter, we showed that the relative asymptotics of restart probability and mixing time fully characterize the asymptotic change in stationary distribution for restart perturbations of lazy, reversible chains. We also showed that a certain notion of perturbation sensitivity is (almost) equivalent to pre-cutoff. Together, these results illustrate that how "sharply" a chain converges to stationarity is intimately related to how robust it is.

There are several immediate extensions of this chapter. An obvious one is to extend the results beyond lazy, reversible chains. Note the upper bounds in Lemmas 5.1 and 5.2 do not require laziness or reversibility, so this would only require generalizing the lower bounds of these lemmas. Here the main challenge would be generalizing the lemma from [22] discussed after (5.9). Another avenue to pursue is to extend the results to the wider class of perturbations $B(P_n, \alpha_n)$ defined by (5.6). Note Lemma 5.1 already holds for such perturbations; moreover, the lower bound in Lemma 5.2 establishes existence of a restart perturbation and thus existence of a perturbation in $B(P_n, \alpha_n)$. Hence, the only challenge is to extend the upper bound in Lemma 5.2 to $B(P_n, \alpha_n)$. One approach would be to show that restart perturbations drive the perturbed chain furthest from stationarity, after which the existing upper bound would immediately extend to $B(P_n, \alpha_n)$. We are unsure if this actually holds, but intuition suggests it might, since the worst restart perturbation drives the chain to a particularly bad part of the state space (see discussion after (5.9)).

CHAPTER VI

Local Non-Bayesian Social Learning with Stubborn Agents[1]

6.1 Introduction

With the rise of social networks like Twitter and Facebook, people increasingly receive news through non-traditional sources. For instance, one recent study shows that two-thirds of American adults have gotten news through social media [104]. Such news sources are fundamentally different than traditional ones like print media and television, in the sense that social media users read and discuss news on the same platform. As a consequence, users turning to these platforms for news receive information not only from major publications but from others users as well; in the words of [105], a user "with no track record or reputation can in some cases reach as many readers as Fox News, CNN, or the New York Times." This phenomenon famously reared its head during the 2016 United States presidential election, when fake news stories were shared tens of millions of times [105].

In this chapter, we study a mathematical model describing this situation. The model includes a large number of agents attempting to learn an underlying true state of the world (e.g. which of two candidates is better suited for office) using information from three sources. First, each agent receives noisy observations of the true state, modeling e.g. news stories from major publications. Second, each agent observes the opinions of a subset of other agents, modeling e.g. discussions with other social media users. Third, each agent may observe the opinions of *stubborn agents* or *bots* who aim to persuade others of an erroneous true state, modeling e.g. users spreading fake news.[2] Based on this information, agents iteratively update their beliefs about the true state in a manner similar to the non-Bayesian social learning model of Jadbabaie *et al.* [24]. This iterative process continues for a finite number

[1]This chapter is adapted from [102]. A preliminary version appeared in the abstract [103].

[2]The term *stubborn agents* has been used in the literature to describe such agents; the term *bots* is used in reference to automated social media accounts spreading fake news while masquerading as real users [106].

of iterations that we refer to as the learning horizon.

Under this model, two competing forces emerge as the learning horizon grows. On the one hand, agents receive more observations of the true state, suggesting that they become more likely to learn. On the other hand, the opinions of the bots gradually propagate through the system, suggesting that agents become increasingly exposed to these opinions and thus less likely to learn. Hence, while a growing horizon clearly affects the learning outcome, the nature of this effect – namely, whether learning becomes more or less likely – is less clear.

This effect of the learning horizon has often been ignored in works with models similar to ours. For example, our model is nearly identical to that in the empirical work [23], in which the authors show that polarized beliefs can arise when there are two types of bots with diametrically opposed viewpoints. However, the experiments in [23] simply fix a large learning horizon and do not consider the effect of varying it. Models similar to ours have also been treated analytically; for example, [24, 26, 25] study non-Bayesian learning models similar to ours. However, these works consider a fixed number of agents and an infinite learning horizon and thus also ignore timescale effects.

In our first set of results (see Section 6.3), we argue that *the learning horizon plays a prominent role in the learning outcome and therefore should not be ignored*. In particular, we show that the learning outcome depends on the relationship between the horizon T_n and a quantity p_n that describes the "density" of bots in the system, where both quantities may depend on the number of agents n. Mathematically, letting $\theta \in (0,1)$ denote the true state and $\theta_{T_n}(i^*)$ denote the belief about the true state for a uniformly random agent i^* at the horizon T_n, we show (see Theorem 6.1)

$$\theta_{T_n}(i^*) \xrightarrow[n\to\infty]{\mathbb{P}} \begin{cases} \theta, & T_n(1-p_n) \xrightarrow[n\to\infty]{} 0 \\ 0, & T_n(1-p_n) \xrightarrow[n\to\infty]{} \infty \end{cases}. \tag{6.1}$$

Here p_n is smaller when more bots are present and 0 is the erroneous true state promoted by the bots. Hence, in words, (6.1) says the following: if there are sufficiently few bots, in the sense that $T_n(1-p_n) \to 0$, i^* learns the true state; if there are sufficiently many bots, in the sense that $T_n(1-p_n) \to \infty$, i^* adopts the extreme belief 0 promoted by the bots.

We note the result in (6.1) assumes a particular model for the graph connecting agents and bots (a modification of the directed configuration model used in Chapter III). For such models, *phase transitions* – wherein small changes to model parameters lead to starkly different behaviors – are often observed. In this case, assuming $T_n = (1-p_n)^{-k}$ for some $k > 0$ and $p_n \to 1$, the limiting belief suddenly drops from θ to 0 as k changes from e.g. 0.99 to 1.01 (see Figure 6.1). Put differently, agents initially (at time $(1-p_n)^{-0.99}$) learn the true

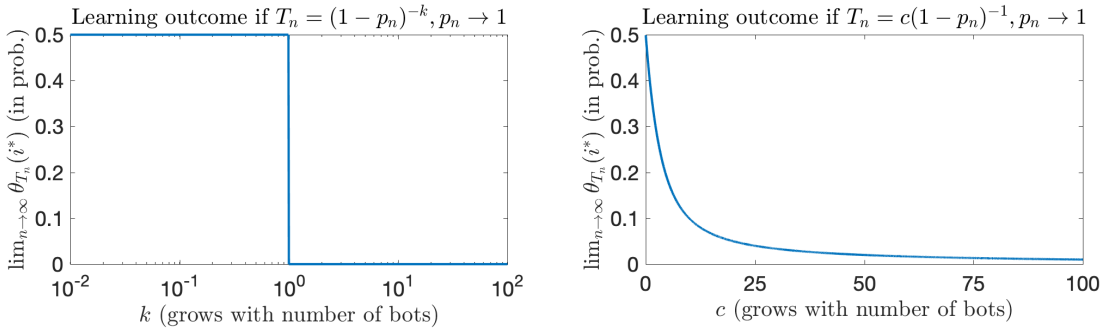


Figure 6.1: Graphical illustration of learning outcome in the case $\eta = \theta = 0.5$.

state, then suddenly (at time $(1-p_n)^{-1.01}$) "forget" the true state and adopt the extreme opinion 0. In light of this, it is interesting to set $k = 1$ and "zoom in" to study the dynamics of this drop from θ to 0. Indeed, in Theorem 6.1, we show that if $T_n(1-p_n) \to c \in (0,\infty)$,

$$\theta_{T_n}(i^*) \xrightarrow[n\to\infty]{\mathbb{P}} \theta(1-e^{-c\eta})/(c\eta), \tag{6.2}$$

where $\eta \in (0,1)$ is a model parameter that dictates the weight agents place on other agents' opinions in their belief updates. The limit in (6.2) is depicted graphically as a function of c in Figure 6.1, which offers an intuitive interpretation: if an adversary deploys bots in hopes of driving agent opinions to 0, the marginal benefit of deploying additional bots is smaller when c is larger. In short, the adversary experiences *diminishing returns*.

To conclude the first part of the chapter, we show in Theorem 6.2 that all but $o(n)$ agents adopt opinion 0 in a certain sub-case of $T_n(1-p_n) \to \infty$ (namely, the sub-case in which the "density" of bots is non-vanishing). Hence, Theorem 6.2 is stronger than Theorem 6.1 and applies to fewer cases; we argue empirically that this stronger result fails in other cases.

Our second set of results (see Section 6.4) consider a setting in which an adversary deploys bots in hopes of disrupting learning. More specifically, the adversary chooses how many bots to connect to each agent (subject to a budget constraint), with the aim of minimizing $\theta_{T_n}(i^*)$. Here we leverage our first set of results to formulate the adversary's problem as an integer program: by (6.1) and (6.2), an adversary can minimize beliefs (at least asymptotically) by minimizing p_n, viewed as a function of the number of bots connected to each agent.

Even after recasting the adversary's problem as an integer program, it remains unclear if it can be solved efficiently. Thus, in Section 6.4, we propose two solutions to this problem. First, we show its objective function belongs to a special class of discrete-domain functions that can be minimized in polynomial time, and we employ an existing algorithm to solve the integer program exactly. However, this runtime is n^2 even in the best case, which too high for social networks like Twitter and Facebook (where n is on the order of 10^8). Thus, we also

propose a randomized approximation algorithm that runs in time $n \log n$ and that produces a constant-fraction approximation of the optimal $1 - p_n$ with high probability (see Theorem 6.4). Using this constant-fraction result, as well as our analysis from the first part of the chapter, we (roughly) show the following: if the most sophisticated adversary can drive the typical belief to 0 (in the sense of (6.1)), then our randomized scheme will drive the typical belief to 0 as well. See Corollary 6.1 for a formal statement.

While the exact solution can only be found algorithmically, our randomized scheme has a somewhat interpretable, and quite interesting, form. In particular, it suggests that *successful adversaries carefully balance agents' influence and susceptibility to influence*. For a social network like Twitter, this means targeting users with many followers (i.e. influential users) who follow very few users themselves (so that fake news tweeted by bots will appear prominently in the targeted users' Twitter feeds). While this is somewhat intuitive, the precise form of the randomized scheme is far from obvious. Thus, we believe our analysis provides new insights into vulnerabilities of news sharing platforms and social learning models.

Finally, we show empirically that our proposed adversary solutions outperform (in terms of minimizing $\theta_{T_n}(i^*)$) a number of intuitive heuristics on graphs representing real social networks. This is somewhat remarkable, because our adversary solutions fundamentally assume that minimizing $\theta_{T_n}(i^*)$ amounts to minimizing p_n, and we only verify this assumption for a certain random graph model (and only in the limit as n grows to infinity). Thus, our empirical results suggest that our insights regarding vulnerabilities in news sharing and social learning extend beyond the random graph model considered in the rest of the chapter.

Before proceeding, we note several of our results also assume $T_n = O(\log n)$, which guarantees that at the learning horizon, an agent's belief is only affected by a vanishing fraction of other agents and bots (at least in the sparse random graph considered). This is why the title of the chapter refers to the learning as "local". More specifically, our choice of T_n is dominated by the mixing time of the random walk on this random graph, which means we cannot leverage global properties like the stationary distribution of this walk, in contrast to many works on social learning (see Section 6.5). Instead, similar to the analysis in Chapter III (see the discussion following Lemma 3.1), we leverage the fact that the random graph has a well-behaved local structure, and we show that analyzing beliefs amounts to analyzing the probability of reaching the absorbing states representing bots. It is from three regimes of these absorption probabilities that the three regimes in (6.1)-(6.2) arise (see Section 6.3.2). Furthermore, we can view the introduction of these absorbing states as a perturbation of the random walk on the agent sub-graph; from a learning perspective, this perturbation may cause learning to fail where it may have succeeded. Thus, this chapter studies a perturbed Markov chain for which the perturbation causes adverse effects.

The remainder of the chapter is organized as follows. In Section 6.2, we define the model studied in this chapter. We present our results concerning the learning outcome in Section 6.3. In Section 6.4, we discuss the adversarial setting. Finally, we discuss related work in Section 6.5 and conclude the chapter in Section 6.6.

Notational conventions for the chapter: Most notation is standard or defined as needed, but we note here that the following conventions are used frequently. For $n \in \mathbb{N}$, we let $[n] = \{1, 2, \ldots, n\}$, and for $n, k \in \mathbb{N}$ we let $k + [n] = [n] + k = \{k + 1, k + 2, \ldots, k + n\}$. All vectors are treated as row vectors. We let e_i denote the vector with 1 in the i-th position and 0 elsewhere. We denote the set of nonnegative integers by $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. We use $1(A)$ for the indicator function, i.e. $1(A) = 1$ if A is true and 0 otherwise. All random variables are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with $\mathbb{E}[\cdot] = \int_\Omega \cdot \, d\mathbb{P}$ denoting expectation, $\xrightarrow{\mathbb{P}}$ denoting convergence in probability, and $a.s.$ meaning $\mathbb{P}-$almost surely.

6.2 Model

6.2.1 Learning model

We begin by defining the model studied throughout the chapter. The main ingredients are (1) a true state of the world, represented as a scalar, (2) a social network connecting two sets of nodes, some who aim to learn the true state and some who wish to persuade others of an erroneous state, and (3) a learning horizon. We discuss each of these ingredients in turn.

The true state of the world is a constant $\theta \in (0, 1)$. For example, in an election between candidates representing two political parties (say, Party 1 and Party 2), $\theta \approx 0$ can be interpreted as the Party 1 candidate being far superior, $\theta \approx 1$ means the Party 2 candidate is far superior, and $\theta \approx 0.5$ implies the candidates are roughly equal. We emphasize that θ is a deterministic constant and does not depend on time, nor on the number of agents.

A directed graph $G = (A \cup B, E)$ connects disjoint sets of nodes A and B (details regarding the graph structure are discussed in Section 6.2.2). We refer to elements of A as *regular agents*, or simply *agents*, and elements of B as *stubborn agents* or *bots*. While agents attempt to learn the true state θ, bots aim to disrupt this learning and convince agents that the true state is instead 0. In the election example, agents represent voters who study the two candidates to learn which is superior, while bots are loyal to Party 1 and aim to convince agents that the corresponding candidate is superior (despite possible evidence to the contrary). Edges in the graph represent connections in a social network over which nodes share opinions in a manner that will be described shortly. An edge from node i to node j, denoted $i \to j$, will be interpreted to mean that j observes i's opinion.

Agents and bots share opinions until a learning horizon $T \in \mathbb{N}$. We will allow the horizon to depend on the number of agents $n \triangleq |A|$ and will thus denote it by T_n at times. In the

election example, T represents the duration of the election season, i.e. the number of time units that agents can learn about the candidates and that bots can attempt to convince agents of the superiority of the Party 1 candidate. Here T_n will be finite for each finite n, and we will let T_n tend to infinity with n.

It remains to specify how agents attempt to learn and how bots aim to disrupt learning. We begin with the agents. Initially, $i \in A$ believes the state to be $\theta_0(i) = \alpha_0(i)/(\alpha_0(i)+\beta_0(i))$, where $\alpha_0(i) \in [0, \bar{\alpha}]$ and $\beta_0(i) \in [0, \bar{\beta}]$ for some $\bar{\alpha}, \bar{\beta} \in (0, \infty)$ that do not depend on n (if $\alpha_0(i) = \beta_0(i) = 0$, we let $\theta_0(i) = 0.5$ by convention). We refer to $\alpha_0(i), \beta_0(i)$ as the *prior parameters* and will not specify them beyond assuming they lie in the aforementioned intervals.[3] In our running example, the initial belief $\theta_0(i)$ can encode i's past opinions regarding the political parties, e.g. $\theta_0(i) < 0.5$ means i historically prefers Party 1 and is predisposed towards the corresponding candidate before the election season begins. At each time $t \in [T]$, $i \in A$ receives a noisy observation of the true state (e.g. i reads a news story regarding the candidates) and modifies its opinion based on this observation and on the opinions of its incoming neighbors in G (e.g. i discusses the election with its social connections). Mathematically, $i \in A$ updates its belief as $\theta_t(i) = \alpha_t(i)/(\alpha_t(i) + \beta_t(i))$, where

$$\alpha_t(i) = (1-\eta)(\alpha_{t-1}(i) + s_t(i)) + \frac{\eta}{d_{in}(i)} \sum_{j \in N_{in}(i)} \alpha_{t-1}(j), \tag{6.3}$$
$$\beta_t(i) = (1-\eta)(\beta_{t-1}(i) + (1 - s_t(i))) + \frac{\eta}{d_{in}(i)} \sum_{j \in N_{in}(i)} \beta_{t-1}(j).$$

Here $s_t(i) \sim \text{Bernoulli}(\theta)$ is the observation of the true state, $N_{in}(i) \subset A \cup B$ are i's incoming neighbors in G, $d_{in}(i) = |N_{in}(i)|$, and $\eta \in (0, 1)$ is a constant (independent of agent i and time t). We note that, as η grows, the effect of the network becomes stronger (i.e. the opinions of agent i's neighbors have a stronger effect on i); this will be reflected in our results. Also, as discussed in Section 6.2.2, we will assume $d_{in}(i) > 0 \; \forall \; i \in A$, so (6.3) is well-defined.

Before discussing the bots, we comment further on the belief update (6.3). First, assuming $\eta = \alpha_0(i) = \beta_0(i) = 0$ temporarily, we simply have $\theta_t(i) = \sum_{\tau=1}^{t} s_\tau(i)/t$, which is an unbiased estimate of the true state θ. Next, if we drop the assumption $\alpha_0(i) = \beta_0(i) = 0$ (but still assume $\eta = 0$), $\theta_t(i)$ is no longer an unbiased estimate. Instead, we can view $\theta_t(i)$ as the mean of a beta distribution with parameters $\alpha_t(i), \beta_t(i)$; in this case, (6.3) is simply a Bayesian update of the prior distribution $\text{Beta}(\alpha_{t-1}(i), \beta_{t-1}(i))$ with a $\text{Bernoulli}(\theta)$ signal. Finally, dropping the assumption $\eta \neq 0$ to obtain the model we actually consider, (6.3) is no longer a Bayesian update, as alluded to by the title of this chapter. This non-Bayesian model is closely related to others in the literature; see Section 6.5 for details.

[3] Appendix E.1.1 shows the effect of the priors vanishes when $T_n \to \infty$, so specifying them is unnecessary.

Having specified the behavior of agents, we turn to the bots. For $i \in B$, we simply set

$$\alpha_t(i) = 0, \quad \beta_t(i) = \bar{\beta} + (1-\eta)t \quad \forall\, t \in [T]. \tag{6.4}$$

Hence, the opinion of $i \in B$ is $\theta_t(i) = \alpha_t(i)/(\alpha_t(i) + \beta_t(i)) = 0$, e.g. bots believe the Party 1 candidate is far superior. To explain the precise form of (6.4), consider a system composed of only agents (i.e. $B = \emptyset$). Since $\beta_0(i) \le \bar{\beta}, s_t(i) \ge 0\ \forall\ i \in A$, it is easy to show via (6.3) that $\beta_t(i) \le \bar{\beta} + (1-\eta)t$ and $\alpha_t(i) \ge 0\ \forall\ i \in A, t \in [T]$. Hence, not only are bots biased towards state 0, but their bias is maximal, in the sense that their parameters $\alpha_t(i), \beta_t(i)$ are as extreme as an agent's can be. We also note we can define bot in an alternative way that will be more convenient for our analysis. Specifically, for $i \in B$, we can set $N_{in}(i) = \{i\}$ (i.e. i has a self-loop and no other incoming edges in G), $\alpha_0(i) = 0$, $\beta_0(i) = \bar{\beta}$, and $s_t(i) = 0\ \forall\ t \in [T]$. Then, assuming $i \in B$ updates its parameters via (6.3), it is straightforward to show (6.4) holds. This alternative definition will be used for the remainder of the chapter. Finally, since all bots $i \in B$ have the same behavior, we assume (without loss of generality) that the outgoing neighbor set of $i \in B$ is $N_{out}(i) = \{i, i'\}$ for some $i' \in A$, i.e. in addition to its self-loop, each bot has a single outgoing neighbor from A.

6.2.2 Graph model

We next specify how the social network G is constructed. For this, we use a modification of the directed configuration model (DCM) from Chapter III. Our modification is needed to account for the distinct node types at hand (agents and bots).

To begin, we realize a random degree sequence $\{d_{out}(i), d_{in}^A(i), d_{in}^B(i)\}_{i \in A}$ from some distribution; here we let $A = [n]$. In the construction described next, $i \in A$ will have $d_{out}(i)$ outgoing neighbors (i will be observed by $d_{out}(i)$ other agents), along with $d_{in}^A(i)$ and $d_{in}^B(i)$ incoming neighbors from A and B, respectively (i will observe $d_{in}^A(i)$ agents and $d_{in}^B(i)$ bots). The total in-degree of i is thus $d_{in}(i) = d_{in}^A(i) + d_{in}^B(i)$. We will assume

$$d_{out}(i), d_{in}^A(i) \in \mathbb{N}, d_{in}^B(i) \in \mathbb{N}_0\ \forall\ i \in A, \quad \sum_{i \in A} d_{out}(i) = \sum_{i \in A} d_{in}^A(i). \tag{6.5}$$

In words, the first condition says i is observed by and observes at least one agent, and may observe by one or more bots. The second condition says sum out-degree equals sum in-degree in the agent sub-graph; this will be necessary to construct a graph with the given degrees.

After realizing the degree sequence, we begin the graph construction.[4] First, we attach $d_{out}(i)$ outgoing half-edges, $d_{in}^A(i)$ incoming half-edges labeled A, and $d_{in}^B(i)$ incoming half-edges labeled B, to each $i \in A$; we will refer to these half-edges as *outstubs*, A-*instubs*, and

[4]This construction is presented more formally as Algorithm E.1 in Appendix E.1.1.

B-instubs, respectively. We let O_A denote the set of all outstubs. We then pair each outstub with an A-instub to form an agent sub-graph in the following breadth-first-search manner:

- Sample i^* from A uniformly. For each of the $d_{in}^A(i^*)$ A-instubs attached to i^*, sample an outstub uniformly from O_A (resampling if the sampled outstub has already been paired), and connect the instub and outstub to form an edge from some agent to i^*.
- Let $A_1 = \{i \in A \setminus \{i^*\} : \text{an outstub of } i \text{ was paired with an } A\text{-instub of } i^*\}$. For each $i \in A_1$, pair i's $d_{in}^A(i)$ A-instubs in the same manner i^*'s A-instubs were paired.
- Continue iteratively until all A-instubs have been paired. In particular, during the l-th iteration, we pair all A-instubs attached to A_l, the set of agents at distance l from i^* (by distance l, we mean a path of length l exists, but no shorter path exists).

At the conclusion of this procedure, we obtain an agent sub-graph, along with unpaired B-instubs attached to some (possibly all) agents. It remains to attach these B-instubs to bots. For this, we define $B = n + [\sum_{i \in A} d_{in}^B(i)]$ to be the set of bots (hence, the node set is $A \cup B = [n + \sum_{i \in A} d_{in}^B(i)]$). To each $i \in B$ we add a single self-loop and a single unpaired outstub (as described at the end of Section 6.2.1). This yields $\sum_{i \in A} d_{in}^B(i)$ unpaired outstubs attached to bots. Finally, we pair these outstubs arbitrarily with the $\sum_{i \in A} d_{in}^B(i)$ unpaired B-instubs from above to form edges from bots to agents (note the exact pairing can be arbitrary since all bots behave the same, per Section 6.2.1).

Before proceeding, we note that the pairing of A-instubs with outstubs from O_A did not prohibit us from forming agent self-loops (i.e. edges $i \to i$ for $i \in A$), nor did it prohibit multiple edges from $i \in A$ to $i' \in A$. This second observation means the set of edges E will in general be a multi-set. For this reason, we re-define the parameter update (6.3) as

$$\alpha_t(i) = (1-\eta)(\alpha_{t-1}(i) + s_t(i)) + \eta \sum_{j \in A \cup B} \frac{|\{j' \to i' \in E : j' = j, i' = i\}|}{d_{in}(i)} \alpha_{t-1}(j), \quad (6.6)$$

$$\beta_t(i) = (1-\eta)(\beta_{t-1}(i) + (1 - s_t(i))) + \eta \sum_{j \in A \cup B} \frac{|\{j' \to i' \in E : j' = j, i' = i\}|}{d_{in}(i)} \beta_{t-1}(j),$$

i.e. we weigh i's incoming neighbors proportional to the number of edges pointing to i. We also note that, instead of attaching bots to B-instubs after pairing all A-instubs as described above, we can pair B-instubs iteratively along with the pairing of A-instubs. Finally, in the case $d_{in}^B(i) = 0 \ \forall \ i \in A$, the construction above reduces to the standard DCM.

6.3 Learning outcome

Having defined our model, we turn to our learning outcome analysis. We begin by defining the required assumptions in Section 6.3.1. We then state and discuss our results in Sections 6.3.2 and 6.3.3. Finally, in Section 6.3.4, we return to comment on our assumptions.

6.3.1 Assumptions

To define our assumptions, we require some notation. First, from the given degree sequence $\{d_{out}(i), d_{in}^A(i), d_{in}^B(i)\}_{i\in A=[n]}$, we define, for each $(i,j,k) \in \mathbb{N}\times\mathbb{N}\times\mathbb{N}_0$,

$$f_n^*(i,j,k) = \frac{1}{n}\sum_{a=1}^{n} 1((d_{out}(a), d_{in}^A(a), d_{in}^B(a)) = (i,j,k)), \tag{6.7}$$
$$f_n(i,j,k) = \sum_{a=1}^{n} \frac{d_{out}(a)}{\sum_{a'\in A} d_{out}(a')} 1((d_{out}(a), d_{in}^A(a), d_{in}^B(a)) = (i,j,k)).$$

In words, f_n^* and f_n are the (random) degree distributions for agents sampled uniformly and proportional to out-degree, respectively. Note that, since the first agent i^* added to the graph is sampled uniformly from A, the degrees of i^* are distributed as f_n^*. Also recall that, to pair A-instubs, we sample outstubs uniformly from O_A, resampling if the sampled outstub is already paired. Thus, each time we add a new agent to the graph (besides i^*), its degrees are distributed as f_n. Using these random distributions, we also define

$$\tilde{p}_n^* = \sum_{j\in\mathbb{N},k\in\mathbb{N}_0} \frac{j}{j+k} \sum_{i\in\mathbb{N}} f_n^*(i,j,k), \quad \tilde{p}_n = \sum_{j\in\mathbb{N},k\in\mathbb{N}_0} \frac{j}{j+k} \sum_{i\in\mathbb{N}} f_n(i,j,k), \tag{6.8}$$
$$\tilde{q}_n = \sum_{j\in\mathbb{N},k\in\mathbb{N}_0} \frac{j}{j+k}\frac{1}{j+k} \sum_{i\in\mathbb{N}} f_n(i,j,k).$$

Following the discussion above, $\tilde{p}_n^*$ is the expected value (conditioned on the degree sequence) of the ratio of A-instubs to total instubs for i^*; $\tilde{p}_n$ is the expected value of this same ratio, but for new agents added to the graph (besides i^*). The interpretation of $\tilde{q}_n$ is similar, i.e. the expected ratio of A-instubs to the square of total instubs for new agents (besides i^*). At the end of Section 6.3.2, we discuss why these random variables arise in our analysis.

We now define the four assumptions that are needed to establish our results. Two of these statements require the degree sequence to be well-behaved (with high probability) – specifically, (A1) requires certain moments of the degree sequence to be finite, while (A3) requires $\{\tilde{p}_n\}_{n\in\mathbb{N}}$ to be close to a deterministic sequence $\{p_n\}_{n\in\mathbb{N}}$. The other statements, (A2) and (A4), impose maximum and minimum rates of growth for the learning horizon T_n. In particular, T_n must be finite for each finite n by (A2) and grow to infinity with n by (A4), as mentioned in Section 6.2.1. We defer further discussion to Section 6.3.4.

(A1) $\lim_{n\to\infty} \mathbb{P}(\Omega_{n,1}) = 1$, where, for some $\nu_1, \nu_2, \nu_3, \gamma > 0$ independent of n with $\nu_3 > \nu_1$,

$$\Omega_{n,1} = \left\{ \max\left\{ \left| \frac{\sum_{i=1}^n d_{out}(i)}{n} - \nu_1 \right|, \left| \frac{\sum_{i=1}^n d_{out}(i)^2}{n} - \nu_2 \right|, \left| \frac{\sum_{i=1}^n d_{out}(i) d_{in}^A(i)}{n} - \nu_3 \right| \right\} < n^{-\gamma} \right\}.$$

(A2) $\exists\ N \in \mathbb{N}$ and $\zeta \in (0, 1/2)$ independent of n s.t. $T_n \leq \zeta \log(n)/\log(\nu_3/\nu_1)\ \forall\ n \geq N$.

(A3) $\lim_{n\to\infty} \mathbb{P}(\Omega_{n,2}) = 1$, where, for some $p_n \in [0,1]$ s.t. $\lim_{n\to\infty} p_n = p \in [0,1]$, some $0 \leq \delta_n = o(1/T_n)$, and some $\xi \in (0,1)$ independent of n,

$$\Omega_{n,2} = \{|p_n - \tilde{p}_n| < \delta_n, \tilde{p}_n^* \geq \tilde{p}_n, \tilde{q}_n < 1 - \xi\}\,.$$

(A4) $\lim_{n\to\infty} T_n = \infty$.

6.3.2 General case

We can now present our first result, Theorem 6.1. The theorem states that the belief at time T_n of a uniformly random agent converges in probability as $n \to \infty$. Interestingly, the limit depends only on the relative asymptotics of the time horizon T_n and the quantity p_n defined in (A3). For example, this limit is θ when $T_n(1-p_n) \to 0$; note that $T_n(1-p_n) \to 0$ requires p_n to quickly approach 1 (since $T_n \to \infty$ by (A4)), which by (A3) and (6.8) suggests the number of bots is small. Hence, i^* learns the true state when there are sufficiently few bots. (The other cases can be interpreted similarly.)

Theorem 6.1. Given (A1), (A2), (A3), and (A4), we have for $i^* \sim A$ uniformly,

$$\theta_{T_n}(i^*) \xrightarrow[n\to\infty]{\mathbb{P}} L(p_n) \triangleq \begin{cases} \theta, & T_n(1-p_n) \xrightarrow[n\to\infty]{} 0 \\ \theta(1-e^{-c\eta})/(c\eta), & T_n(1-p_n) \xrightarrow[n\to\infty]{} c \in (0,\infty)\,. \\ 0, & T_n(1-p_n) \xrightarrow[n\to\infty]{} \infty \end{cases} \tag{6.9}$$

Before discussing the proof of the theorem, we make several observations:

- Suppose p_n is fixed and consider varying T_n, e.g. let $p_n = 1 - (\log n)^{-1/2}$ and define $T_{n,1} = (\log n)^{1/4}$ and $T_{n,2} = (\log n)^{3/4}$ (note $T_{n,1}, T_{n,2}$ satisfy (A2), (A4)). Then $T_{n,1}(1-p_n) \to 0$ and $T_{n,2}(1-p_n) \to \infty$, so by Theorem 6.1, the belief of i^* converges to θ at time $T_{n,1}$ and to 0 at time $T_{n,2}$. In words, i^* initially (at time $(\log n)^{1/4}$) learns the state of the world, then later (at time $(\log n)^{3/4}$) forgets it and adopts the bot opinions.
- Alternatively, suppose T_n is fixed and consider varying p_n, e.g. let $p_n = 1 - c/T_n$ for some $c \in (0,\infty)$. Here smaller c implies fewer bots, and Theorem 6.1 says the limiting belief of i^* is a decreasing convex function of c (see Figure 6.1). One interpretation is that, if an adversary deploys bots in hopes of driving agent beliefs to 0, the marginal benefit of deploying additional bots is smaller when c is larger, i.e. the adversary experiences "diminishing returns". It is also worth noting that, since $(1-e^{-c\eta})/(c\eta) \to 1$ as $c \to 0$ and $(1-e^{-c\eta})/(c\eta) \to 0$ as $c \to \infty$, the limiting belief of i^* is continuous in c.
- If $T_n(1-p_n) \to c \in (0,\infty)$, consider the limiting belief of i^* as a function of η. By

Theorem 6.1, this belief tends to θ as $\eta \to 0$ and tends to $(1-e^{-c})/c$ as $\eta \to 1$. This is expected from (6.6): when $\eta = 0$, agents ignore the network (and thus avoid exposure to biased bot opinions) and form opinions based only on unbiased signals; when $\eta = 1$, the opposite is true. Interestingly, though, there is an asymmetry here: when $\eta \to 0$, the belief approaches the $T_n(1-p_n) \to 0$ case, but when $\eta \to 1$, it does *not* approach the $T_n(1-p_n) \to \infty$ case (since $(1-e^{-c})/c > 0$).

- If $p_n \to p < 1$, we must have $T_n(1-p_n) \to \infty$ (since $T_n \to \infty$ by (A4)), and the belief of i^* tends to 0 by Theorem 6.1. Loosely speaking, this says that a necessary condition for learning is that the bots vanish asymptotically (in the sense that $p_n \to 1$).

The proof of Theorem 6.1 is lengthy; for readability, we outline it in Appendix E.1 and defer computational details to Appendix E.2. However, we next present a (non-rigorous) argument to illustrate why the three cases of the limiting belief arise in Theorem 6.1.

At a high level, these three cases arise as follows. First, when $T_n(1-p_n) \to 0$, the "density" of bots within the T_n-step incoming neighborhood of i^* is small. As a consequence, i^* is not exposed to the biased opinions of bots by time T_n and is able to learn the true state (i.e. $\theta_{T_n}(i^*) \to \theta$ in $\mathbb{P}$). On the other hand, when $T_n(1-p_n) \to \infty$, this "density" is large; i^* is exposed to bot opinions and thus adopts them (i.e. $\theta_{T_n}(i^*) \to 0$ in $\mathbb{P}$). Finally, when $T_n(1-p_n) \to c \in (0,\infty)$, the "density" is moderate; i^* does not fully learn, nor does i^* fully adopt bot opinions (i.e. $\theta_{T_n}(i^*) \to \theta(1-e^{-c\eta})/(c\eta)$ in $\mathbb{P}$).

The explanation of the previous paragraph is not at all surprising; what is more subtle is what precisely *density of bots within the T_n-step incoming neighborhood of i^** means. It turns out that the relevant quantity (and what we mean by this "density") is the probability that a random walker exploring this neighborhood reaches the set of bots.

To illustrate this, we consider a random walk $\{X_l\}_{l\in\mathbb{N}}$ that begins at $X_0 = i^*$ and, for $l \geq 0$, chooses X_{l+1} uniformly from all incoming neighbors of X_l (agents and bots); note here that the walk follows edges in the direction opposite their polarity in the graph. For this walk, it is easy to see that, conditioned on $X_l \in A$, $X_{l+1} \in A$ occurs with probability

$$\frac{d_{in}^A(X_l)}{d_{in}^A(X_l) + d_{in}^B(X_l)}. \tag{6.10}$$

Importantly, we can sample this walk as we construct the graph, by choosing which instub of X_{l-1} to follow *before* pairing them. Assuming they are later paired with uniform agent outstubs, and hence connected to agents chosen proportional to out-degree, we can average

(6.10) over the out-degree distribution and conclude $X_{l+1} \in A$ occurs with probability

$$\sum_{a \in A} \frac{d_{in}^A(a)}{d_{in}^A(a) + d_{in}^B(a)} \frac{d_{out}(a)}{\sum_{a' \in A} d_{out}(a')} = \tilde{p}_n. \tag{6.11}$$

Now since bots have a self-loop and no other incoming edges, they are absorbing states, so $X_{T_n} \in A$ if and only if $X_l \in A \; \forall \; l \in [T_n]$; by the argument above, this latter event occurs with probability $\tilde{p}_n^{T_n}$. Since $\tilde{p}_n \approx p_n$ by (A3), we conclude $X_{T_n} \in A$ occurs with probability

$$\tilde{p}_n^{T_n} \approx p_n^{T_n} \approx \left(1 - \frac{\lim_{n\to\infty} T_n(1-p_n)}{T_n}\right)^{T_n} \approx e^{-\lim_{n\to\infty} T_n(1-p_n)}.$$

From this expression, the three regimes of Theorem 6.1 emerge: when $T_n(1-p_n) \to 0$, the random walker remains in the agent set with probability ≈ 1; this corresponds to i^* avoiding exposure to bot opinions and learning the true state. Conversely, $T_n(1-p_n) \to \infty$ means the walker is absorbed into the bot set with probability ≈ 1 and thus adopts bot opinions. Finally, $T_n(1-p_n) \to c \in (0,\infty)$ means the walker stays in the agent set with probability $\approx e^{-c} \in (0,1)$, corresponding to i^* not fully learning nor fully adopting bot opinions.

We note that the actual proof of Theorem 6.1 does not precisely follow this argument. Instead, we locally approximate the graph construction with a certain branching process; we then study random walks on the tree resulting from this branching process.[5] However, the foregoing argument illustrates the basic reason why the three cases of Theorem 6.1 arise.

Finally, we note the argument leading to (6.11) shows why $\tilde{p}_n$ enters into our analysis. The other random variables defined in (6.8) enter similarly. Specifically, $\tilde{p}_n^*$ arises in almost the same manner, but pertains only to the first step of the walk; this distinction arises since the walk starts at i^*, the degrees of which relate to $\tilde{p}_n^*$. On the other hand, $\tilde{q}_n$ arises when we analyze the variance of agent beliefs, which involves studying *two* random walks; by an argument similar to (6.11), the probability of both walks visiting the same agent is

$$\sum_{a \in A} \frac{d_{in}^A(a)}{d_{in}^A(a) + d_{in}^B(a)} \frac{1}{d_{in}^A(a) + d_{in}^B(a)} \frac{d_{out}(a)}{\sum_{a' \in A} d_{out}(a')} = \tilde{q}_n.$$

6.3.3 Special case

While Theorem 6.1 establishes convergence for the belief of a typical agent, a natural question to ask is how many agents have convergent beliefs. Our second result, Theorem 6.2, provides a partial answer to this question. To prove the result, we require slightly stronger

[5] This is necessary because the argument leading to (6.11) assumes instubs are paired with with uniform outstubs, which is not true if resampling of outstubs occurs in the construction from Section 6.2.2.

assumptions than those required for Theorem 6.1 (we will return shortly to comment on why these are needed). First, we strengthen (A1) and (A3) to include particular rates of convergence for the probabilities $\mathbb{P}(\Omega_{n,i}), i \in \{1,2\}$. Second, we strengthen (A4) with a minimum rate at which $T_n \to \infty$ (specifically, $T_n = \Omega(\log n)$). Third, and perhaps most restrictively, we require $p_n \to p < 1$ in (A1). As a result, Theorem 6.2 only applies to the case $T_n(1-p_n) \to \infty$, for which Theorem 6.1 states the belief of a uniform agent converges to zero. In this setting, Theorem 6.2 provides an upper bound on how many agents' beliefs do *not* converge to zero. In particular, this bound is $O(n^k)$ for some $k < 1$.

Theorem 6.2. Assume $\exists\ \kappa, \mu > 0$ and $N' \in \mathbb{N}$ independent of n s.t. the following hold:

- (A1), with $\mathbb{P}(\Omega_{n,1}) = O(n^{-\kappa})$.
- (A2).
- (A3), with $\mathbb{P}(\Omega_{n,2}) = O(n^{-\kappa})$ and $p < 1$.
- (A4), with $T_n \geq \mu \log n\ \forall\ n \geq N'$.

Then for any $\varepsilon > 0$, $k > 1 - \min\{\frac{1}{2} - \zeta, \frac{\mu(\varepsilon\eta(1-p)/\theta)^2}{16}, \kappa\}$, and $K > 0$, all independent of n,

$$\lim_{n\to\infty} \mathbb{P}\left(|\{i \in [n] : \theta_{T_n}(i) > \varepsilon\}| > Kn^k\right) = 0.$$

We reiterate that $\zeta < 1/2$ by (A2) and $\mu, \kappa > 0$ by the theorem statement. Hence, $\min\{\frac{1}{2} - \zeta, \frac{\mu(\varepsilon\eta(1-p)/\theta)^2}{16}, \kappa\} > 0$, so we can choose $k < 1$ in Theorem 6.2 to show that the size of the non-convergent set of agents vanishes relative to n. We suspect that such a result is the best one could hope for; in particular, we suspect that showing *all* agent beliefs converge to zero is impossible. This is in part because our assumptions do not preclude the graph from being disconnected. Hence, there may be small connected components composed of agents but no bots; in such components, agent beliefs will converge to θ (not zero). Additionally, while the lower bound for k in Theorem 6.2 is somewhat unwieldy, certain terms are easily interpretable: the bound sharpens as η grows (i.e. as agents place less weight on their unbiased signals), as p decays (i.e. as the number of bots grows), and as θ decays (i.e. as signals are more likely to be zero, pushing beliefs to zero).

As for Theorem 6.1, the proof of Theorem 6.2 is outlined in Appendix E.1 with details provided in Appendix E.2. The crux of the proof involves obtaining a sufficiently fast rate for the convergence in Theorem 6.1; namely, we show that for some $\gamma > 0$, $\mathbb{P}(\theta_{T_n}(i^*) > \varepsilon) = O(n^{-\gamma})$.[6] At a high level, obtaining such a bound requires bounding three probabilities by $O(n^{-\gamma})$, which also helps explain the stronger assumptions of Theorem 6.2:

[6]One may wonder why we derive a new bound for Theorem 6.2, since we already bounded $\mathbb{P}(\theta_{T_n}(i^*) > \varepsilon)$ for Theorem 6.1. The reason is that the bound for Theorem 6.1 does not decay quickly enough as $n \to \infty$ to prove Theorem 6.2; on the other hand, the bound for Theorem 6.2 does not decay as $n \to \infty$ for the case $T_n(1-p_n) \to [0,\infty)$ and thus cannot be used for all cases of Theorem 6.1.

- As for Theorem 6.1, we first approximate the graph construction with a branching process so as to analyze beliefs on the tree. Here strengthening (A1) with $\mathbb{P}(\Omega_{n,1}) = O(n^{-\kappa})$ is necessary to ensure this approximation fails with probability at most $O(n^{-\gamma})$.
- To analyze beliefs on the tree, we first condition on the random tree structure and treat the belief as a weighted sum of i.i.d. signals using an approach similar to Hoeffding's inequality. Namely, we obtain the Hoeffding-like tail $O(e^{-2\varepsilon^2 T_n})$; strengthening (A4) with $T_n \geq \mu \log n$ is necessary to show this tail is $O(e^{-2\varepsilon^2 \mu \log n}) = O(n^{-\gamma})$.
- Finally, after conditioning on the tree structure, we show this structure is close to its mean. Specifically, letting $\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]$ denote the expected belief for the root node in the tree conditioned on the tree structure (see Appendix E.1 for details), we show

$$\mathbb{P}(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon) = O\left(n^{-\gamma}\right).$$

  Note the only source of randomness in $\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]$ is the tree; because this tree is recursively generated, it has a martingale-like structure that can be analyzed using an approach similar to the proof of Lemma 3.1 in Chapter III. Here we require $\mathbb{P}(\Omega_{n,2}) = O(n^{-\kappa})$ to ensure the degree sequence is ill-behaved with probability at most $O(n^{-\gamma})$; we also require $p_n \to p < 1$ in this step (and only in this step).

We now address the most notable difference between Theorems 6.1 and 6.2; namely, that the latter only applies when $p_n \to p < 1$. We believe this reflects a fundamental distinction between the cases $p_n \to p < 1$ and $p_n \to 1$ and is *not* an artifact of our analysis. An intuitive reason for this is that more bots are present in the former case, so fewer random signals are present (recall bot signals are deterministically zero). As a result, $\theta_{T_n}(i^*)$ is "less random", so its concentration is stronger. Toward a more rigorous explanation, we note that Appendix E.1.4.1 provides the following condition for extending Theorem 6.2 to other cases of p_n:

$$\exists\ \gamma' > 0\ s.t.\ \mathbb{P}(|\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] - L(p_n)| > \varepsilon) = O\left(n^{-\gamma'}\right), \tag{6.12}$$

where $L(p_n)$ is the limit from Theorem 6.1 defined in (6.9). It is the convergence of $|\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] - L(p_n)|$ in (6.12) that we suspect is fundamentally different in the cases $p_n \to p < 1$ and $p_n \to 1$. To illustrate this, we provide empirical results in Figure 6.2. In the leftmost plot, we show $1 - \tilde{p}_n$ versus T_n; here the plot is on a log-log scale, so a line with slope m means $(1 - \tilde{p}_n) \propto T_n^m$. Hence, we are comparing four cases: $m \approx 0$, so that $p_n \approx p < 1$ (blue circles); $m \approx -0.5$, so that $T_n(1 - p_n) \to \infty$ and $p_n \to 1$ (orange squares); $m \approx -1$, so that $T_n(1 - p_n) \to 1$ (yellow diamonds); and $m \approx -1.5$, so that $T_n(1 - p_n) \to 0$ (purple triangles). The second plot reflects the corresponding cases of $L(p_n)$: $\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]$ decays to zero in the first two cases, grows towards $\theta = 0.5$ in the fourth case, and approaches

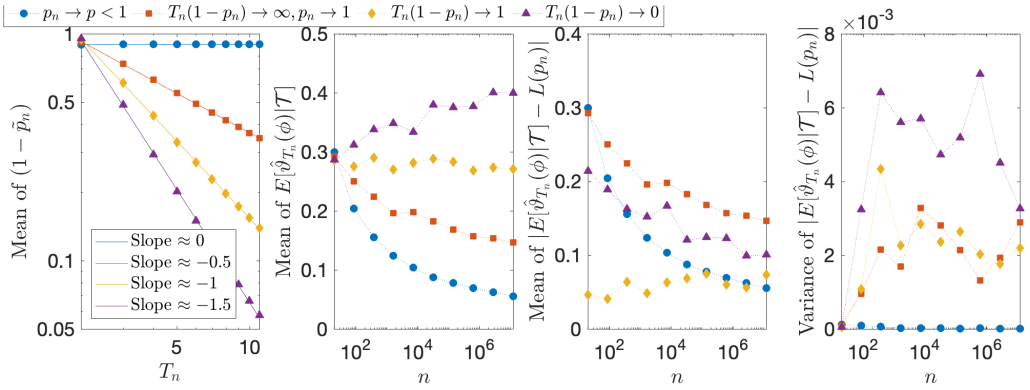


Figure 6.2: Empirical comparison of cases $p_n \to p < 1$, $T_n(1-p_n) \to \infty$ with $p_n \to 1$, $T_n(1-p_n) \to 1$, and $T_n(1-p_n) \to 0$ (leftmost plot). Note $\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]$ approaches the corresponding limit from Theorem 6.1 in all cases (second plot from left). However, the error term $|\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] - L(p_n)|$ behaves markedly differently in the case $p_n \to p < 1$, with a faster decay on average (second plot from right) and a strikingly lower variance (rightmost plot).

an intermediate limit in the third case. The final two plots illustrate the convergence (or lack thereof) in (6.12). Here the empirical mean of the error term $|\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] - L(p_n)|$ decays quickly for the first case but decays more slowly (or is even non-monotonic) in the other cases. More strikingly, the empirical variance of this error term is several orders of magnitude smaller in the first case. This suggests that $\mathbb{P}(|\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] - L(p_n)| > \varepsilon)$ decays much more rapidly in the case $p_n \to p < 1$, which is why we believe this is the only case for which (6.12) is satisfied. (See Appendix E.5 for further details on this experiment.)

6.3.4 Comments on assumptions

We now return to comment on the assumptions needed to prove our results. First, (A1) states that certain empirical moments of the degree distribution – namely, for $i^* \sim A$ uniformly, the first two moments of $d_{out}(i^*)$ and the correlation between $d_{out}(i^*)$ and $d_{in}^A(i^*)$ – converge to finite limits. This says our graph lies in a sparse regime, where typical node degrees do not grow with the number of nodes.[7] We also note $\nu_3 > \nu_1$ in (A1) is minor and simply eliminates an uninteresting case. To see this, first note that when $\Omega_{n,1}$ holds,

$$\frac{\nu_3}{\nu_1} \approx \frac{\sum_{i=1}^n d_{out}(i) d_{in}^A(i)/n}{\sum_{i=1}^n d_{out}(i)/n} = \sum_{i=1}^n \frac{d_{out}(i)}{\sum_{i'=1}^n d_{out}(i')} d_{in}^A(i) \geq 1, \tag{6.13}$$

where we have used the assumed inequality $d_{in}^A(i) \geq 1 \; \forall \, i \in [n]$. Hence, $\nu_3 < \nu_1$ cannot occur, so assuming $\nu_3 > \nu_1$ simply eliminates the case $\nu_3 = \nu_1$. This remaining case is uninteresting

[7] This is analogous to e.g. an Erdős-Rényi model with edge formation probability λ/n for some $\lambda > 0$ independent of n, in which degrees converge in distribution to Poisson(λ) random variables.

because ν_3/ν_1 is the limiting number of offspring for each node in the branching process we analyze; thus, if $\nu_3 = \nu_1$, the tree resulting from this process is simply a line graph.

Next, (A2) states $T_n = O(\log n)$. Together with (A1), these assumptions are standard given our analysis approach (locally approximating the graph construction with a branching process). We also note that, with the interpretation of ν_3/ν_1 above, it follows that the number of agents within the T_n-step neighborhood of i^* can be upper bounded by

$$(\nu_3/\nu_1)^{T_n} = O\left((\nu_3/\nu_1)^{\zeta \log(n)/\log(\nu_3/\nu_1)}\right) = O\left(n^{\zeta}\right) = o(n).$$

In words, the size of the aforementioned neighborhood vanishes relative to n. As mentioned in the introduction, this is why the chapter title refers to the learning as "local": only a vanishing fraction of other agents (those within this neighborhood) affect the belief of i^*.

The remaining statements are needed to establish belief convergence on the tree resulting from the branching process. (A4) states $T_n \to \infty$ with n, which is an obvious requirement for convergence. (A3) essentially says that three events occur with high probability. First, $\tilde{p}_n$ should be close to a convergent, deterministic sequence p_n; this is necessary since the asymptotics of p_n play a prominent role in Theorem 6.1. Second, $\tilde{p}^*_n \geq \tilde{p}_n$ essentially says that bots prefer to attach to agents with higher out-degrees, i.e. more influential agents; this is the behavior one would intuitively expect. Third, $\tilde{q}_n < 1 - \xi \in (0, 1)$ is a minor assumption; for example, if all agents have total in-degree at least 2, $\tilde{q}_n \leq 1/2$.

6.4 Adversarial setting

In the previous two sections, we defined and then analyzed the following model:

(I) A degree sequence $\{d_{out}(i), d^A_{in}(i), d^B_{in}(i)\}_{i\in[n]}$ is realized.
(II) From $\{d_{out}(i), d^A_{in}(i)\}_{i\in[n]}$, an agent sub-graph is constructed.
(III) To each $i \in [n]$, $d^B_{in}(i)$ bots are connected (each also containing a self-loop).
(IV) A learning process occurs on the graph connecting agents and bots.

We next consider an adversarial model, which modifies steps (I) and (III) of this model as follows: in (I), only $\{d_{out}(i), d^A_{in}(i)\}_{i\in[n]}$ is realized, and in (III), an adversary chooses $\{d^B_{in}(i)\}_{i\in[n]}$ (subject to a budget constraint). Put differently, we first construct a graph of agents aiming to learn; an adversary then deploys bots in hopes of disrupting this learning.

In this adversarial model, we will assume the adversary observes $\{d_{out}(i), d^A_{in}(i)\}_{i\in[n]}$ but does *not* observe the agent sub-graph. This is a reasonable assumption for social networks like Twitter, where follower and followee counts (i.e. out- and in-degree) are displayed on each user's profile, but where the actual graph of follower/followee relationships is not publicly available. Under this assumption, our adversarial model can be equivalently defined by replacing (I) with (I') in the model above (but otherwise proceeding as above), where

(I') A sequence $\{d_{out}(i), d_{in}^A(i)\}_{i\in[n]}$ is realized and observed by the adversary, who then chooses $\{d_{in}^B(i)\}_{i\in[n]}$, yielding the degree sequence $\{d_{out}(i), d_{in}^A(i), d_{in}^B(i)\}_{i\in[n]}$.

To be clear, we will assume (as in previous sections) the observed sequence satisfies

$$d_{out}(i), d_{in}^A(i) \in \mathbb{N} \ \forall \ i \in [n], \quad \sum_{i\in[n]} d_{out}(i) = \sum_{i\in[n]} d_{in}^A(i),$$

and the adversary's choice satisfies $d_{in}^B(i) \in \mathbb{N}_0 \ \forall \ i \in [n]$, i.e. the full sequence satisfies (6.5).

The adversary's goal is to disrupt learning, by which we mean minimizing the average belief at the learning horizon $\theta_{T_n}(i^*)$, subject to the budget constraint $\sum_{i=1}^n d_{in}^B(i) \leq b_n$ (here b_n is a given non-negative integer). At first glance, it is far from obvious how the adversary should deploy bots to achieve this goal. Hence, we appeal to Theorem 6.1: to achieve this goal asymptotically, the adversary should deploy bots so as to minimize p_n. In particular, if the adversary can drive p_n to e.g. $1 - T_n^{-1+\varepsilon}$ for some $\varepsilon > 0$, Theorem 6.1 ensures $\theta_{T_n}(i^*) \to 0$ in $\mathbb{P}$ (when our assumptions hold). Furthermore, if the adversary's choice of $\{d_{in}^B(i)\}_{i\in[n]}$ yields a degree sequence $\{d_{out}(i), d_{in}^A(i), d_{in}^B(i)\}_{i\in[n]}$ satisfying the assumptions of Theorem 6.1[8], minimizing p_n is asymptotically equivalent to minimizing $\tilde{p}_n$ (since $|p_n - \tilde{p}_n| \to 0$ with high probability by (A3)). Hence, we will assume the adversary's goal is to minimize $\tilde{p}_n$ after observing the realization of $\{d_{out}(i), d_{in}^A(i)\}_{i\in[n]}$. More concretely, we first let $m_n = \sum_{i=1}^n d_{out}(i)$ and (with slight abuse of notation) define the function $\tilde{p}_n : \mathbb{N}_0^n \to [0, 1]$ by

$$\tilde{p}_n(d) = \sum_{i=1}^n \frac{d_{in}^A(i)}{d_{in}^A(i) + d(i)} \frac{d_{out}(i)}{m_n} \ \forall \ d = (d(1), \ldots, d(n)) \in \mathbb{N}_0^n, \tag{6.14}$$

which is simply $\tilde{p}_n$, as defined in (6.8), viewed as a function of the bot in-degrees $d(i) \triangleq d_{in}^B(i)$[9]. In light of the preceding discussion, we then define the adversary's problem as

$$\min_{d\in\mathbb{N}_0^n} \tilde{p}_n(d) \ s.t. \ \sum_{i=1}^n d(i) \leq b_n. \tag{6.15}$$

While a solution to (6.15) exists, it is not clear if it can be found efficiently, since the minimization is over a discrete set that grows exponentially in n. Thus, in the remainder of this section, we propose two approaches to efficiently solve (or approximate the solution of) (6.15). The first, discussed in Section 6.4.1, employs an existing algorithm for so-called *M-convex minimization*, which computes the solution of (6.15) exactly. When accounting for

[8] We cannot verify these assumptions in general, as the structures of our proposed solutions are not amenable to analysis. Thus, there is a slight gap in our argument that minimizing $\theta_{T_n}(i^*)$ amounts to minimizing $\tilde{p}_n$. However, we show empirically in Section 6.4.3 that $\theta_{T_n}(i^*)$ and $\tilde{p}_n$ are closely correlated in practice.

[9] Here and for the remainder of the section, we suppress the sub- and super-scripts in $d_{in}^B(i)$.

the separable nature of our objective function $\tilde{p}_n$, this approach is somewhat efficient, with computational complexity between $O(n^2)$ and $O(n^2 b_n)$. However, this complexity is too high for social networks like Twitter, where n is on the order of 10^8. Hence, our second approach, discussed in Section 6.4.2, is a randomized algorithm that approximates the solution of (6.15) with $O(n \log n + b_n)$ complexity and provably high accuracy. In addition to these advantages, the randomized scheme has a non-obvious but interpretable form, which provides new insights regarding vulnerabilities in non-Bayesian social learning models.

6.4.1 Exact solution

For the exact solution, we first rewrite (6.15) as $\min_{d \in \mathbb{Z}^n} \hat{p}_n(d)$, where

$$\hat{p}_n(d) = \begin{cases} \tilde{p}_n(d), & d(i) \geq 0 \ \forall \ i \in [n], \sum_{i=1}^n d(i) = b_n \\ \infty, & \text{otherwise} \end{cases}.$$

In words, we incorporated the constraints from (6.15) into the objective; we also used the fact that the solution of (6.15) satisfies the budget constraint with equality. In this equivalent problem, the objective $\hat{p}_n$ belongs to a special class of discrete-domain functions that can be efficiently minimized. This class is the set of *M-convex* functions, defined as follows.

Definition 6.1. [107, Section 1.4.2] Let $f : \mathbb{Z}^n \to \mathbb{R} \cup \{\infty\}$ be a function with *effective domain* $dom(f) = \{x \in \mathbb{Z}^n : f(x) \in \mathbb{R}\}$. Then f is called *M-convex* if for any $x, y \in dom(f)$ and any $i \in [n]$ satisfying $x(i) > y(i)$, there exists $j \in [n]$ satisfying

$$y(j) > x(j), \quad f(x) + f(y) \geq f(x - e_i + e_j) + f(y + e_i - e_j).$$

To verify that our objective is M-convex, first note that by definition,

$$dom\,(\hat{p}_n) = \left\{ d \in \mathbb{Z}^n : d(i) \geq 0 \ \forall \ i \in [n], \sum_{i=1}^n d(i) = b_n \right\}.$$

Now let $d, d' \in dom(\hat{p}_n), i \in [n]$ s.t. $d(i) > d'(i)$. Then since $\sum_{k=1}^n d(k) = \sum_{k=1}^n d'(k) = b_n$, we have $d'(j) > d(j)$ for some $j \in [n]$. From $\sum_{k=1}^n d(k) = \sum_{k=1}^n d'(k) = b_n$ and $d(i), d'(j) \geq 1$, it is also clear that $d - e_i + e_j, d' + e_i - e_j \in dom(\hat{p}_n)$. Hence, letting $\mu(k) = d_{out}(k) d_{in}^A(k)/m_n$,

$$\begin{aligned} \hat{p}_n(d - e_i + e_j) &= \sum_{k \in [n] \setminus \{i,j\}} \frac{\mu(k)}{d_{in}^A(k) + d(k)} + \frac{\mu(i)}{d_{in}^A(i) + d(i) - 1} + \frac{\mu(j)}{d_{in}^A(j) + d(j) + 1} \qquad (6.16) \\ &= \hat{p}_n(d) + \frac{\mu(i)}{(d_{in}^A(i) + d(i) - 1)(d_{in}^A(i) + d(i))} - \frac{\mu(j)}{(d_{in}^A(j) + d(j) + 1)(d_{in}^A(j) + d(j))}, \end{aligned}$$

where we have simply used the definitions of $\hat{p}_n, \tilde{p}_n$. Similarly, we obtain

$$\hat{p}_n(d' + e_i - e_j) = \hat{p}_n(d') - \frac{\mu(i)}{(d^A_{in}(i) + d'(i) + 1)(d^A_{in}(i) + d'(i))} + \frac{\mu(j)}{(d^A_{in}(j) + d'(j) - 1)(d^A_{in}(j) + d'(j))}.$$

Adding the previous two equations, and using the inequalities $d(i) \geq d'(i)+1, d'(j) \geq d(j)+1$ (where the first holds since $d(i) > d'(i)$ and $d(i), d'(i) \in \mathbb{Z}$, and the second holds similarly) gives $\hat{p}_n(d - e_i + e_j) + \hat{p}_n(d' + e_i - e_j) \leq \hat{p}_n(d) + \hat{p}_n(d')$, i.e. $\hat{p}_n$ is M-convex.

Any M-convex function f satisfies the following optimality criterion, which says x minimizes f if and only if f cannot be decreased by "exchanging" x by $x - e_i + e_j$.

Theorem 6.3. [107, Theorem 6.26] Let f be M-convex, and let $x \in dom(f)$. Then

$$f(x) \leq f(y) \ \forall \ y \in \mathbb{Z}^n \quad \Leftrightarrow \quad f(x) \leq f(x - e_i + e_j) \ \forall \ i, j \in [n].$$

From Theorem 6.3, our exact solution emerges: we begin with an initial bot deployment d; we then iteratively "exchange" bots and check whether or not the objective has decreased. More formally, our exact solution is Algorithm 6.1; it is taken from [107, Section 10.1.1], where it is called *steepest descent*. Note that the algorithm terminates when the optimality criterion of Theorem 6.3 is satisfied; thus, Algorithm 6.1 provides an exact solution of of (6.15).

We offer several remarks on the algorithm's complexity:

- Line 3 dominates each iteration's complexity. Naively, this requires $O(n)$ time per i, j, so each iteration's complexity is $O(n^3)$. However, by (6.16), we can accelerate this by computing $\hat{p}_n(d - e_i + e_j)$ in $O(1)$ time, which yields $O(n^2)$ complexity per iteration.
- In the best case, the initial choice of d is actually a solution. However, it still requires one iteration to verify this, so the best-case complexity is $O(n^2)$.
- In the general case, [107, Section 10.1.1] provides a tie-breaking rule for the choice of (i^*, j^*) that guarantees termination in $\max\{\|d - d'\|_1 : d, d' \in dom(\hat{p}_n)\} = O(b_n)$ iterations. Furthermore, this tie-breaking does not increase each iteration's runtime (in an order sense). Thus, for an arbitrary choice of initial d, the complexity is $O(n^2 b_n)$.

6.4.2 Approximation algorithm

We now turn to our approximation algorithm. The idea to first solve the relaxed problem

$$\min_{d \in \mathbb{R}^n_+} \tilde{p}_n(d) \ s.t. \ \sum_{i=1}^{n} d(i) \leq b_n, \tag{6.17}$$

Algorithm 6.1: Exact solution of (6.15)

Let $d \in dom(\hat{p}_n)$, compute $\hat{p}_n(d)$ (in practice, we use a rounded version of the relaxed solution (6.18))
while 1 **do**
  Compute $\hat{p}_n(d - e_i + e_j)\ \forall\ i, j \in [n]$ s.t. $i \neq j$ (using $\hat{p}_n(d)$ and (6.16), this requires $O(1)$ time per i, j pair)
  Let $(i^*, j^*) \in \arg\min_{(i,j)\in[n]^2: i\neq j} \hat{p}_n(d - e_i + e_j)$
  if $\hat{p}_n(d) \leq \hat{p}_n(d - e_{i^*} + e_{j^*})$ **then** terminate (d solves (6.15) by Theorem 6.3)
  else Set $d = d - e_{i^*} + e_{j^*}$

Algorithm 6.2: Approximate solution of (6.15)

Compute $d_n^{rel}(i)$ as in (6.18) and set $d_n^{rand}(i) = 0\ \forall\ i \in [n]$
for $j = 1$ **to** b_n **do**
  Sample W_j from the distribution $\frac{d_n^{rel}}{\sum_{k=1}^n d_n^{rel}(k)}$, i.e. $\mathbb{P}(W_j = i) = \frac{d_n^{rel}(i)}{\sum_{k=1}^n d_n^{rel}(k)}\ \forall\ i \in [n]$
Set $d_n^{rand}(i) = \sum_{j=1}^{b_n} 1(W_j = i)\ \forall\ i \in [n]$

and then to sample bot locations in proportion to the relaxed solution. More formally, our approximate solution d_n^{rand} is constructed via Algorithm 6.2. We note that, as shown in Appendix E.3.1, the solution of the relaxed problem (6.17) is

$$d_n^{rel}(i) = d_{in}^A(i) \left(\frac{\sqrt{r(i)}}{h^*} - 1 \right)_+ \quad \forall\ i \in [n], \tag{6.18}$$

where $x_+ = x1(x > 0)$, $r(i) = d_{out}(i)/d_{in}^A(i)\ \forall\ i \in [n]$, $h^* = \max_{x\in\mathbb{R}_+} h(x)$, and

$$h(x) = \frac{\sum_{i\in[n]: r(i)\geq x^2} \sqrt{d_{out}(i) d_{in}^A(i)}}{b_n + \sum_{i\in[n]: r(i)\geq x^2} d_{in}^A(i)}\ \forall\ x \in \mathbb{R}_+. \tag{6.19}$$

While this randomized scheme is somewhat opaque, it in fact yields useful insights. In particular, the randomized and relaxed solutions d_n^{rand} and d_n^{rel} are equal in expectation, and the relaxed solution d_n^{rel} satisfies some intuitive properties:

- $d_n^{rel}(i)$ grows with $r(i) = d_{out}(i)/d_{in}^A(i)$, i.e. the adversary targets i with large $d_{out}(i)$ and small $d_{in}^A(i)$ under the relaxed solution. Here large $d_{out}(i)$ means i is *influential* (e.g. i has many Twitter followers), while small $d_{in}^A(i)$ means i is *susceptible to influence* (e.g. i has few followees, so bot tweets will appear prominently in i's Twitter feed).
- If $r(i) < (h^*)^2$, then $d_n^{rel}(i) = d_n^{rand}(i) = 0$. Hence, if i is sufficiently non-influential, and/or sufficiently non-susceptible, then targeting i gives no value to the adversary.

- If $r(i) = r(j) > (h^*)^2$, the relaxed solution yields

$$\frac{d_{in}^A(i)}{d_{in}^A(i) + d_n^{rel}(i)} = \frac{h^*}{\sqrt{r(i)}} = \frac{h^*}{\sqrt{r(j)}} = \frac{d_{in}^A(j)}{d_{in}^A(j) + d_n^{rel}(j)}.$$

  This can be interpreted as follows: the adversary strives for a similar proportion of fake news in the feeds of users with similar ratios of influence to susceptibility.

In short, our approximate solution balances influence and susceptibility. While somewhat intuitive, the precise manner in which this balance occurs (in particular, the precise form of (6.18)-(6.19)) is highly non-obvious. Thus, in the absence of Theorem 6.1 and the subsequent formulation of the adversary problem (6.15), one would not have arrived at this solution.

We now turn to the analysis of the randomized scheme. For the complexity analysis, first observe that by definition of h, $\{h(x)\}_{x\in\mathbb{R}_+} = \{h(\sqrt{r(i)})\}_{i\in[n]}$. Furthermore, $\{h(\sqrt{r(i)})\}_{i\in[n]}$, and thus $\{h(x)\}_{x\in\mathbb{R}_+}$, can be computed in time $O(n\log n)$ as follows:

- Compute a vector containing $\{r(i)\}_{i\in[n]}$ sorted in decreasing order ($O(n\log n)$ time).
- Iteratively compute the sums in (6.19) at each $x \in \{\sqrt{r(i)}\}_{i\in[n]}$ ($O(n)$ time).
- Compute $\{h(\sqrt{r(i)}) : i \in [n]\}$ ($O(n)$ time).

In summary, $\{h(x)\}_{x\in\mathbb{R}_+}$ (which contains at most n elements) can be computed in $O(n\log n)$ time. After computing this set, h^*, and then d_n^{rel}, can each be computed in linear time. Thus, computing the relaxed solution (6.18) requires $O(n\log n)$ complexity. Finally, assuming we can obtain one sample from d_n^{rel} in $O(1)$ time after $O(n\log n)$ pre-processing time (using e.g. the alias method [108, 109],[110, Section 3.4.1]), Algorithm 6.2 has complexity $O(n\log n+b_n)$.

Analyzing the accuracy of Algorithm 6.2 is more difficult. We will prove a guarantee that says that with high probability, and for any $\delta > 0$

$$\tilde{p}_n\left(d_n^{rand}\right) < \frac{1+\delta+\tilde{p}_n\left(d_n^{opt}\right)}{2+\delta} \Leftrightarrow 1-\tilde{p}_n\left(d_n^{rand}\right) > 1-\frac{1+\delta+\tilde{p}_n\left(d_n^{opt}\right)}{2+\delta} = \frac{1-\tilde{p}_n\left(d_n^{opt}\right)}{2+\delta},$$

where d_n^{opt} is any solution of (6.15), i.e. $1-\tilde{p}_n(d_n^{rand})$ provides a constant-factor approximation of $1-\tilde{p}_n(d_n^{opt})$ with high probability. More formally, we have the following theorem.

Theorem 6.4. Assume the following holds:

$$\exists\ \{x_n\}_{n\in\mathbb{N}} \subset [0,\infty)\ s.t.\ \lim_{n\to\infty} x_n = \infty, \lim_{n\to\infty} \mathbb{P}\left(\frac{m_n(1-\tilde{p}_n(d_n^{opt}))}{\max_{j\in[n]} r(j)} \geq x_n\right) = 1. \tag{6.20}$$

Then for any $\delta > 0$,

$$\lim_{n\to\infty} \mathbb{P}\left(\tilde{p}_n\left(d_n^{rand}\right) \geq \frac{1+\delta+\tilde{p}_n(d_n^{opt})}{2+\delta}\right) = 0.$$

Proof. See Appendix E.3. □

We reiterate that $\tilde{p}_n(d_n^{opt})$ and $\tilde{p}_n(d_n^{rand})$ are both random variables; the former some complicated function of the given (random) degrees $\{d_{out}(i), d_{in}^A(i)\}_{i\in[n]}$, the latter also depending on the random sampling in Algorithm 6.2. To prove Theorem 6.4, we first condition on the given degrees – so that the only randomness is the Algorithm 6.2 sampling – and bound the probability that $\tilde{p}_n(d_n^{rand})$ exceeds $(1+\delta+\tilde{p}_n(d_n^{opt}))/(2+\delta)$ (viewed as a fixed quantity when conditioning on the given degrees). We then average over the realization of $\{d_{out}(i), d_{in}^A(i)\}_{i\in[n]}$ and use (6.20) to show that this tail bound decays in n. In particular, the conditional tail bound almost surely decays in $m_n(1-\tilde{p}_n(d_n^{opt}))/\max_{j\in[n]} r(j)$; this is a complicated function of the given degrees and thus a difficult random variable to understand, so we assume it behaves as in (6.20) to prove concentration of $\tilde{p}_n(d_n^{rand})$.

In light of this proof approach, one may think (6.20) "assumes away" the difficulty of the proof, but we argue that this assumption is in fact minor. Indeed, in the setting of Theorem 6.1, and in particular by (A1), the following occur with high probability:

$$m_n \approx \nu_1 n, \quad \max_{j\in[n]} r(j) \leq \max_{j\in[n]} d_{out}(j) = \max_{j\in[n]} \sqrt{d_{out}(j)^2} < \sqrt{\sum_{j=1}^{n} d_{out}(j)^2} \approx \sqrt{\nu_2 n}.$$

Thus, assumption (6.20) in Theorem 6.4 holds if (A1) holds and with high probability,

$$\sqrt{n}\left(1-\tilde{p}_n\left(d_n^{opt}\right)\right) \xrightarrow[n\to\infty]{} \infty. \tag{6.21}$$

If instead (6.21) fails, assumption (6.20) may fail as well. However, if (6.21) does fail, the $T_n = O(\log n)$ assumption of Theorem 6.1 implies

$$T_n\left(1-\tilde{p}_n\left(d_n^{opt}\right)\right) = O\left(\log n/\sqrt{n}\right) \xrightarrow[n\to\infty]{} 0,$$

which, by Theorem 6.1, suggests that agents successfully learn. In short, (6.20) only eliminates cases in which even the most sophisticated adversary (i.e. one using the optimal strategy) cannot prevent learning. This is an uninteresting case, so (6.20) is minor.

As a corollary of Theorem 6.4 (and of the Theorem 6.1 analysis), we can prove the following. It essentially says that *if the most sophisticated adversary can drive the typical belief to zero, then the randomized scheme will drive the typical belief to zero as well.* It also establishes the reverse implication; while this is intuitively obvious, it requires some work to prove (though no additional effort than is needed to establish the forward implication, so we include it for completeness). Here we write $\theta_t^{opt}(i)$ and $\theta_t^{rand}(i)$, respectively, for the belief of

Table 6.1: Dataset details.

Name	**Description**	n	$\lvert E_n \rvert$
Gnutella	Peer-to-peer network	6,301	20,777
Wiki-Vote	Wikipedia administrator elections	7,115	103,689
Pokec	Slovakian social network	1,632,803	30,622,564
LiveJournal	Blogging platform	4,847,571	68,993,773

agent $i \in [n]$ at time $t \in [T_n]$ in the graphs with bot degrees d_n^{opt} and d_n^{rand}, respectively.

Corollary 6.1. Assume (A1), (A2), (A4), and (6.20) hold. Then for $i^* \sim [n]$ uniformly,

$$\theta_{T_n}^{opt}(i^*) \xrightarrow[n\to\infty]{\mathbb{P}} 0 \quad \Leftrightarrow \quad \theta_{T_n}^{rand}(i^*) \xrightarrow[n\to\infty]{\mathbb{P}} 0.$$

Proof. See Appendix E.4. □

Note the corollary assumes (A1), (A2), and (A4); this allows us to leverage the branching process approximation from the Theorem 6.1 proof. Importantly, these assumptions only involve the learning horizon T_n and the given degrees $\{d_{out}(i), d_{in}^A(i)\}_{i\in[n]}$, *not* the bot degrees $\{d_{in}^B(i)\}_{i\in[n]}$. We again assume (6.20), but as discussed above, this is minor given (A1), (A2).

6.4.3 Empirical results

A fundamental assumption in our adversary solutions is that $\tilde{p}_n$ and $\theta_{T_n}(i^*)$ are correlated, in the sense that minimizing $\tilde{p}_n$ also minimizes $\theta_{T_n}(i^*)$. While Theorem 6.1 states this correlation holds for the random graph model of Section 6.2.2, it is unclear if it holds in practice. To conclude, we present empirical results suggesting that this indeed occurs.

In our experiments, we compare our proposed solutions against some natural heuristics:

- A naive baseline, which uses Algorithm 6.2 but samples each W_j uniformly from $[n]$.
- Three schemes which similarly use Algorithm 6.2, along with the observed degrees: sampling W_j proportional to d_{out} (i.e. targeting influential nodes), d_{in}^A (i.e. targeting susceptible nodes), and d_{out}/d_{in}^A (i.e. naively balancing influence and susceptibility).
- Sampling proportional to PageRank with restart probability ε, denoted PageRank(ε).

We compare our proposed solutions with these heuristics using four datasets from [43], described in Table 6.1. We chose these datasets so we could test our proposed solutions on real social networks of two scales: Gnutella and Wiki-Vote have $n < 10^4$, a scale at which the exact solution Algorithm 6.1 is feasible; Pokec and LiveJournal have $n > 10^6$, a scale that renders Algorithm 6.1 infeasible but that more closely resembles social networks of interest.

For the experiments, we set $\theta = 0.5$ (the case of maximal variance for the signals), $\eta = 0.9$ (to emphasize network effects), and $T_n = 101$ (to ensure the code had reasonable runtime). We set $b_n = \lceil |E_n|/400 \rceil$, so that (roughly) 0.25% of all agent in-edges are connected to bots.

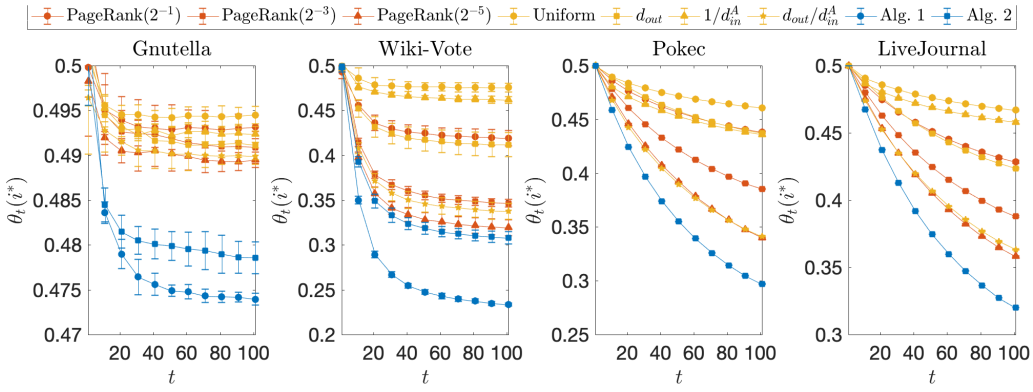


Figure 6.3: Average belief over time when simulating our learning model on real datasets; our proposed solutions (Algorithms 6.1 and 6.2) outperform heuristics, even those using graph structure (i.e. PageRank).

For each graph and for five experimental trials, we chose $\{d_{in}^B(i)\}_{i\in[n]}$ as above, added bots to the original graph accordingly, and simulated the learning process from Section 6.2.1.

In Figure 6.3, we plot the mean and standard deviation (across trials) of $\theta_t(i^*)$ as a function of t (to avoid cluttering the plot, we only show $\theta_t(i^*)$ for $t \in \{1, 11, \ldots, 101 = T_n\}$). For all datasets, our solutions outperform all heuristics, in the sense that our solutions yield the lowest average $\theta_t(i^*)$ for most values of t. More specifically, we note the following:

- Across datasets, our solutions outperform PageRank(ε) for all values of ε. This is quite surprising, since PageRank uses the entire *graph structure*, whereas our solutions only use *degrees*. Also, as ε becomes increasingly smaller, PageRank(ε) performs increasingly better, but this comes at the cost of higher runtime to estimate PageRank(ε).
- Among the heuristics using only degree information, d_{out}/d_{in}^A performs best – though worse than Algorithm 6.2 – across all datasets. Thus, naively balancing influence and susceptibility is not enough; the form of Algorithm 6.2 yields better performance.
- For Gnutella and Wiki-Vote, Algorithm 6.1 outperforms Algorithm 6.2. Though the former is exact and the latter is an approximation, this is still surprising, since it is unclear that these schemes are even optimizing the correct objective for real graphs.

While Figure 6.3 only considers one choice of b_n, we believe our conclusions are robust. In particular, we also tested the cases $b_n = \lceil \tilde{b}|E_n| \rceil$ for each $\tilde{b} \in \{\frac{1}{1600}, \frac{1}{800}, \frac{1}{400}, \frac{1}{200}, \frac{1}{100}\}$, so that between $\approx 0.0625\%$ and $\approx 1\%$ of edges connected to bots (thus, Figure 6.3 shows the middle case $\tilde{b} = \frac{1}{400}$). Appendix E.5 contains a figure analogous to Figure 6.3 for the other choices of $\tilde{b}$; the plots are qualitatively similar. In Figure 6.4, we also summarize this set of results by plotting the final average belief $\theta_{T_n}(i^*)$ as a function of b_n. Generally speaking, the gap between our solutions and the heuristics increases as b_n decreases. Put differently, if an adversary with a limited budget spends this budget intelligently (i.e. using our proposed

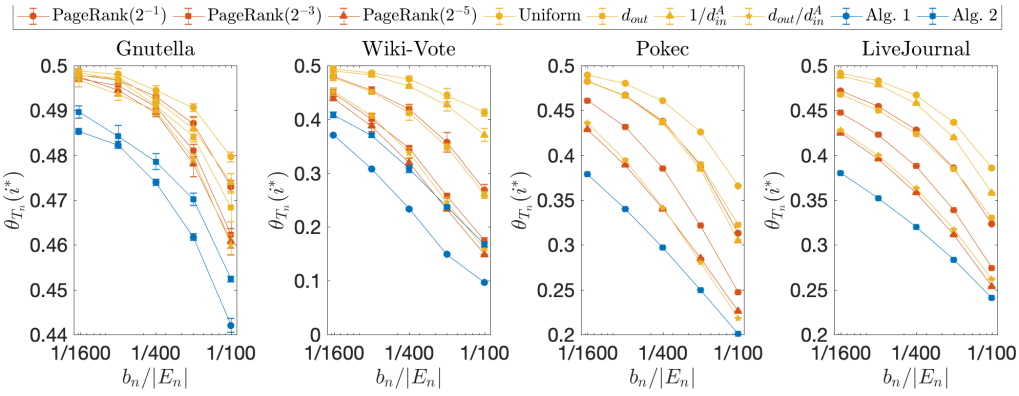


Figure 6.4: Average belief at the learning horizon versus budget on real datasets. Generally, the improvement of our solutions over heuristics increases as b_n decreases.

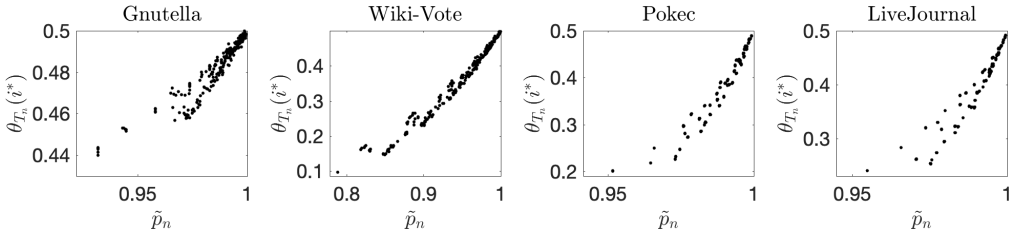


Figure 6.5: As suggested by Figures 6.3 and 6.4, $\theta_{T_n}(i^*)$ and $\tilde{p}_n$ are closely correlated.

solutions), they can still disrupt learning; in contrast, an adversary with a large budget need not be as careful. Also, as expected, $\theta_{T_n}(i^*)$ decreases in b_n in Figure 6.4.

We have thus far shown that our solutions outperform the heuristics. This is somewhat remarkable: our solutions were derived under the fundamental assumption that *minimizing* $\theta_{T_n}(i^*)$ *amounts to minimizing* $\tilde{p}_n$, but we only verified this assumption for a class of random graphs. Thus, our empirical results suggest that even for real social networks, this assumption holds. Indeed, in Figure 6.5 we show scatter plots of $\theta_{T_n}(i^*)$ against $\tilde{p}_n$ (each dot represents one experimental trial). For all datasets, the two quantities are closely correlated.

6.5 Related work

Before closing, we discuss some connections between existing work and this chapter. First, from a modeling perspective, we note our belief update (6.3) is closely related to the non-Bayesian social learning model from [24]. In that model, agent beliefs are distributions over a finite set of possible states of the world (not scalars, as in our model), but belief updates are similar. Specifically, at time t agent i updates its belief $\mu_t(i)$ as

$$\mu_t(i) = \eta_{ii}\mathrm{BU}(\mu_{t-1}(i), \omega_t(i)) + \sum_{j \in N_{in}(i)} \eta_{ij}\mu_{t-1}(j),$$

where $\omega_t(i)$ is the signal received by i at t, $\text{BU}(\mu_{t-1}(i), \omega_t(i))$ means a Bayesian update of the prior belief $\mu_{t-1}(i)$ with the observed signal $\omega_t(i)$, and $\sum_{j \in N_{in}(i) \cup \{i\}} \eta_{ij} = 1$. In [24, Proposition 3], it is shown that, under certain assumptions, including the graph being fixed and strongly connected, these distributions converge to point masses on the true state as $t \to \infty$.

Per the discussion following (6.3) in Section 6.2.1, our model can be viewed as a variant of this one, in which all agents have beta beliefs and Bernoulli signals, communicate parameters of distributions instead of the distributions themselves, and average parameters instead of distributions. From this viewpoint, our quantity of interest $\theta_t(i)$ is simply the mean of agent i's belief. However, the crucial assumptions of strong connectedness and an infinite learning horizon from [24] are violated in our model (the former since bots have self-loops but no other incoming edges; the latter since we take $T_n = O(\log n)$). This necessitates a different analysis, which in turn requires us to simplify the model from [24] by communicating scalars and by taking a simple form of the weights $\{\eta_{ij}\}_{j \in N_{in}(i) \cup \{i\}}$.

We also note our variant of the model from [24] is quite similar to the model in the working paper [23]. In fact, our belief update and inclusion of bots are both taken from this work (with minor differences to bot behavior). However, this work only includes theoretical results in the case $B = \emptyset$; the case $B \neq \emptyset$ is studied empirically. This allows [23] to use a richer model than ours, including a time-varying graph structure, agent-dependent mixture parameters $\sum_{j \in N_{in}(i) \cup \{i\}} \eta_{ij}$, and three types of nodes (bots, agents who observe bots, and agents who do not observe bots). Notably, the empirical results from [23] for the case $B \neq \emptyset$ fix a learning horizon, so the delicate relationship between timescale and bot prevalence that we describe in Theorem 6.1 is not brought to light in [23].

From an analytical perspective, our approach of analyzing beliefs by studying random walks is not new. Perhaps the most obvious example is the classical deGroot model [111], in which agent i updates its (scalar) belief as $\theta_t(i) = \sum_j \theta_{t-1}(j) W(j, i)$ for some column-stochastic matrix W. Collecting beliefs in vector form yields $\theta_t = \theta_0 W^t$, where θ_0 is the vector of initial beliefs. From here, it is clear that beliefs relate closely to random walks, since the i-th column of W^t gives the distribution of a t-step random walk from i on the weighted graph defined by W. This observation has been exploited in the literature; see the surveys [112, Section 3] and [113, Section 4], and the references therein. For example, assuming W is irreducible and aperiodic, and therefore has a well-defined stationary distribution π, [26] establishes conditions for learning using the fact that $\theta_t(i) = \theta_0 W^t e_i^\mathsf{T} \approx \theta_0 \pi^\mathsf{T} \ \forall \ i$ when t is large. Beyond the deGroot model and deGroot-like models such as ours, random walk interpretations have also been leveraged in Bayesian learning models. For example, [114] considers a model for which agents perform a Bayesian update using their own signal but

using the prior of a randomly-chosen neighbor. Exchanging priors with neighbors yields a natural connection to random walks; assuming strong connectedness, the authors exploit the fact that the walk visits every agent infinitely often (i.o.) to derive conditions for learning.

Similar to [24], these works typically assume strong connectedness and long horizons so as to leverage properties such as stationary distributions and i.o. visits, which is a fundamental distinction from this chapter. Indeed, even if we disregard stubborn agents, so that the random walk has a stationary distribution, it does *not* converge within our learning horizon. This is because, as shown in [56], the sparse DCM we consider has mixing time that exceeds

$$\frac{\log n}{\sum_{i\in[n]} \log(d_{in}^A(i)) \frac{d_{out}(i)}{\sum_{i'\in A} d_{out}(i')}} \geq \frac{\log n}{\log(\sum_{i\in[n]} d_{in}^A(i) \frac{d_{out}(i)}{\sum_{i'\in A} d_{out}(i')})} \approx \frac{\log n}{\log(\nu_3/\nu_1)},$$

where the inequality is Jensen's and the approximate equality is (6.13). The final expression exceeds T_n by (A2), i.e. our learning horizon occurs before the underlying random walk mixes. But the situation is in fact more severe, since the random walk on this DCM exhibits cutoff (see Section 5.6). Thus, the T_n-step distribution of this walk can be maximally far from the stationarity. Hence, not only can we not use this stationary distribution, we cannot even use an approximation of it. Again, this means our analysis cannot leverage global properties typically used when relating beliefs to random walks and thus requires a different approach. We also note that our idea to simultaneously construct the graph and sample the walk (as discussed in Section 6.3.2) is taken from [56].

Some other works have considered social learning with stubborn agents. For example, [25] studies a model in which agents meet and either retain their own (scalar) beliefs, adopt the average of their beliefs, or adopt a weighted average; the agent whose belief has a larger weight is called a "forceful" agent. Here the authors show that all agent beliefs converge to a common random variable and study its deviation from the true state. A crucial difference between [25] and this chapter is that [25] assumes even forceful agents occasionally observe other agents' opinions. This yields an underlying Markov chain that is irreducible; the analysis then relies on this chain having a well-defined stationary distribution.

Stubborn agents have also been considered in the consensus setting. This setting is similar to the social learning setting we consider, but instead of asking whether agents learn an underlying state, one asks whether agent beliefs converge to a common value, i.e. a consensus. For example, [115] considers a model in which regular agents adopt weighted averages of beliefs upon meeting other agents (regular or stubborn), while stubborn agents always retain their own beliefs. This intuitively prohibits a consensus from forming; indeed, it is shown that agent beliefs fail to converge, and therefore that disagreement can persist indefinitely. Another example is [116], in which an agent's belief at time $t+1$ is a weighted

average of their own belief at time 0 and their neighbors' beliefs at time t. In this model, stubborn agents place all weight on their own belief from time 0 and thus do not update their beliefs. The analysis in [116] is similar to ours as it relates agent beliefs to hitting probabilities of the stubborn agent set, but it differs as the learning horizon is infinite in [116]. Also in the consensus setting, [117] investigates protocols for robust consensus that may lessen the undesirable effects of stubborn agents in e.g. [115, 116].

6.6 Conclusions and future directions

In this chapter, we analyzed a model for social learning in the presence of stubborn agents. Our learning outcome analysis identified a close relationship between the learning horizon, the "density" of stubborn agents, and the learning outcome. We also considered an adversarial setting which, paired with our learning outcome analysis, yielded insights regarding social learning vulnerabilities.

Several extensions of our learning outcome analysis can be considered. First, it would be useful to generalize our model to allow for agent- and/or time-dependent mixture parameters (i.e. allowing η to vary with i and/or t in (6.6)). Allowing agent dependence suggests a more heterogeneous model in which some agents place more value on private observations, while others place more value on the opinions of their social connections. Allowing time dependence, and specifically allowing η_t to vanish as t grows, suggests a model in which agents become more "set in their ways" over time. Second, one could keep T_n finite for each finite n but allow it to asymptotically dominate our "local" $O(\log n)$ horizon. Here our branching process approximation fails, so this would require a different analysis. However, it would be interesting to see if the three regimes of Theorem 6.1 still hold for such T_n, or if a different phenomenon emerges when global effects of the network take hold.

Each of these extensions of our model would likely yield a different learning outcome, and thus a different objective function in the adversarial setting. Hence, each may require a different analysis to determine optimal or near-optimal bot strategies. Subsequently, each may also yield new and useful insights regarding the sensitivity of the associated models.

CHAPTER VII

Conclusion

Throughout the thesis, we discussed conclusions and immediate extensions of our results on a per-chapter basis; see Sections 2.7, 3.8, 4.4, 5.7, and 6.6. In this chapter, we mention some more holistic takeaways and some less immediate future directions.

7.1 High-level takeaways

Put succinctly, the main ideas developed in this thesis are as follows:

- PPR estimation can be accelerated by exploiting local graph structure. The resulting algorithms are amenable to random graph-based analyses, and if the random graph model is well-chosen, the key insights hold empirically for real graphs.
- Algorithmic and analytical ideas from the PPR literature can be leveraged in a number of different settings. In this thesis, we specifically showed how empirical policy evaluation can be accelerated and how Markov chain perturbations can be analyzed.
- In non-Bayesian social learning, adversaries should carefully balance an agent's influence and susceptibility when deciding whether or not to target the agent. Practically, this may give insights regarding fake news on social networks (see Section 7.2.6).

Given the diversity of the topics considered in this thesis, we lack a unifying conclusion regarding a particular application or real-world problem. However, we next describe some unifying methodological insights provided by this thesis.

7.1.1 Perturbed Markov chains

As discussed in Chapter I, a recurring mathematical object in the thesis was a perturbed Markov chain. In Chapters II, III, and IV, we specifically considered the PPR Markov chain, and we proposed algorithms derived from this chain's power iteration property (1.1) and/or its perfect sampling property (1.4). These properties were fundamentally necessary. For instance, the algorithms in Chapters II and III estimate the PPR chain's stationary distribution but cannot be used to estimate stationary distributions of general chains. Similarly,

the algorithm in Chapter IV relies on the discounted cost objective's connection to PPR and does not immediately apply to other objectives used in reinforcement learning (though we believe an analogue can be derived for the finite horizon objective; see Section 4.4.2).

The special properties of the PPR chain are also useful analytically. The primary example of this thesis arose in Chapter V. For extremely small and extremely large perturbation magnitudes, we were able to characterize the behavior of a general class of perturbations (Lemma 5.1). However, for perturbations of moderate magnitude, the behavior is more subtle, and we needed the closed-form solution of the power iteration (Lemma 5.2).

While the PPR chain did not arise in Chapter VI, a different perturbed Markov chain provided a conceptual approach for our analysis. Namely, we viewed the introduction of bots into our social learning model as a perturbation of a certain random walk relating to agent beliefs (see end of Section 6.1). In contrast to other chapters, we do not believe this viewpoint was fundamentally necessary to analyze the learning outcome; however, it did provide a tractable framework that made the analysis more intuitive. This was especially welcome in Chapter VI, since there we considered a random process unfolding over a random graph, perhaps the most complex mathematical object studied in this thesis.

7.1.2 Concentration of measure

Another recurring mathematical theme was the concentration of measure phenomena – the idea that a function of a large (but finite) set of independent random variables is close to its expected value with high probability. This phenomena was crucial in our analysis of a number of randomized algorithms that estimated expected values of functions by taking many samples and averaging. In Chapters II and IV, the function was simply a sum, so we only used basic Chernoff bounds; in Chapter VI, the function was more complex, and we required the theory of *self-bounding* functions (see Appendix E.3.3).

Concentration of measure arose in a different manner when we analyzed PPR dimensionality in Chapter III and learning outcomes in Chapter VI. We discuss this in some detail as the analysis is similar in both cases and speaks to a more general approach. In both cases, we considered quantities defined on a per-node basis, recursively in terms of the node's neighbors. In Chapter III, this recursion arose from the PPR power iteration (1.1):

$$\pi_s = \alpha e_s^\mathsf{T} + (1-\alpha) e_s^\mathsf{T} P \sum_{t=0}^{\infty} (1-\alpha)^t P^t = \alpha e_s^\mathsf{T} + (1-\alpha) \sum_{v \in V} P(s,v) \pi_v, \tag{7.1}$$

i.e. s's PPR vector can be written in terms of PPR vectors of s's neighbors (those v for which

$P(s, v) > 0$). In Chapter VI, the recursion arose from the belief update

$$\beta_t(i) = (1 - \eta)(\beta_{t-1}(i) + (1 - s_t(i))) + \frac{\eta}{d_{in}(i)} \sum_{j \in N_{in}(i)} \beta_{t-1}(j), \tag{7.2}$$

i.e. i's belief is written in terms of its neighbors' beliefs (see discussion preceding (6.3)). In both cases, the recursive form implied that the quantity could be estimated on a local neighborhood in the graph, and we exploited the concentration phenomena to approximate these neighborhoods with branching processes. Moreover, the recursive nature of the branching processes, paired with the recursive nature of (7.1) and (7.2), yielded martingale-like processes that could be treated with modifications of existing martingale techniques.

We have ignored many details in this discussion; see the proof outlines of Lemma 3.1 and Theorem 6.2 in Appendices B.1 and E.1, respectively, for a more detailed description. The key point is that we believe this analytical approach is more generally useful. In particular, we believe that many quantities defined recursively on sparse random graphs can be estimated in terms of related branching processes, and that the recursion of the branching processes, paired with the recursively-defined quantity, will yield tractable, martingle-like processes.

Finally, we mention that another manifestation of the concentration phenomena involved non-standard norms of random matrices. This arose twice in the thesis: in Section 2.4.1.1, we encountered the $l_{\infty,1}$ norm (sum of column-wise maximums), while in Chapter III, we studied the l_∞ norm (max of row-wise sums). At present, there is less theory regarding concentration in these norms than in more standard ones like the Frobenius or spectral norm [42]. Thus, it may be worth investigating if these non-standard norms have wider utility, and if so, to develop a more formal concentration theory for such norms.

7.2 Future directions and open problems

We close by discussing some future directions. In the spirit of the thesis, we discuss several PPR problems, two applications of PPR ideas, and social learning.

7.2.1 Accuracy criteria for PPR estimation

The PPR estimators discussed in Chapters II and III, along with most all appearing in the literature, assess accuracy in terms of the estimates of PPR *values* – e.g. relative or absolute error of each estimate, l_p error of the vector of estimates, etc. While such guarantees are far more tractable to establish, the *ranking* of these values is more relevant in many of the motivating applications discussed in Section 1.2.3 (ordering of Internet search results or friend recommendations on social networks, for instance). Thus, we believe the "gold standard" of PPR estimators is an efficient algorithm with rigorous guarantees on how far

the true and estimated rankings differ, quantified in terms of some metric on permutations (e.g. Cayley distance, Kendall's tau distance, etc. [118, Ch. 6B]). Such an analysis would be novel in the PPR literature and could be extremely useful in applications.

7.2.2 PPR with non-backtracking random walks

Another significant deviation from the PPR literature would be replacing the underlying random walk of PPR with a non-backtracking one. More precisely, one could consider a Markov chain on a graph $G = (V, E)$ that restarts at $v \in V$ with probability α and takes a non-backtracking random walk step[1] with probability $1 - \alpha$. At a high level, this chain explores the neighborhood of v between restarts in more efficient manner than the standard PPR chain. For example, it may take fewer samples of the Geometric(α)-length trajectories described by (1.4) to obtain an accurate approximation of the corresponding stationary distribution. In fact, it is known that the non-backtracking version of certain chains can mix much faster than the backtracking version (see e.g. [119]), which further supports this conjecture. Thus, we expect the mixing time of this non-backtracking version of PPR would play an important role in the analysis, which may involve connections to our ideas considering mixing times and PPR (namely, those in Section 3.7.4 and Chapter V).

7.2.3 PPR microfoundations

In Section 1.2.2, we discussed the PPR interpretation of "similarity" or "relevance" between nodes. We noted that this is intuitively reasonable, since many graphs exhibit homophily and PPR is a measure of "inverse distance" in networks. Some empirical studies, e.g. [2], have also attempted to justify PPR's use in personalized web search. Nevertheless, there is a lack of theoretical microfoundations for PPR as a relevance/similarity metric. One approach to formalize this mathematically is as follows. Let V be a set of nodes and associate each $v \in V$ with a vector x_v (perhaps generated from some distribution). For instance, V could represent a set of people and x_v could quantify certain characteristics of person v (i.e. age, geographic location, political leanings, employment status, etc.). From $\{x_v\}_{v \in V}$, generate a social network with homophily, perhaps by adding edge (v, w) with probability proportional to $e^{-\|x_v - x_w\|}$ for some norm $\|\cdot\|$ (i.e. v, w are more likely to be friends if they have similar age, location, politics, and jobs). From this random graph model, one could consider the following questions. (1) Are $\|x_v - x_w\|$ and the corresponding PPR value $\Pi(v, w)$ strongly correlated? If so, this would offer justification for PPR as a similarity/relevance metric. (2) If $\|x_v - x_w\|$ and $\Pi(v, w)$ are only weakly correlated (or even uncorrelated), is there a metric that correlates more strongly with $\|x_v - x_w\|$? If so, this would suggest the

[1] By "non-backtracking random walk step," we mean that if the previous state is $u \in V$, the current state is $w \in V$, and the outgoing neighbors of w are $N_{out}(w)$, the next state is chosen randomly from $N_{out}(w) \setminus \{u\}$.

new metric is a more appropriate measure of similarity/relevance than PPR. (3) If a new metric is more appropriate, can it be efficiently estimated for large graphs?

7.2.4 Empirical policy iteration

As discussed in Section 4.2, `Backward-EPE` can reduce the sample complexity of the existing scheme from [19], while attaining the same accuracy in the l_∞ norm. Since l_∞ is the same norm used in [19], our algorithm can be "plugged in" as the EPE subroutine to improve sample complexity bounds for the overall empirical policy iteration (EPI) algorithm discussed in Section 4.1. In contrast, `Bidirectional-EPE` offers a different accuracy guarantee, and thus would necessitate a new analysis if used in EPI. Put roughly, this accuracy guarantee (see Section 4.3) provides a better estimate of the value function for "good" starting states (i.e. states s for which $v_\pi(s)$ is small) than for "bad" starting states (s such that $v_\pi(s)$ is large). This is arguably more natural than an l_∞ guarantee: if s is a particularly bad starting state, there is little utility in accurately estimating $v_\pi(s)$, since we will likely change the policy $\pi(s)$ at the next policy improvement step. Thus, it would be useful to analyze EPI with this EPE subroutine to determine if the overall sample complexity is reduced.

7.2.5 Computational estimation of mixing times

In Markov chain Monte Carlo, one aims to estimate $\mathbb{E}_{X\sim\pi} f(X)$, where f is some function and π is the stationary distribution of some Markov chain $\{X_t\}_{t=0}^{\infty}$ with transition matrix P. Consider the case for which P is unknown but, given a state i, one can obtain samples from $P(i,\cdot)$ (as in Chapter IV). In this case, a natural method is to choose some state X_0, iteratively sample X_{t+1} from $P(X_t,\cdot)$ until the mixing time $t = t_{\text{mix}}(\varepsilon)$ (as defined in Chapter V) and estimate $\mathbb{E}_{X\sim\pi} f(X)$ as $f(X_{t_{\text{mix}}(\varepsilon)})$ (perhaps averaging over many samples). However, when P is unknown, $t_{\text{mix}}(\varepsilon)$ is unknown as well, so a recent line of work has proposed algorithms for estimating $t_{\text{mix}}(\varepsilon)$ in this setting [120, 121, 122, 123]. These existing works indirectly estimate $t_{\text{mix}}(\varepsilon)$ by instead estimating the *relaxation time* t_{rel} (see Appendix D.1). A classical result in the mixing times literature states $(t_{\text{rel}} - 1)\log(1/2\varepsilon) \le t_{\text{mix}}(\varepsilon) \le t_{\text{rel}}\log(1/\varepsilon\pi_{\min})$, where $\pi_{\min}$ is the minimum stationary distribution of any state, and thus the upper bound is never tight in the number of states. Hence, a remaining challenge is *direct* estimation of $t_{\text{mix}}(\varepsilon)$ in the aforementioned sampling model. Our understanding of mixing times and PPR-like perturbations from Chapter V, and our extension of PPR estimators to this sampling model in Chapter IV, may be useful tools in tackling this challenge.

7.2.6 Social learning and fake news

Our main finding in Chapter VI was that in order to prevent social learning, adversaries should target agents who are influential, yet susceptible to influence themselves. An obvious

follow-up question is whether this finding holds in practice; namely, whether influential-yet-susceptible users encountering fake news facilitates its spread. This question could be addressed in several ways. From a social sciences perspective, one could investigate whether our social learning model is a reasonable approximation of how social media users' opinions evolve. From a data science perspective, one could use historical data to investigate if fake news spreads more extensively when it reaches influential-yet-susceptible users (quantified in some manner). If our finding indeed holds in practice, the next question would be how to leverage it to mitigate the spread of fake news. And if such solutions exist, incentivizing social media platforms to adopt them would be yet another issue. In short, Chapter VI suggests a more sprawling research agenda than the other directions proposed in this chapter. Nevertheless, we believe these are important questions given the societal impact of fake news.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[2] T. H. Haveliwala, "Topic-sensitive pagerank," in *Proceedings of the 11th international conference on World Wide Web*. ACM, 2002, pp. 517–526.

[3] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh, "WTF: The who to follow service at twitter," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 505–514.

[4] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly, "Video suggestion and discovery for YouTube: Taking random walks through the view graph," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 895–904.

[5] V. Freschi, "Protein function prediction from interaction networks using a random walk ranking algorithm," in *Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering*. IEEE, 2007, pp. 42–48.

[6] J. L. Morrison, R. Breitling, D. J. Higham, and D. R. Gilbert, "Generank: Using search engine technology for the analysis of microarray experiments," *BMC bioinformatics*, vol. 6, no. 1, p. 233, 2005.

[7] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using PageRank vectors," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, 2006, pp. 475–486.

[8] ——, "Local partitioning for directed graphs using PageRank," in *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 2007, pp. 166–178.

[9] I. M. Kloumann, J. Ugander, and J. Kleinberg, "Block models and personalized pagerank," *Proceedings of the National Academy of Sciences*, vol. 114, no. 1, pp. 33–38, 2017.

[10] D. Koutra, J. T. Vogelstein, and C. Faloutsos, "Deltacon: A principled massive-graph similarity function," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 162–170.

[11] D. F. Gleich, "PageRank beyond the web," *SIAM Review*, vol. 57, no. 3, pp. 321–363, 2015.

[12] K. B. Athreya and O. Stenflo, "Perfect sampling for doeblin chains," Cornell University Operations Research and Industrial Engineering, Tech. Rep., 2000.

[13] J. G. Propp and D. B. Wilson, "Exact sampling with coupled markov chains and applications to statistical mechanics," *Random Structures & Algorithms*, vol. 9, no. 1-2, pp. 223–252, 1996.

[14] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova, "Monte carlo methods in PageRank computation: When one iteration is sufficient," *SIAM Journal on Numerical Analysis*, vol. 45, no. 2, pp. 890–904, 2007.

[15] P. Lofgren, S. Banerjee, and A. Goel, "Personalized PageRank estimation and search: A bidirectional approach," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 2016, pp. 163–172.

[16] G. Jeh and J. Widom, "Scaling personalized web search," in *Proceedings of the 12th international conference on World Wide Web*. ACM, 2003, pp. 271–279.

[17] P. Berkhin, "Bookmark-coloring algorithm for personalized pagerank computing," *Internet Mathematics*, vol. 3, no. 1, pp. 41–62, 2006.

[18] N. Chen and M. Olvera-Cravioto, "Directed random graphs with given degree distributions," *Stochastic Systems*, vol. 3, no. 1, pp. 147–186, 2013.

[19] W. B. Haskell, R. Jain, and D. Kalathil, "Empirical dynamic programming," *Mathematics of Operations Research*, vol. 41, no. 2, pp. 402–429, 2016.

[20] L. Avena, H. Güldaş, R. van der Hofstad, and F. den Hollander, "Random walks on dynamic configuration models: a trichotomy," *Stochastic Processes and their Applications*, 2018.

[21] P. Caputo and M. Quattropani, "Mixing time of pagerank surfers on sparse random digraphs," *arXiv preprint arXiv:1905.04993*, 2019.

[22] R. Basu, J. Hermon, and Y. Peres, "Characterization of cutoff for reversible markov chains," in *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2015, pp. 1774–1791.

[23] M. Azzimonti and M. Fernandes, "Social media networks, fake news, and polarization," National Bureau of Economic Research, Tech. Rep., 2018.

[24] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, "Non-bayesian social learning," *Games and Economic Behavior*, vol. 76, no. 1, pp. 210–225, 2012.

[25] D. Acemoglu, A. Ozdaglar, and A. ParandehGheibi, "Spread of (mis) information in social networks," *Games and Economic Behavior*, vol. 70, no. 2, pp. 194–227, 2010.

[26] B. Golub and M. O. Jackson, "Naive learning in social networks and the wisdom of crowds," *American Economic Journal: Microeconomics*, vol. 2, no. 1, pp. 112–49, 2010.

[27] D. Vial and V. Subramanian, "On the role of clustering in personalized pagerank estimation," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, vol. 4, no. 4, p. 21, 2019.

[28] ——, "Towards fast algorithms for estimating personalized pagerank using commonly generated random walks," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 852–855.

[29] H. Tong, C. Faloutsos, and J.-Y. Pan, "Random walk with restart: Fast solutions and applications," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 327–346, 2008.

[30] K. Shin, J. Jung, S. Lee, and U. Kang, "Bear: Block elimination approach for random walk with restart on large graphs," in *Proc. of the 2015 ACM SIGMOD International Conf. on Management of Data*. ACM, 2015, pp. 1571–1585.

[31] D. Fogaras, B. Rácz, K. Csalogány, and T. Sarlós, "Towards scaling fully personalized PageRank: Algorithms, lower bounds, and experiments," *Internet Mathematics*, vol. 2, no. 3, pp. 333–358, 2005.

[32] R. Andersen, C. Borgs, J. Chayes, J. Hopcroft, V. Mirrokni, and S.-H. Teng, "Local computation of PageRank contributions," *Internet Mathematics*, vol. 5, no. 1-2, pp. 23–45, 2008.

[33] C. E. Lee, A. Ozdaglar, and D. Shah, "Computing the stationary distribution locally," in *Advances in Neural Information Processing Systems*, 2013, pp. 1376–1384.

[34] ——, "Asynchronous approximation of a single component of the solution to a linear system," *arXiv preprint arXiv:1411.2647*, 2014.

[35] C. E. Lee, A. E. Ozdaglar, and D. Shah, "Solving systems of linear equations: Locally and asynchronously," *CoRR*, vol. abs/1411.2647, 2014.

[36] C. Borgs, M. Brautbar, J. Chayes, and S.-H. Teng, "Multiscale matrix sampling and sublinear-time pagerank computation," *Internet Mathematics*, vol. 10, no. 1-2, pp. 20–48, 2014.

[37] H. Ishii and R. Tempo, "Distributed randomized algorithms for the pagerank computation," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 1987–2002, 2010.

[38] J. Lei and H.-F. Chen, "Distributed randomized pagerank algorithm based on stochastic approximation," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1641–1646, 2014.

[39] V. S. Borkar and A. S. Mathkar, "Reinforcement learning for matrix computations: Pagerank as an example," in *International Conference on Distributed Computing and Internet Technology*. Springer, 2014, pp. 14–24.

[40] L. Dai and N. M. Freris, "Fully distributed pagerank computation with exponential convergence," *arXiv preprint arXiv:1705.09927*, 2017.

[41] A. D. Sarma, A. R. Molla, G. Pandurangan, and E. Upfal, "Fast distributed pagerank computation," in *International Conference on Distributed Computing and Networking*. Springer, 2013, pp. 11–26.

[42] J. A. Tropp, "An introduction to matrix concentration inequalities," *Foundations and Trends® in Machine Learning*, vol. 8, no. 1-2, pp. 1–230, 2015.

[43] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data.

[44] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[45] D. Vial and V. Subramanian, "A structural result for personalized pagerank and its algorithmic consequences," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 2, p. 25, 2019.

[46] P. Boldi, M. Santini, and S. Vigna, "Pagerank as a function of the damping factor," in *Proceedings of the 14th international conference on World Wide Web*. ACM, 2005, pp. 557–566.

[47] ——, "Pagerank: functional dependencies," *ACM Transactions on Information Systems (TOIS)*, vol. 27, no. 4, p. 19, 2009.

[48] N. Chen, N. Litvak, and M. Olvera-Cravioto, "Generalized pagerank on directed configuration networks," *Random Structures & Algorithms*, vol. 51, no. 2, pp. 237–274, 2017.

[49] D. J. Aldous and A. Bandyopadhyay, "A survey of max-type recursive distributional equations," *The Annals of Applied Probability*, vol. 15, no. 2, pp. 1047–1110, 2005.

[50] J. Lee and M. Olvera-Cravioto, "PageRank on inhomogeneous random digraphs," *arXiv preprint arXiv:1707.02492*, 2017.

[51] N. Chen, N. Litvak, and M. Olvera-Cravioto, "PageRank in scale-free random graphs," in *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 2014, pp. 120–131.

[52] N. Litvak, W. R. Scheinhardt, and Y. Volkovich, "In-degree and PageRank: Why do they follow similar power laws?" *Internet Mathematics*, vol. 4, no. 2-3, pp. 175–198, 2007.

[53] Y. Volkovich and N. Litvak, "Asymptotic analysis for personalized web search," *Advances in Applied Probability*, vol. 42, no. 2, pp. 577–604, 2010.

[54] Y. Volkovich, N. Litvak, and D. Donato, "Determining factors behind the PageRank log-log plot," in *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 2007, pp. 108–123.

[55] K. Avrachenkov, A. Kadavankandy, L. O. Prokhorenkova, and A. Raigorodskii, "Pagerank in undirected random graphs," in *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 2015, pp. 151–163.

[56] C. Bordenave, P. Caputo, and J. Salez, "Random walk on sparse random digraphs," *Probability Theory and Related Fields*, vol. 170, no. 3-4, pp. 933–960, 2018.

[57] E. A. Bender and E. R. Canfield, "The asymptotic number of labeled graphs with given degree sequences," *Journal of Combinatorial Theory, Series A*, vol. 24, no. 3, pp. 296–307, 1978.

[58] B. Bollobás, "A probabilistic proof of an asymptotic formula for the number of labelled regular graphs," *European Journal of Combinatorics*, vol. 1, no. 4, pp. 311–316, 1980.

[59] N. C. Wormald, "Some problems in the enumeration of labelled graphs," *Bulletin of the Australian Mathematical Society*, vol. 21, no. 1, pp. 159–160, 1980.

[60] R. van der Hofstad, G. Hooghiemstra, and P. Van Mieghem, "Distances in random graphs with finite variance degrees," *Random Structures & Algorithms*, vol. 27, no. 1, pp. 76–123, 2005.

[61] M. Molloy and B. Reed, "A critical point for random graphs with a given degree sequence," *Random structures & algorithms*, vol. 6, no. 2-3, pp. 161–180, 1995.

[62] P. Lofgren and A. Goel, "Personalized PageRank to a target node," *arXiv preprint arXiv:1304.4658*, 2013.

[63] T. Sarlós, A. A. Benczúr, K. Csalogány, D. Fogaras, and B. Rácz, "To randomize or not to randomize: space optimal summaries for hyperlink analysis," in *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006, pp. 297–306.

[64] D. A. Spielman and S.-H. Teng, "Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems," in *Proceedings of the STOC*, vol. 4, 2004.

[65] M. B. Cohen, R. Kyng, G. L. Miller, J. W. Pachocki, R. Peng, A. B. Rao, and S. C. Xu, "Solving sdd linear systems in nearly m log 1/2 n time," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. ACM, 2014, pp. 343–352.

[66] I. Koutis, G. L. Miller, and R. Peng, "Approaching optimality for solving sdd linear systems," *SIAM Journal on Computing*, vol. 43, no. 1, pp. 337–354, 2014.

[67] M. B. Cohen, J. Kelner, R. Kyng, J. Peebles, R. Peng, A. B. Rao, and A. Sidford, "Solving directed laplacian systems in nearly-linear time through sparse lu factorizations," in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2018, pp. 898–909.

[68] M. B. Cohen, J. Kelner, J. Peebles, R. Peng, A. Sidford, and A. Vladu, "Faster algorithms for computing the stationary distribution, simulating random walks, and more," in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2016, pp. 583–592.

[69] M. B. Cohen, J. Kelner, J. Peebles, R. Peng, A. B. Rao, A. Sidford, and A. Vladu, "Almost-linear-time algorithms for markov chains and new spectral primitives for directed graphs," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2017, pp. 410–419.

[70] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[71] R. Sen, K. Shanmugam, M. Kocaoglu, A. G. Dimakis, and S. Shakkottai, "Contextual bandits with latent confounders: An nmf approach," *arXiv preprint arXiv:1606.00119*, 2016.

[72] I. Markovsky, *Low rank approximation: algorithms, implementation, applications*. Springer Science & Business Media, 2011.

[73] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 1024–1034.

[74] B. Amento, L. Terveen, and W. Hill, "Does "authority" mean quality? predicting expert quality ratings of web documents," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2000, pp. 296–303.

[75] S. Fortunato, M. Boguñá, A. Flammini, and F. Menczer, "Approximating pagerank from in-degree," in *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 2006, pp. 59–71.

[76] C. Bordenave, "Lecture notes on random graphs and probabilistic combinatorial optimization," https://www.math.univ-toulouse.fr/~bordenave/coursRG.pdf.

[77] P. Boldi, M. Santini, and S. Vigna, "Laboratory for Web Algorithmics datasets," http://law.di.unimi.it/datasets.php.

[78] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[79] A. Clauset, C. R. Shalizi, and M. E. Newman, "Power-law distributions in empirical data," *SIAM Review*, vol. 51, no. 4, pp. 661–703, 2009.

[80] P. Boldi and S. Vigna, "The WebGraph framework I: Compression techniques," in *Proceedings of the 13th International Conference on World Wide Web*. ACM, 2004, pp. 595–602.

[81] C. Bordenave, P. Caputo, and J. Salez, "Cutoff at the "entropic time" for sparse markov chains," *Probability Theory and Related Fields*, pp. 1–32, 2016.

[82] M. Charikar and A. Sahai, "Dimension reduction in the l_1 norm," in *2002 IEEE Symposium on Foundations of Computer Science*. IEEE, 2002, pp. 551–560.

[83] P. Indyk, "Stable distributions, pseudorandom generators, embeddings and data stream computation," in *2000 IEEE Symposium on Foundations of Computer Science*. IEEE, 2000, pp. 189–197.

[84] K. Avrachenkov and D. Lebedev, "Pagerank of scale-free growing networks," *Internet Mathematics*, vol. 3, no. 2, pp. 207–231, 2006.

[85] A. Garavaglia, R. van der Hofstad, and N. Litvak, "Local weak convergence for pagerank," *arXiv preprint arXiv:1803.06146*, 2018.

[86] P. R. Kumar and P. Varaiya, *Stochastic systems: Estimation, identification, and adaptive control*. SIAM, 2015, vol. 75.

[87] A. Gupta, R. Jain, and P. W. Glynn, "An empirical algorithm for relative value iteration for average-cost MDPs," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 5079–5084.

[88] W. B. Haskell, R. Jain, and D. Kalathil, "Empirical value iteration for approximate dynamic programming," in *2014 American Control Conference*. IEEE, 2014, pp. 495–500.

[89] H. Sharma and R. Jain, "An approximately optimal relative value learning algorithm for averaged MDPs with continuous states and actions," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2019, pp. 734–740.

[90] H. Sharma, R. Jain, and A. Gupta, "An empirical relative value learning algorithm for non-parametric MDPs with continuous state space," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 1368–1373.

[91] C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel, "Reverse curriculum generation for reinforcement learning," *arXiv preprint arXiv:1707.05300*, 2017.

[92] A. D. Edwards, L. Downs, and J. C. Davidson, "Forward-backward reinforcement learning," *arXiv preprint arXiv:1803.10227*, 2018.

[93] A. Goyal, P. Brakel, W. Fedus, S. Singhal, T. Lillicrap, S. Levine, H. Larochelle, and Y. Bengio, "Recall traces: Backtracking models for efficient reinforcement learning," *arXiv preprint arXiv:1804.00379*, 2018.

[94] N. R. Ke, A. G. A. P. GOYAL, O. Bilaniuk, J. Binas, M. C. Mozer, C. Pal, and Y. Bengio, "Sparse attentive backtracking: Temporal credit assignment through reminding," in *Advances in Neural Information Processing Systems*, 2018, pp. 7640–7651.

[95] S. Banerjee and P. Lofgren, "Fast bidirectional probability estimation in markov models," in *Advances in Neural Information Processing Systems*, 2015, pp. 1423–1431.

[96] Twitter developer documentation, "Rate limiting," https://developer.twitter.com/en/docs/basics/rate-limiting, accessed: 12 December 2019.

[97] D. Vial and V. Subramanian, "Restart perturbations for lazy, reversible markov chains: trichotomy and pre-cutoff equivalence," *arXiv preprint arXiv:1907.02926*, 2019.

[98] P. Caputo and M. Quattropani, "Mixing time trichotomy in regenerating dynamic digraphs," *arXiv preprint arXiv:1911.07025*, 2019.

[99] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*. American Mathematical Society, 2009.

[100] A. Y. Mitrophanov, "Stability and exponential convergence of continuous-time markov chains," *Journal of applied probability*, vol. 40, no. 4, pp. 970–979, 2003.

[101] ——, "Sensitivity and convergence of uniformly ergodic markov chains," *Journal of Applied Probability*, vol. 42, no. 4, pp. 1003–1014, 2005.

[102] D. Vial and V. Subramanian, "Local non-bayesian social learning with stubborn agents," *arXiv preprint arXiv:1904.12767*, 2019.

[103] ——, "Local non-bayesian social learning with stubborn agents," in *Proceedings of the 2019 ACM Conference on Economics and Computation*. ACM, 2019, pp. 551–552.

[104] E. Shearer and J. Gottfried, "News use across social media platforms 2017," *Pew Research Center, Journalism and Media*, 2017.

[105] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–36, 2017.

[106] C. Shao, G. L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," *Nature communications*, vol. 9, no. 1, p. 4787, 2018.

[107] K. Murota, *Discrete convex analysis*. SIAM, 2003.

[108] A. J. Walker, "New fast method for generating discrete random numbers with arbitrary frequency distributions," *Electronics Letters*, vol. 10, no. 8, pp. 127–128, 1974.

[109] ——, "An efficient method for generating discrete random variables with general distributions," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 253–256, 1977.

[110] D. E. Knuth, *Art of computer programming, volume 2: Seminumerical algorithms*. Addison-Wesley Professional, 2014.

[111] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[112] B. Golub and E. Sadler, "Learning in social networks," 2017.

[113] D. Acemoglu and A. Ozdaglar, "Opinion dynamics and learning in social networks," *Dynamic Games and Applications*, vol. 1, no. 1, pp. 3–49, 2011.

[114] M. A. Rahimian, S. Shahrampour, and A. Jadbabaie, "Learning without recall by random walks on directed graphs," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 5538–5543.

[115] D. Acemoglu, G. Como, F. Fagnani, and A. Ozdaglar, "Opinion fluctuations and persistent disagreement in social networks," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 2347–2352.

[116] J. Ghaderi and R. Srikant, "Opinion dynamics in social networks with stubborn agents: Equilibrium and convergence rate," *Automatica*, vol. 50, no. 12, pp. 3209–3215, 2014.

[117] T. Rocket, "Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies," 2018.

[118] P. Diaconis, "Group representations in probability and statistics," *Lecture notes-monograph series*, vol. 11, pp. i–192, 1988.

[119] N. Alon, I. Benjamini, E. Lubetzky, and S. Sodin, "Non-backtracking random walks mix faster," *Communications in Contemporary Mathematics*, vol. 9, no. 04, pp. 585–603, 2007.

[120] R. Combes and M. Touati, "Computationally efficient estimation of the spectral gap of a markov chain," *arXiv preprint arXiv:1806.06047*, 2018.

[121] D. J. Hsu, A. Kontorovich, and C. Szepesvári, "Mixing time estimation in reversible markov chains from a single sample path," in *Advances in neural information processing systems*, 2015, pp. 1459–1467.

[122] D. A. Levin and Y. Peres, "Estimating the spectral gap of a reversible markov chain from a short trajectory," *arXiv preprint arXiv:1612.05330*, 2016.

[123] G. Wolfer and A. Kontorovich, "Estimating the mixing time of ergodic markov chains," *arXiv preprint arXiv:1902.01224*, 2019.

[124] D. Dubhashi and A. Panconesi, *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[125] A. J. Laub, *Matrix analysis for scientists and engineers*. Siam, 2005.

[126] R. G. Gallager, *Stochastic processes: theory for applications*. Cambridge University Press, 2013.

[127] Z. Zhu, Z. Yang, and E. Oja, "Multiplicative updates for learning with stochastic matrices," in *Scandinavian Conference on Image Analysis*. Springer, 2013, pp. 143–152.

[128] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[129] S. Boucheron, G. Lugosi, and P. Massart, *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

APPENDICES

APPENDIX A

Proofs and Experimental Details for Chapter II

A.1 Analysis of FW-BW-MCMC and comparison to Bidirectional-PPR

Here we state and prove the guarantees that were stated informally in Section 2.3. We include the corresponding results for Bidirectional-PPR for comparison. We first state the accuracy guarantee, Theorem A.1. The idea is to bound relative error when $\pi_s(t) \geq \delta$ and to bound absolute error when $\pi_s(t) < \delta$. The authors of [15] suggest choosing $\delta = O(\frac{1}{n})$. This choice dictates that we desire the relative bound when t's PPR exceeds a uniform distribution over all nodes, which suggests that t is "significant" to s in this case. The proof applies the Chernoff bound to a variety of cases, which vary between the algorithms.

Theorem A.1. Fix minimum PPR threshold δ, relative error tolerance ε, and failure probability p_{fail}. For FW-BW-MCMC, assume the following hold:

$$\varepsilon \in \left(0, \frac{1}{\sqrt{2e}}\right), \quad w = \frac{c r^s_{\max} r^t_{\max}}{\delta}, \quad c > \frac{3(2e)^{1/3}\log(2/p_{\text{fail}})}{\varepsilon^{7/3}}. \tag{A.1}$$

For Bidirectional-PPR, assume the following hold:

$$r^t_{\max} \in \left(\frac{2e\delta}{\alpha\varepsilon}, 1\right), \quad w = \frac{c r^t_{\max}}{\delta}, \quad c > \frac{3\log(2/p_{\text{fail}})}{\varepsilon^2}. \tag{A.2}$$

Then with probability $\geq 1 - p_{\text{fail}}$, the estimate $\hat{\pi}_s(t)$ from either algorithm satisfies

$$|\pi_s(t) - \hat{\pi}_s(t)| \leq \begin{cases} \varepsilon\pi_s(t), & \pi_s(t) \geq \delta \text{ (significant case)} \\ 2e\delta, & \pi_s(t) < \delta \text{ (insignificant case)} \end{cases}. \tag{A.3}$$

Proof. See [15] for Bidirectional-PPR; see Appendix A.1.1 for FW-BW-MCMC. □

From Theorem A.1, FW-BW-MCMC offers the same accuracy as Bidirectional-PPR. However, our assumptions on ε and c are stronger than those required for Bidirectional-PPR. The first assumption is mild, since $\frac{1}{\sqrt{2e}} \approx 0.43$ and we typically desire a tighter relative error bound. The second affects complexity and will be discussed next. Note also that our guarantee holds $\forall\ r^t_{\max} \in (0, 1)$, while Bidirectional-PPR requires a lower bound on $r^t_{\max}$.

Next, we have a worst-case complexity result in Theorem A.2 (by worst case, we for any $s, t \in V$). The idea is to choose $r^s_{\max}, r^t_{\max}$ to balance the complexity of the DP and MCMC stages of the algorithm. The result requires the additional assumption $m\delta < \log(1/p_{\text{fail}})/\varepsilon^2$, which guarantees that these $r^s_{\max}, r^t_{\max}$ values lie in $(0, 1)$. Note that with $\delta = O(\frac{1}{n})$, this implies $m = O(n)$, i.e. nodes have constant degrees as n grows.

Theorem A.2. Fix minimum PPR threshold δ, relative error tolerance ε, and failure probability p_{fail}. Assume (A.1)-(A.2) hold and $m\delta < \log(1/p_{\text{fail}})/\varepsilon^2$. Then setting $r^s_{\max} = r^t_{\max} = \frac{m^{1/3}\delta^{1/3}\varepsilon^{7/9}}{(\log(1/p_{\text{fail}}))^{1/3}}$ in `FW-BW-MCMC` yields complexity $O\left(\frac{m^{2/3}(\log(1/p_{\text{fail}}))^{1/3}}{\alpha\varepsilon^{7/9}\delta^{1/3}}\right)$, and setting $r^t_{\max} = \frac{\sqrt{m\delta}\varepsilon}{\sqrt{\log(1/p_{\text{fail}})}}$ in `Bidirectional-PPR` yields complexity $O\left(\frac{\sqrt{m\log(1/p_{\text{fail}})}}{\alpha\varepsilon\sqrt{\delta}}\right)$.

Proof. See Appendix A.1.2. □

Note that, with $\delta = O(\frac{1}{n})$, so that $m = O(n)$, both algorithms have complexity linear in n, while `FW-BW-MCMC` has strictly better dependence on the parameters p_{fail} and ε.

Finally, we present an average-case complexity result for `FW-BW-MCMC-Practical` (Algorithm A.2), which uses termination criteria $\|D^{-1}r^s\|_\infty \le r^s_{\max}$ in the forward DP.

Theorem A.3. For any $s \in V$ and $t \sim V$ uniformly, `FW-BW-MCMC-Practical` produces an estimate satisfying the Theorem A.1 and has complexity $O\left(\frac{\sqrt{m\log(1/p_{\text{fail}})}}{\sqrt{n\delta}\alpha\varepsilon^{7/6}}\right)$.

Proof. See Appendix A.2. □

With $\delta = O(\frac{1}{n})$, the average-case complexity is $O(\sqrt{m})$, as claimed in Section 2.3. The guarantee for `Bidirectional-PPR` in [15] has ε instead of $\varepsilon^{7/6}$ but is otherwise identical.

A.1.1 Proof of Theorem A.1

We will use the following result from [124].

Theorem A.4. (from Theorem 1.1 in [124]) Let $\{Z_i\}$ be a set of independent random variables with $Z_i \in [0, 1]\ \forall\ i$, and let $Z = \sum_i Z_i$. Then for any $\eta \in (0, 1)$ and any $d > 2e\mathbb{E}[Z]$,

$$\mathbb{P}[Z > (1+\eta)\mathbb{E}[Z]] \le \exp(-\eta^2\mathbb{E}[Z]/3), \quad \mathbb{P}[Z < (1-\eta)\mathbb{E}[Z]] \le \exp(-\eta^2\mathbb{E}[Z]/2), \tag{A.4}$$

$$\mathbb{P}[Z > d] \le 2^{-d}. \tag{A.5}$$

To begin the proof, we define $Y_i = X_i/r^t_{\max}$ and $Y = \sum_{i=1}^w Y_i$, where X_i is from Algorithm 2.3. Observe the Y_i's are independent and $Y_i \in [0, 1]$ (by the terminating condition of Algorithm 2.2), so Theorem A.4 applies for appropriate choices of η and d. We also observe that (A.6) holds, which follows by linearity and $w = \frac{cr^s_{\max}r^t_{\max}}{\delta}$ in the statement of the theorem.

$$\mathbb{E}[Y] = \frac{w}{r^t_{\max}}\mathbb{E}[X_i] = \frac{cr^s_{\max}}{\delta}\mathbb{E}[X_i]. \tag{A.6}$$

We now turn to the case $\pi_s(t) \ge \delta$, for which we aim to show $\mathbb{P}[|\hat{\pi}_s(t) - \pi_s(t)| > \varepsilon\pi_s(t)] < p_{\text{fail}}\ \forall\ \varepsilon \in (0, \frac{1}{\sqrt{2e}})$. We will examine three sub-cases. The first two sub-cases depend on the

constant $k := (\frac{\varepsilon}{2e})^{1/3}$ (we motivate the choice of this constant at the conclusion of the proof). We also observe the following, which follows from the assumption $c > \frac{3(2e)^{1/3}\log(2/p_{\text{fail}})}{\varepsilon^{7/3}}$:

$$\frac{k}{3} = \frac{\varepsilon}{6ek^2} = \frac{\varepsilon^{1/3}}{3(2e)^{1/3}} > \frac{\log(2/p_{\text{fail}})}{\varepsilon^2 c}. \tag{A.7}$$

For the first sub-case, assume $\mathbb{E}[Y] \geq kc$. Then we have the following:

$$\begin{aligned}\mathbb{P}[|\hat{\pi}_s(t) - \pi_s(t)| > \varepsilon\pi_s(t)] &\leq \mathbb{P}\left[\left|\frac{1}{w}\sum_{i=1}^{w} X_i - \mathbb{E}[X_i]\right| > \varepsilon\mathbb{E}[X_i]\right] \\ &= \mathbb{P}\left[|Y - \mathbb{E}[Y]| > \varepsilon\mathbb{E}[Y]\right] \leq 2\exp\left(-\varepsilon^2\mathbb{E}[Y]/3\right) \leq 2\exp\left(-\varepsilon^2 kc/3\right) < p_{\text{fail}}.\end{aligned}$$

Here the first inequality holds by definition of $\hat{\pi}_s(t)$ in Algorithm 2.3 and the invariant (2.3); the equality holds by (A.6) and the definition of Y; the second inequality uses Theorem A.4 (note $\varepsilon < \frac{1}{\sqrt{2e}} < 1$); and the final two inequalities hold by $\mathbb{E}[Y] \geq kc$ and (A.7).

For the second sub-case, assume $\mathbb{E}[Y] \in [\frac{\varepsilon c}{2e}, kc)$. First, observe that by (A.6), the assumption $\mathbb{E}[Y] < kc$, and the Algorithm 2.1 terminating condition,

$$\|r^s\|_1\mathbb{E}[X_i] = \frac{\|r^s\|_1\delta\mathbb{E}[Y]}{cr^s_{\max}} < \frac{\|r^s\|_1 k\delta}{r^s_{\max}} \leq k\delta.$$

and so $\pi_s(t) \geq \|r^s\|_1\mathbb{E}[X_i] + (1-k)\delta$ (else, $\pi_s(t) < \delta$ by (2.3), a contradiction). Then:

$$\begin{aligned}\mathbb{P}[|\hat{\pi}_s(t) - \pi_s(t)| > \varepsilon\pi_s(t)] &\leq \mathbb{P}\left[\left|\frac{1}{w}\sum_{i=1}^{w} X_i - \mathbb{E}[X_i]\right| > \varepsilon\left(\mathbb{E}[X] + \frac{(1-k)\delta}{\|r^s\|_1}\right)\right] \\ &= \mathbb{P}\left[|Y - \mathbb{E}[Y]| > \varepsilon\left(\mathbb{E}[Y] + \frac{(1-k)\delta w}{\|r^s\|_1 r^t_{\max}}\right)\right] \leq \mathbb{P}\left[|Y - \mathbb{E}[Y]| > \varepsilon\left(\mathbb{E}[Y] + (1-k)c\right)\right] \\ &= \mathbb{P}\left[|Y - \mathbb{E}[Y]| > \varepsilon\left(\mathbb{E}[Y] + \left(\frac{1-k}{k}\right)kc\right)\right] < \mathbb{P}\left[|Y - \mathbb{E}[Y]| > \frac{\varepsilon}{k}\mathbb{E}[Y]\right] \\ &\leq 2\exp\left(-\varepsilon^2\mathbb{E}[Y]/\left(3k^2\right)\right) < 2\exp\left(-\varepsilon^3 c/\left(6ek^2\right)\right) < p_{\text{fail}}.\end{aligned}$$

Here the first inequality and first equality follow similar arguments as Case 1; the second inequality is by the Algorithm 2.1 terminating condition and $w = \frac{cr^s_{\max}r^t_{\max}}{\delta}$; the second equality multiplies and divides k; the third inequality holds by assumption $\mathbb{E}[Y] \in [\frac{\varepsilon c}{2e}, kc)$; the fourth inequality holds by Theorem A.4 (note $\frac{\varepsilon}{k} = \varepsilon^{2/3}(2e)^{1/3} < 1$ by assumption $\varepsilon < \frac{1}{\sqrt{2e}}$); the fifth inequality follows from $\mathbb{E}[Y] \in [\frac{\varepsilon c}{2e}, kc)$; and the final inequality holds by (A.7). Note we have assumed $1 - k > 0$ in the third and fifth inequality; this follows from $\varepsilon < \frac{1}{\sqrt{2e}}$.

For the third and final sub-case, assume $\mathbb{E}[Y] < \frac{\varepsilon c}{2e}$. We have the following:

$$\begin{aligned}&\mathbb{P}[|\hat{\pi}_s(t) - \pi_s(t)| > \varepsilon\pi_s(t)] \\ &\quad= \mathbb{P}\left[\left|\frac{1}{w}\sum_{i=1}^{w} X_i - \mathbb{E}[X_i]\right| > \frac{\varepsilon\pi_s(t)}{\|r^s\|_1}\right] = \mathbb{P}\left[|Y - \mathbb{E}[Y]| > \frac{\varepsilon\pi_s(t)w}{\|r^s\|_1 r^t_{\max}}\right]\end{aligned}$$

$$\leq \mathbb{P}\left[|Y - \mathbb{E}[Y]| > \frac{\varepsilon\delta w}{r^s_{\max} r^t_{\max}}\right] = \mathbb{P}\left[|Y - \mathbb{E}[Y]| > \varepsilon c\right] \leq \mathbb{P}[Y > \varepsilon c] \leq 2^{-\varepsilon c}. \quad \text{(A.8)}$$

Here the first three equalities and first inequality follow similar arguments as previous cases; the penultimate inequality holds since $\{|Y - \mathbb{E}[Y]| > \varepsilon c\} \subset \{Y > \varepsilon c\}$ when $Y \geq \mathbb{E}[Y]$, whereas $\{|Y - \mathbb{E}[Y]| > \varepsilon c\} \subset \{\mathbb{E}[Y] > \varepsilon c\} \subset \{2e\mathbb{E}[Y] > \varepsilon c\} = \emptyset$ when $Y < \mathbb{E}[Y]$; and the final inequality holds by Theorem A.4; note $\varepsilon c > 2e\mathbb{E}[Y]$ by assumption. Next, observe

$$\varepsilon c > \frac{6e\log(1/p_{\text{fail}})}{(\sqrt{2e}\varepsilon)^{4/3}} > 6e\log(1/p_{\text{fail}}) = \frac{6e}{\log_2(e)}\log_2(1/p_{\text{fail}}) > \log_2(1/p_{\text{fail}}). \quad \text{(A.9)}$$

where the first two inequalities hold by $c > \frac{3(2e)^{1/3}\log(1/p_{\text{fail}})}{\varepsilon^{7/3}}$ and $\varepsilon < \frac{1}{\sqrt{2e}}$, and the final one holds since $\log_2(e) < 2 \Rightarrow \frac{6e}{\log_2(e)} > \frac{3e}{2} > 1$. Combining (A.8) and (A.9) completes Case 3.

Finally, note the bounds in Cases 1 and 3 grow with decreasing and increasing k, respectively. Hence, $k = (\frac{\varepsilon}{2e})^{1/3}$ arises from equating the two to minimize failure probability.

We now turn to the case $\pi_s(t) < \delta$. Observe that by $\pi_s(t) < \delta$ and the invariant (2.3), $\|r^s\|_1\mathbb{E}[X_i] < \delta$. By (A.6), this implies $2e\mathbb{E}[Y] < \frac{2ew\delta}{r^t_{\max}\|r^s\|_1} =: b$. Then

$$\begin{aligned}\mathbb{P}[|\hat{\pi}_s(t) - \pi_s(t)| > 2e\delta] &= \mathbb{P}\left[\left|\frac{1}{w}\sum_{i=1}^{w} X_i - \mathbb{E}[X_i]\right| > \frac{2e\delta}{\|r^s\|_1}\right] = \mathbb{P}\left[|Y - \mathbb{E}[Y]| > \frac{2e\delta w}{\|r^s\|_1 r^t_{\max}}\right] \\ &= \mathbb{P}\left[|Y - \mathbb{E}[Y]| > b\right] \leq P[Y > b] \leq 2^{-b}. \quad \text{(A.10)}\end{aligned}$$

Here the equalities follow similar steps as previous cases, the first inequality holds by the same argument in the Case 3 analysis, and the final inequality holds by Theorem A.4 (note (A.5) applies since $b > 2e\mathbb{E}[Y]$). We also observe

$$b = \frac{2ew\delta}{r^t_{\max}\|r^s\|_1} > \frac{2ew\delta}{r^t_{\max} r^s_{\max}} = 2ec > \varepsilon c > \log_2(1/p_{\text{fail}}), \quad \text{(A.11)}$$

where the first inequality is by the Algorithm 2.1 terminating condition, the second inequality holds since $2e > 1 > \varepsilon$, and the third inequality follows from (A.9); the equalities are by definition. Finally, we combine (A.10) and (A.11) to complete the proof.

A.1.2 Proof of Theorem A.2

The complexity of Algorithm 2.3 is the total complexity of Algorithm 2.2, Algorithm 2.1, and the random walks. Below, we show Algorithms 2.2 and 2.1 have complexity $\frac{m}{\alpha r^t_{\max}}$ and $\frac{m}{\alpha r^s_{\max}}$, respectively (using arguments from [32] and [7]). Furthermore, the complexity of the random walk stage is $O(\frac{r^s_{\max} r^t_{\max}\log(1/p_{\text{fail}})}{\alpha\delta\varepsilon^{7/3}})$, where $\frac{1}{\alpha}$ is the expected complexity of sampling a single random walk, and where the remaining factors give the number of walks required for (A.1) to hold. Hence, the complexity of Algorithm 2.3 is $O(C(r^s_{\max} r^t_{\max})/\alpha)$, where

$$C(r^s_{\max} r^t_{\max}) = \frac{m}{r^s_{\max}} + \frac{r^s_{\max} r^t_{\max}\log(1/p_{\text{fail}})}{\delta\varepsilon^{7/3}} + \frac{m}{r^t_{\max}}. \quad \text{(A.12)}$$

We now aim choose $r^s_{\max}, r^t_{\max}$ to minimize $O(C(r^s_{\max} r^t_{\max})/\alpha)$, or equivalently, to minimize

$C(r^s_{\max} r^t_{\max})$. For this, we let $K = \frac{\log(1/p_{\text{fail}})}{\delta\varepsilon^{7/3}} > 0$ and note $\frac{\partial C}{\partial r^s_{\max}} = K r^t_{\max} - \frac{m}{(r^s_{\max})^2} = 0$ if and only if $(r^s_{\max})^2 r^t_{\max} = \frac{m}{K}$, and similarly, $\frac{\partial C}{\partial r^t_{\max}} = 0$ if and only if $(r^t_{\max})^2 r^s_{\max} = \frac{m}{K}$; hence, $(\frac{m}{K})^{1/3}, (\frac{m}{K})^{1/3}$ is a stationary point of $C(r^s_{\max}, r^t_{\max})$. To verify this is a minimizer, we observe

$$\begin{bmatrix} \frac{\partial^2 C}{\partial (r^s_{\max})^2} & \frac{\partial^2 C}{\partial r^s_{\max} r^t_{\max}} \\ \frac{\partial^2 C}{\partial r^t_{\max} r^s_{\max}} & \frac{\partial^2 C}{\partial (r^t_{\max})^2} \end{bmatrix} = \begin{bmatrix} \frac{2m}{(r^s_{\max})^3} & K \\ K & \frac{2m}{(r^t_{\max})^3} \end{bmatrix},$$

from which it follows that the Hessian of C evaluated at $r^s_{\max} = r^t_{\max} = (\frac{m}{K})^{1/3}$ is $K(I + 11^{\mathsf{T}})$. This is positive definite, since $z^{\mathsf{T}}(K(I+11^{\mathsf{T}}))z = K(\|z\|_2^2 + (z^{\mathsf{T}}1)^2) > 0$ for any nonzero vector z. To summarize, we have shown $r^s_{\max} = r^t_{\max} = (\frac{m}{K})^{1/3}$ minimizes $C(r^s_{\max}, r^t_{\max})$ and thus minimizes the complexity of Algorithm 2.3, i.e. the choice of $r^s_{\max}, r^t_{\max}$ in the statement of the theorem minimizes complexity. Finally, substituting $r^s_{\max} = r^t_{\max} = (\frac{m}{K})^{1/3}$ into (A.12) and dividing by α gives the complexity expression of the theorem. Following the same approach establishes the Algorithm `Bidirectional-PPR` complexity bound given in the theorem.

We return to bound the complexities of Algorithms 2.2 and 2.1. For Algorithm 2.2, we use an argument from [32]. First, let $v \in V$. From Algorithm 2.2, $p^t(v)$ increases by at least $\alpha r^t_{\max}$ at each iteration for which $v^* = v$. By the invariant (2.2), $p^t(v) \le \pi_v(t)$. Taken together, $v^* = v$ for at most $\frac{\pi_v(t)}{\alpha r^t_{\max}}$ iterations. Furthermore, the complexity of each iteration for which $v^* = v$ is $d_{\text{in}}(v)$. Hence, the complexity of all iterations for which $v^* = v$ is bounded by $d_{\text{in}}(v)\frac{\pi_v(t)}{\alpha r^t_{\max}}$. Finally, the complexity of Algorithm 2.2 can be bounded by summing over all $v \in V$, i.e. $\sum_{v\in V} d_{\text{in}}(v)\frac{\pi_v(t)}{\alpha r^t_{\max}} \le \frac{1}{\alpha r^t_{\max}} \sum_{v\in V} d_{\text{in}}(v) = \frac{m}{\alpha r^t_{\max}}$.

We turn to Algorithm 2.1. As mentioned in the main text, Algorithm 2.1 changes the termination criteria from [7]; for clarity, we include the original definition in Algorithm A.1. Here we use tilde marks to distinguish quantities from those in Algorithm 2.1, and we indicate iteration number k to improve clarity of the arguments to follow. Besides these notational changes, the only difference between Algorithms 2.1 and A.1 is the termination criteria.

With this notation in place, the complexity of Algorithm A.1 can be bounded as follows (using arguments from [7]). First, observe that for any iteration k,

$$\begin{aligned} \|\tilde{r}^s_k\|_1 &= \sum_{v\in V\setminus(\{v_k\}\cup N_{\text{out}}(v_k))} \tilde{r}^s_{k-1}(v) + \sum_{v\in N_{\text{out}}(v_k)} \left(\tilde{r}^s_{k-1}(v) + \frac{(1-\alpha)\tilde{r}^s_{k-1}(v_k)}{d_{\text{out}}(v_k)}\right) \\ &= \|\tilde{r}^s_{k-1}\|_1 - \alpha\tilde{r}^s_{k-1}(v_k), \end{aligned} \tag{A.13}$$

where the first equality holds via Algorithm A.1. Next, let k^* be the iteration at which Algorithm A.1 terminates. Then the complexity of the algorithm is $\sum_{k=1}^{k^*} d_{\text{out}}(v_k)$, and

$$\begin{aligned} \sum_{k=1}^{k^*} d_{\text{out}}(v_k) &= \sum_{k=1}^{k^*} \frac{d_{\text{out}}(v_k)}{\tilde{r}^s_{k-1}(v_k)}\tilde{r}^s_{k-1}(v_k) < \frac{1}{\tilde{r}^s_{\max}} \sum_{k=1}^{k^*} \tilde{r}^s_{k-1}(v_k) \\ &= \frac{1}{\alpha\tilde{r}^s_{\max}} \sum_{k=1}^{k^*} \left(\|\tilde{r}^s_{k-1}\|_1 - \|\tilde{r}^s_k\|_1\right) = t\frac{1}{\alpha\tilde{r}^s_{\max}} \left(\|\tilde{r}^s_0\|_1 - \|\tilde{r}^s_{k^*}\|_1\right) \le \frac{1}{\alpha\tilde{r}^s_{\max}}, \end{aligned}$$

where the first inequality holds since $\tilde{r}^s_{\max} < \|D^{-1}\tilde{r}^s_k\|_\infty = \frac{\tilde{r}^s_{k-1}(v_k)}{d_{\text{out}}(v_k)}$ for $k \le k^*$ (i.e. for each k

before termination), the second equality holds by the previous display, and the final inequality holds since $\|\tilde{r}_0^s\|_1 = \|e_s\|_1 = 1$ and $\|\tilde{r}_{k^*}^s\|_1 \geq 0$ (the remaining steps are straightforward).

Using this, we bound the complexity of Algorithm 2.1. First, in Algorithm 2.1 we have

$$\|r^s\|_1 = \sum_{v \in V} \frac{r^s(v)}{d_{\text{out}}(v)} d_{\text{out}}(v) \leq m \max_{v \in V} \frac{r^s(v)}{d_{\text{out}}(v)} = m \|D^{-1} r^s\|_\infty,$$

so to guarantee termination of Algorithm 2.1 (i.e. to ensure $\|r^s\|_1 \leq r_{\max}^s$), it suffices to have $\|D^{-1} r^s\|_\infty \leq \frac{r_{\max}^s}{m}$. But from the analysis of Algorithm A.1, the complexity required to ensure $\|D^{-1} r^s\|_\infty \leq \frac{r_{\max}^s}{m}$ is $\frac{m}{\alpha r_{\max}^s}$; hence, the complexity of Algorithm 2.1 is at most $\frac{m}{\alpha r_{\max}^s}$.

Algorithm A.1: $(\tilde{p}^s, \tilde{r}^s) =$ `Approximate-PageRank-Original`$(G, s, \alpha, \tilde{r}_{\max}^s)$

Set $k = 0, \tilde{p}_k^s = 0, \tilde{r}_k^s = e_s$
while $\|D^{-1} \tilde{r}_k^s\|_\infty > \tilde{r}_{\max}^s$ **do**
  Set $k \leftarrow k + 1$; let $v_k \in \arg\max_{v \in V} \tilde{r}_{k-1}^s(v)/d_{\text{out}}(v)$
  Set $\tilde{r}_k^s(v) = \tilde{r}_{k-1}^s(v) + (1-\alpha)\tilde{r}_{k-1}^s(v_k)/d_{\text{out}}(v_k)$, $\tilde{p}_k^s(v) = \tilde{p}_{k-1}^s(v)\ \forall\ v \in N_{\text{out}}(v_k)$
  Set $\tilde{r}_k^s(v_k) = 0$, $\tilde{p}_k^s(v_k) = \tilde{p}_{k-1}^s(v_k) + \alpha \tilde{r}_{k-1}^s(v_k)$
  Set $\tilde{p}_k^s(v) = \tilde{p}_{k-1}^s(v)$, $\tilde{r}_k^s(v) = \tilde{r}_{k-1}^s(v)\ \forall\ v \in V \setminus (\{v_k\} \cup N_{\text{out}}(v_k))$

A.2 Practical version of `FW-BW-MCMC`

In this appendix, we define and analyze a modified version of `FW-BW-MCMC` that is more useful in practice. Before proceeding to the formal definition and analysis, we first motivate the practical algorithm. First, suppose for an instance of `FW-BW-MCMC` we have already run the backward DP (Algorithm 2.2) and we are currently running the forward DP (Algorithm 2.1). Though `FW-BW-MCMC` dictates we run the forward DP until $\|r^s\|_1 < r_{\max}^s$ for some predefined $r_{\max}^s$, we could instead terminate the forward DP (even if $\|r^s\|_1 > r_{\max}^s$) and proceed to the random walks. In other words, we dynamically change $r_{\max}^s$ from the predefined value to the current value of $\|r^s\|_1$. Then, if the number of walks sampled is $w = c\|r^s\|_1 r_{\max}^t/\delta$, where

$$c = 3(2e)^{1/3} \log(2/p_{\text{fail}})/\varepsilon^{7/3}, \tag{A.14}$$

the proof of Theorem A.1 goes through. Furthermore, this argument holds at any iteration of the forward DP. In other words, we can terminate the forward DP at any iteration and achieve the accuracy guarantee, as long as we scale w with the $\|r^s\|_1$ value obtained at termination. From this observation, we aim to terminate the forward DP at the "optimal" iteration, i.e. the iteration for which the overall complexity of the algorithm is minimized.

Towards determining this optimal iteration, let C_{FDP} denote the complexity of the forward DP until the current iteration, and define $C_{MCMC} = \frac{3(2e)^{1/3} r_{\max}^t \log(2/p_{\text{fail}})}{\alpha \delta \varepsilon^{7/3}}$, so that $\|r^s\|_1 C_{MCMC}$ gives the complexity of the MCMC stage (since $c\|r^s\|_1 r_{\max}^t/\delta$ walks are sampled, each in expected time $\frac{1}{\alpha}$, with c satisfying (A.14)). Then, if we terminate the forward DP at the current iteration, the combined complexity of forward DP and MCMC stages will be $C_{FDP} + \|r^s\|_1 C_{MCMC}$. Suppose instead that we decide to run one more iteration, i.e. to terminate the forward DP at the *next* iteration. Then, by Algorithm 2.1, the next iteration

will have complexity $d_{\text{out}}(v^*)$. Furthermore, by (A.13) in Appendix A.1.2, $\|r^s\|_1$ will decrease by $\alpha r^s(v^*)$ at the next iteration. Hence, if we run one more iteration, the combined complexity of forward DP and MCMC will be $(C_{FDP} + d_{\text{out}}(v^*)) + (\|r^s\|_1 - \alpha r^s(v^*))\, C_{MCMC}$. Now clearly, we should terminate the forward DP if and only if the resulting complexity is less than the complexity resulting from running another iteration, i.e. if and only if

$$\begin{aligned} C_{FDP} + \|r^s\|_1 C_{MCMC} < (C_{FDP} + d_{\text{out}}(v^*)) + (\|r^s\|_1 - \alpha r^s(v^*))\, C_{MCMC} \\ \Leftrightarrow r^s(v^*)/d_{\text{out}}(v^*) < 1/(\alpha C_{MCMC}). \end{aligned} \tag{A.15}$$

In other words, to optimize the tradeoff between forward DP and MCMC, we should run the forward DP until $\|D^{-1}r^s\|_\infty$ falls below the threshold in (A.15). This motivates the practical version of `FW-BW-MCMC` in Algorithm A.2. Algorithm A.2 changes two aspects of `FW-BW-MCMC`: it replaces Algorithm 2.1 with Algorithm A.1 (which uses $\|D^{-1}\tilde{r}^s\|_\infty$ termination), and it scales the the number of random walks sampled with $\|\tilde{r}^s\|_1$ (as discussed above.)

Algorithm A.2: $\hat{\pi}_s(t) =$ `FW-BW-MCMC-Practical`$(G, s, t, \alpha, \tilde{r}^s_{\max}, r^t_{\max}, w)$

1 Let $(p^t, r^t) =$ `Approximate-Contributions`$(G, t, \alpha, r^t_{\max})$ (Algorithm 2.2)
2 Let $(\tilde{p}^s, \tilde{r}^s) =$ `Approximate-PageRank-Original`$(G, s, \alpha, \tilde{r}^s_{\max})$ (Algorithm A.1); set $\tilde{\sigma}_s = \tilde{r}^s/\|\tilde{r}^s\|_1$
3 **for** $i = 1$ **to** $w\|\tilde{r}^s\|_1$ **do**
4 Sample random walk starting at $\nu \sim \tilde{\sigma}_s$ of length $\sim$ geom(α); let $X_i = r^t(U_i)$, where U_i is endpoint of walk
5 Let $\hat{\pi}_s(t) = p^t(s) + \langle \tilde{p}^s, r^t \rangle + \frac{1}{w}\sum_{i=1}^{w\|\tilde{r}^s\|_1} X_i$

We now establish accuracy and average-case complexity guarantees for Algorithm A.2.

Theorem A.5. Fix min. PPR threshold δ, error tolerance ε, failure probability p_{fail}. Let

$$\varepsilon \in \left(0, \frac{1}{\sqrt{2e}}\right), \quad w = \frac{c r^t_{\max}}{\delta}, \quad c > \frac{3(2e)^{1/3}\log(2/p_{\text{fail}})}{\varepsilon^{7/3}}. \tag{A.16}$$

Then the estimate $\hat{\pi}_s(t)$ produced by Algorithm A.2 satisfies (A.3)with probability $\geq 1 - p_{\text{fail}}$.

Proof. As discussed above, the proof of Theorem A.1 establishes this result. □

Theorem A.6. Fix minimum PPR threshold δ, relative error tolerance ε, and failure probability p_{fail}. Assume (A.16) holds. Then for any $s \in V$ and for $t \sim V$ uniformly, setting $\tilde{r}^s_{\max} = \frac{\delta\varepsilon^{7/3}}{r^t_{\max}\log(1/p_{\text{fail}})}$, $r^t_{\max} = \frac{\sqrt{m\delta}\varepsilon^{7/6}}{\sqrt{n\log(1/p_{\text{fail}})}}$ in Algorithm A.2 yields complexity $O\left(\frac{\sqrt{m\log(1/p_{\text{fail}})}}{\sqrt{n\delta}\alpha\varepsilon^{7/6}}\right)$.

Proof. For the backward DP (Algorithm 2.2), we use the result from [62], which we include for completeness. Recall from Appendix A.1.2 that the complexity of Algorithm 2.2 for $t \in V$ is bounded by $\sum_{v\in V} d_{\text{in}}(v)\frac{\pi_v(t)}{\alpha r^t_{\max}}$. Hence, for $t \sim V$ uniformly, the expected complexity is

$$\frac{1}{n}\sum_{t\in V}\sum_{v\in V} d_{\text{in}}(v)\frac{\pi_v(t)}{\alpha r^t_{\max}} = \frac{1}{n\alpha r^t_{\max}}\sum_{v\in V} d_{\text{in}}(v)\sum_{t\in V}\pi_v(t) = \frac{m}{n\alpha r^t_{\max}},$$

since $\sum_{t\in V}\pi_v(t)=1$ by definition. Next, we consider the complexity of the forward DP (Algorithm A.1). From Appendix A.1.2, for any $s\in V$ we have complexity $\frac{1}{\alpha\tilde{r}^s_{\max}}=\frac{r^t_{\max}\log(1/p_{\text{fail}})}{\alpha\delta\varepsilon^{7/3}}$. Finally, for the MCMC stage, we sample $w\|\tilde{r}^s\|_1\le w$ walks, where $w=cr^t_{\max}/\delta$ with c satisfying (A.16). Each walk is sampled in average time $\frac{1}{\alpha}$. Therefore, the MCMC stage complexity is $O(\frac{r^t_{\max}\log(1/p_{\text{fail}})}{\alpha\delta\varepsilon^{7/3}})$. Thus, the overall complexity of Algorithm A.2 is bounded by

$$O\left(\frac{m}{n\alpha r^t_{\max}}+\frac{r^t_{\max}\log(1/p_{\text{fail}})}{\alpha\delta\varepsilon^{7/3}}\right). \tag{A.17}$$

Substituting $r^t_{\max}$ given in the statement of the theorem yields the desired complexity bound. Further, viewing (A.17) as a function of $r^t_{\max}$, one can verify this $r^t_{\max}$ is the minimizer. □

A.3 Proof of Theorem 2.1

We first observe

$$\begin{aligned}
&\mathbb{P}\left[\sum_{v\in V}\max_{s\in S}X^{(w)}_s(v)>(1+\varepsilon)w\sum_{v\in V}\max_{s\in S}\sigma_s(v)\right]\\
&\quad\le\mathbb{P}\left[\cup_{s\in S,v\in V}\{X^{(w)}_s(v)>(1+\varepsilon)w\sigma_s(v)\}\right]\\
&\quad\le\sum_{s\in S,v\in V:\sigma_s(v)>0}\mathbb{P}\left[X^{(w)}_s(v)>(1+\varepsilon)w\sigma_s(v)\right],
\end{aligned}\tag{A.18}$$

where the second inequality holds since $X^{(w)}_s(v)\sim\text{Binomial}(w,\sigma_s(v))$ (so $X^{(w)}_s(v)=0$ when $\sigma_s(v)=0$). Again using this fact, we have by (A.4) from Theorem A.4 in Appendix A.1.1,

$$\mathbb{P}\left[X^{(w)}_s(v)>(1+\varepsilon)w\sigma_s(v)\right]\le\exp\left(-\frac{\varepsilon^2}{3}w\sigma_s(v)\right). \tag{A.19}$$

Combining (A.18) and (A.19), we obtain

$$\begin{aligned}
&\mathbb{P}\left[\sum_{v\in V}\max_{s\in S}X^{(w)}_s(v)>(1+\varepsilon)w\sum_{v\in V}\max_{s\in S}\sigma_s(v)\right]\le\sum_{s\in S,v\in V:\sigma_s(v)>0}\exp\left(-\frac{\varepsilon^2}{3}w\sigma_s(v)\right)\\
&\le\left(\max_{s\in S,v\in V:\sigma_s(v)>0}\left\{\exp\left(-\frac{\varepsilon^2}{3}w\sigma_s(v)\right)\right\}\right)\left(\sum_{s\in S,v\in V}1(\sigma_s(v)>0)\right)\\
&=\exp\left(\frac{-\varepsilon^2}{3}w\min_{s\in S,v\in V:\sigma_s(v)>0}\sigma_s(v)\right)\left(\sum_{s\in S,v\in V}1(\sigma_s(v)>0)\right)<p_{\text{fail}}/2,
\end{aligned}\tag{A.20}$$

where the final inequality holds by the bound on w in the statement of the theorem. For the lower tail, following the same steps used to obtain (A.20) gives

$$\mathbb{P}\left[\sum_{v\in V}\max_{s\in S}X^{(w)}_s(v)<(1-\varepsilon)w\sum_{v\in V}\max_{s\in S}\sigma_s(v)\right]<p_{\text{fail}}/2. \tag{A.21}$$

Finally, by the union bound, (A.20) and (A.21) together establish the theorem.

A.4 Proof of Theorem 2.2

The theorem relies on two key lemmas. The first (Lemma A.1) shows that the out-degrees in our stochastic block model (SBM) concentrate, in the sense that these degrees are all close to $p\sqrt{n}$ with high probability. The proof, deferred to Appendix A.4.1, is a modified version of a standard result for similar random graph families (such as the Erdős-Rényi model).

Lemma A.1. Let $\{G_n = (V_n, E_n)\}_{n\in\mathbb{N}:\sqrt{n}\in\mathbb{N}}$ be the sequence of SBMs defined in Section 2.4.1.1, with $p_n = p$ for some constant $p \in (0,1)$. For $\varepsilon, C > 0$, define the following events:

$$\mathcal{E}_{n,\varepsilon} = \cap_{v\in V_n} \left\{ d_{\text{out}}(v) \in \left((1-\varepsilon)p\sqrt{n}, (1+\varepsilon)p\sqrt{n}\right)\right\},$$
$$\mathcal{F}_{n,C} = \left\{ \max_{v\in V_n} d^-_{\text{out}}(v) \le Cq_n n \right\}, \quad \mathcal{G}_{n,C} = \left\{ \max_{v\in V_n} d^-_{\text{out}}(v) \le \frac{C\log n}{\log\log n} \right\}.$$

Then the following hold:

- If $q_n = o(1/\sqrt{n})$, then for any constant $\varepsilon > 0$, $\lim_{n\to\infty} \mathbb{P}(\mathcal{E}_{n,\varepsilon}) = 1$.
- If $q_n = \Omega(\log n/n)$, then for some constant $C > 0$, $\lim_{n\to\infty} \mathbb{P}(\mathcal{F}_{n,C}) = 1$.
- If $q_n = \Theta(1/n)$, then for some constant $C > 0$, $\lim_{n\to\infty} \mathbb{P}(\mathcal{G}_{n,C}) = 1$.

Proof. See Appendix A.4.1. □

The second lemma (Lemma A.2) contains bounds regarding $\sigma^s_k = r^s_k/\|r^s_k\|_1$, where r^s_k is the r^s vector in the k-th iteration of Algorithm 2.1. (Here and moving forward, we explicitly denote the iteration of Algorithm 2.1 via subscripts, as in Algorithm A.1 from Appendix A.1.2). In fact, these bounds hold more generally than will be required for the theorem; namely, we formulate the lemma for any deterministic graph on n nodes for which the out-degree condition $\mathcal{E}_{n,\varepsilon}$ holds. The proof is tedious so is deferred to Appendix A.4.2.

Lemma A.2. Let $G_n = (V_n = \{1,\dots,n\}, E_n)$ be a deterministic graph satisfying

$$d_{\text{out}}(v) \in \left((1-\varepsilon)p\sqrt{n}, (1+\varepsilon)p\sqrt{n}\right) \ \forall\ v \in V_n \tag{A.22}$$

for some $p, \varepsilon \in (0,1)$, and let $k \in \{1,\dots,\lceil (1-\varepsilon)^2 p\sqrt{n}(1-\alpha)/(2e)\rceil\}$. Then for any $s \in V_n$,

$$\sigma^s_k(v) < \frac{1}{\sqrt{n}} \frac{e}{(1-\alpha)(1-\varepsilon)^2 p - 2ek/\sqrt{n}} \ \forall\ v \in V_n,$$

and for any $S_n \subset V_n$ s.t. $s \in S_n$,

$$\sum_{v\in V_n\setminus S_n} \sigma^s_k(v) < \frac{\max_{s'\in S_n} |N_{\text{out}}(s')\setminus S_n|}{\sqrt{n}} \frac{1 + 2ek/\sqrt{n}}{(1-\varepsilon)p\left((1-\alpha)(1-\varepsilon)^2 p - 2ek/\sqrt{n}\right)}.$$

Proof. See Appendix A.4.2. □

We now turn to the proof of the theorem. First, suppose all sources belong to the same community, and consider the sub-case $q_n = o(1/\sqrt{n}), q_n = \Omega(\log n/n)$. Then for any

$\varepsilon \in (0,1)$, Lemma A.2 implies that any realization of G_n satisfying $\mathcal{E}_{n,\varepsilon}$ also satisfies

$$\begin{aligned}\sum_{v\in V_n} \max_{s\in S_n} \sigma_k^s(v) &\le \sum_{v\in S_n} \max_{s\in S_n} \sigma_k^s(v) + \sum_{s\in S_n} \sum_{v\in V_n\setminus S_n} \sigma_k^s(v) \\ &\le |S_n| \times \frac{1}{\sqrt{n}} \frac{e}{(1-\alpha)(1-\varepsilon)^2 p - 2ek/\sqrt{n}} \\ &\quad + |S_n| \times \frac{\max_{s\in S_n} d_{\text{out}}^-(s)}{\sqrt{n}} \frac{1+2ek/\sqrt{n}}{(1-\varepsilon)p\left((1-\alpha)(1-\varepsilon)^2 p - 2ek/\sqrt{n}\right)}.\end{aligned}$$

Recall α, ε, p are constants and $|S_n| = \sqrt{n}, k = o(\sqrt{n})$ in the statement of the theorem. Hence, for some $C'' > 0$ and all n large, any realization of G_n satisfying $\mathcal{E}_{n,\varepsilon}$ also satisfies

$$\sum_{v\in V_n} \max_{s\in S_n} \sigma_k^s(v) \le C'' \max_{s\in S_n} d_{\text{out}}^-(s).$$

Now let $C' > 0, C = C'C''$. Then for n large, any realization satisfying $\mathcal{E}_{n,\varepsilon} \cap \mathcal{F}_{n,C'}$ satisfies

$$\sum_{v\in V_n} \max_{s\in S_n} \sigma_k^s(v) \le C q_n n.$$

In other words, we have shown that for some $C > 0$ and any $C' > 0$,

$$\lim_{n\to\infty} \mathbb{P}\left(\sum_{v\in V_n} \max_{s\in S_n} \sigma_k^s(v) \le C q_n n \middle| \mathcal{E}_{n,\varepsilon}, \mathcal{F}_{n,C'} \right) = 1.$$

Finally, for C' satisfying the second statement of Lemma A.1, we obtain

$$\mathbb{P}\left(\sum_{v\in V_n} \max_{s\in S_n} \sigma_k^s(v) \le C q_n n \right) \ge \mathbb{P}\left(\sum_{v\in V_n} \max_{s\in S_n} \sigma_k^s(v) \le C q_n n \middle| \mathcal{E}_{n,\varepsilon}, \mathcal{F}_{n,C'} \right) \mathbb{P}\left(\mathcal{E}_{n,\varepsilon}, \mathcal{F}_{n,C'}\right) \to 1.$$

In the sub-case $q_n = \Theta(1/n)$, a similar argument implies that for some $C, C' > 0$,

$$\begin{aligned}&\mathbb{P}\left(\sum_{v\in V_n} \max_{s\in S_n} \sigma_k^s(v) \le \frac{C\log n}{\log\log n} \right) \\ &\quad \ge \mathbb{P}\left(\sum_{v\in V_n} \max_{s\in S_n} \sigma_k^s(v) \le \frac{C\log n}{\log\log n} \middle| \mathcal{E}_{n,\varepsilon}, \mathcal{G}_{n,C'} \right) \mathbb{P}\left(\mathcal{E}_{n,\varepsilon}, \mathcal{G}_{n,C'}\right) \to 1.\end{aligned}$$

We next consider the case for which all sources belong to different communities, i.e. $S_n = \{\sqrt{n}, 2\sqrt{n}, \ldots, n\}$ (which is without loss of generality by symmetry). Then clearly

$$\sum_{v\in V_n} \max_{s\in S_n} \sigma_k^s(v) \ge \sum_{i=1}^{\sqrt{n}} \sum_{v=1+(i-1)\sqrt{n}}^{i\sqrt{n}} \sigma_k^{i\sqrt{n}}(v). \tag{A.23}$$

Further, for any $\varepsilon \in (0,1)$, Lemma A.2 implies that any realization satisfying $\mathcal{E}_{n,\varepsilon}$ satisfies

$$\sum_{v=1}^{\sqrt{n}} \sigma_k^{\sqrt{n}}(v) \geq 1 - \frac{\max_{v\in V_n} d_{\text{out}}^-(v)}{\sqrt{n}} \frac{1+2ek/\sqrt{n}}{(1-\varepsilon)p\left((1-\alpha)(1-\varepsilon)^2 p - 2ek/\sqrt{n}\right)}.$$

Now suppose $q_n = o(1/\sqrt{n}), q_n = \Omega(\log n/n)$, and let $\delta \in (0,1)$ be a constant. Then for $C > 0$ and n sufficiently large, any realization satisfying $\mathcal{E}_{n,\varepsilon}$ and $\mathcal{F}_{n,C}$ will also satisfy

$$\sum_{v=1}^{\sqrt{n}} \sigma_k^{\sqrt{n}}(v) \geq 1 - \frac{Cq_n n}{\sqrt{n}} = 1 - Cq_n\sqrt{n} \geq 1 - \delta.$$

where we again used the fact that α, ε, p are constant and $k = o(\sqrt{n})$. The same argument holds for each $i \in \{2, \ldots, \sqrt{n}\}$ in (A.23). It follows that, for appropriate choice of $C > 0$,

$$\lim_{n\to\infty} \mathbb{P}\left(\sum_{v\in V_n} \max_{s\in S_n} \sigma_k^s(v) \geq (1-\delta)\sqrt{n} \middle| \mathcal{E}_{n,\varepsilon}, \mathcal{F}_{n,C}\right) \mathbb{P}\left(\mathcal{E}_{n,\varepsilon}, \mathcal{F}_{n,C}\right) = 1.$$

A similar approach establishes the desired result in the case $q_n = \Theta(1/n)$.

Note that the only feature of the SBM used above was the degree concentration of Lemma A.1. In other words, we considered the number of edges for each node, while ignoring how exactly these edges were connected. Consequently, the same analysis can be used to obtain results for sequences of *deterministic* graphs $\{G_n = (V_n, E_n)\}_{n\in\mathbb{N}:\sqrt{n}\in\mathbb{N}}$. For example, if such a sequence satisfies $\mathcal{E}_{n,\varepsilon}, \mathcal{G}_{n,C}$ for some constants ε, C and for all n large, the analysis above implies $\|\Sigma_{S_n}\|_{\infty,1} = O(\log n/\log\log n)$ when $\sqrt{n}$ sources belong to the same community, whereas $\|\Sigma_{S_n}\|_{\infty,1} = \Omega(\sqrt{n})$ when $\sqrt{n}$ sources belong to different communities.

A.4.1 Proof of Lemma A.1

For the first statement, we begin by showing $d_{\text{out}}(1)$ concentrates around $p\sqrt{n}$; we will then use the union bound to establish the lemma. Towards this end, first note that since edges from node 1 to each $v \in \{2, \ldots, \sqrt{n}\}$ are present with probability p, and since edges from node 1 to each $v \in \{\sqrt{n}+1, \ldots, n\}$ are present with probability q_n, we have

$$\mathbb{E}\left[d_{\text{out}}(1)\right] = p\left(\sqrt{n}-1\right) + q_n\left(n-\sqrt{n}\right) = p\sqrt{n} + \left(q_n\left(n-\sqrt{n}\right) - p\right). \tag{A.24}$$

Next, since $q_n = o(1/\sqrt{n})$ and p is constant by assumption, we have for n sufficiently large,

$$\frac{q_n(n-\sqrt{n})-p}{p\sqrt{n}} \leq \frac{\varepsilon/2}{1+\varepsilon/2} \quad \Rightarrow \quad \left(1+\frac{\varepsilon}{2}\right)\left(q_n(n-\sqrt{n})-p\right) \leq \frac{\varepsilon}{2}p\sqrt{n}.$$

Thus, combining the previous two lines, we obtain (for such n),

$$\left(1+\frac{\varepsilon}{2}\right)\mathbb{E}\left[d_{\text{out}}(1)\right] = \left(1+\frac{\varepsilon}{2}\right)p\sqrt{n} + \left(1+\frac{\varepsilon}{2}\right)\left(q_n\left(n-\sqrt{n}\right)-p\right) \leq (1+\varepsilon)p\sqrt{n}.$$

We can then use monotonicity and (A.4) from Appendix A.1.1 to obtain

$$\mathbb{P}\left(d_{\text{out}}(1) > (1+\varepsilon)p\sqrt{n}\right) \leq \mathbb{P}\left(d_{\text{out}}(1) > \left(1+\frac{\varepsilon}{2}\right)\mathbb{E}\left[d_{\text{out}}(1)\right]\right) \leq \exp\left(-\frac{\varepsilon^2 p}{12}\sqrt{n}\right),$$

where we also used $\mathbb{E}[d_{\text{out}}(1)] \geq p\sqrt{n}$ by (A.24). Using the same argument for the lower tail, and then using the union bound, we thus obtain

$$\mathbb{P}\left(d_{\text{out}}(1) \notin \left[(1-\varepsilon)p\sqrt{n}, (1+\varepsilon)p\sqrt{n}\right]\right) \leq 2\exp\left(-\frac{\varepsilon^2 p}{12}\sqrt{n}\right).$$

Since also $\{d_{\text{out}}(v)\}_{v\in V}$ are identically-distributed, and by the union bound,

$$\mathbb{P}\left(\cup_{v\in V}\left\{d_{\text{out}}(v) \notin \left[(1-\varepsilon)p\sqrt{n}, (1+\varepsilon)p\sqrt{n}\right]\right\}\right) \leq 2n\exp\left(-\frac{\varepsilon^2 p}{12}\sqrt{n}\right) \xrightarrow[n\to\infty]{} 0,$$

which completes the proof of the first statement.

For the second statement, we similarly begin with a tail bound for $d^-_{\text{out}}(1)$. First note that, since $q_n = \Omega(\log n/n)$, we can find $C' > 0$ such that for all n sufficiently large, $q_n n > C' \log n$. Now let $C > \max\{2e, 2/(C'\log 2)\}$. Then clearly

$$Cq_n n > 2eq_n\left(n - \sqrt{n}\right) = 2e\mathbb{E}\left[d^-_{\text{out}}(1)\right].$$

Hence, we can use (A.5) from Appendix A.1.1 to obtain

$$\mathbb{P}\left(d^-_{\text{out}}(1) > Cq_n n\right) \leq 2^{-Cq_n n}.$$

By the union bound argument used above, we then have

$$\mathbb{P}\left(\max_{v\in V_n} d^-_{\text{out}}(v) > Cq_n n\right) \leq n\mathbb{P}\left(d^-_{\text{out}}(1) > Cq_n n\right) = 2^{-(Cq_n n - \log_2 n)}.$$

Also, by our choice of C and for n sufficiently large (so that $q_n n > C'\log n$),

$$Cq_n n - \log_2 n > \frac{2}{C'\log 2}C'\log n - \log_2 n = \log_2 n.$$

Combining the previous two inequalities then yields, for n sufficiently large,

$$\mathbb{P}\left(\max_{v\in V_n} d^-_{\text{out}}(v) > Cq_n n\right) \leq 1/n,$$

from which the second statement clearly follows.

For the third statement, we again derive a tail bound for $d^-_{\text{out}}(1)$, but this requires a different approach. First, for any $M \in \{1, \ldots, \lfloor n - \sqrt{n}\rfloor\}$, the event $\{d^-_{\text{out}}(1) \geq M\}$ means that node 1 has outgoing edges to M nodes in other communities, so

$$\mathbb{P}\left(d^-_{\text{out}}(1) \geq M\right) = \mathbb{P}\left(\cup_{U_n \subset \{1+\sqrt{n},\ldots,n\}:|U_n|=M}\{1 \to u \in E_n \ \forall\ u \in U_n\}\right)$$

$$\leq \sum_{U_n \subset \{1+\sqrt{n},\ldots,n\}:|U_n|=M} \mathbb{P}\left(1 \to u \in E_n \ \forall\ u \in U_n\right)$$
$$= \binom{n-\sqrt{n}}{M} q_n^M \leq \binom{n}{M} q_n^M,$$

where the first inequality is the union bound, the second equality holds by definition of our SBM, and the inequality is immediate. Now by assumption $q_n = \Theta(1/n)$, we can find C_1 such that $q_n n \leq C_1$ for n sufficiently large; combined with the standard binomial coefficient approximation $\binom{n}{M} \leq (\frac{ne}{M})^M$, we can further bound the above as

$$\mathbb{P}\left(d^-_{\text{out}}(1) \geq M\right) \leq \left(\frac{nq_n e}{M}\right)^M \leq \left(\frac{C_2}{M}\right)^M$$

for all n large (we also defined $C_2 = C_1 e$). Thus, by the union bound and the fact that $\{d^-_{\text{out}}(v)\}_{v \in V}$ are identically-distributed, we obtain for all n large and any constant $C > 0$,

$$\mathbb{P}\left(\max_{v \in V_n} d^-_{\text{out}}(v) \geq \frac{C \log n}{\log\log n}\right) \leq n \left(\frac{C_2 \log\log n}{C \log n}\right)^{C \log n / \log\log n}.$$

Next, we note

$$\log\left(n \left(\frac{C_2 \log\log n}{C \log n}\right)^{C \log n/\log\log n}\right) = \log n + \frac{C \log n}{\log\log n}\left(\log(C_2 \log\log n) - \log(C \log n)\right)$$
$$= \log n \left(1 + \frac{C \log\log(\log n)^{C_2}}{\log\log n} - \frac{C \log\log n^C}{\log\log n}\right).$$

Choosing any $C \geq 1$ clearly implies

$$(\log\log n^C)/(\log\log n) \geq 1.$$

Also, since $C_2 > 0$ is a constant, we have for all n large (for example)

$$(\log\log(\log n)^{C_2})/(\log\log n) < \frac{1}{2}.$$

Combining the previous four lines, we then obtain, for all n large,

$$\log \mathbb{P}\left(\max_{v \in V_n} d^-_{\text{out}}(v) > \frac{C \log n}{\log\log n}\right) \leq (1 - C/2)\log n,$$

so that choosing any $C > 2$ establishes the third statement.

A.4.2 Proof of Lemma A.2

We begin with another lemma, which in fact holds for any underlying graph G.

Lemma A.3. For any graph $G = (V, E)$, any source node $s \in V$, and any iteration $k \in$

$\{1, \ldots, d_{\text{out}}(s)\}$ of Algorithm 2.1,

$$\frac{1-\alpha}{\max_{v \in V} d_{\text{out}}(v)} \leq \max_{v \in V} r_k^s(v) \leq \frac{1-\alpha}{\min_{v \in V} d_{\text{out}}(v)} \exp\left(\frac{(1-\alpha)(k-1)}{\min_{v \in V} d_{\text{out}}(v)}\right).$$

Proof. For the lower bound, first note $r_1^s(v) = (1-\alpha)/d_{\text{out}}(s)\ \forall\ v \in N_{\text{out}}(s)$. Furthermore, for each such v, $r_k^s(v)$ is non-decreasing in k for $k < k_v$, where k_v is the first iteration k for which $v_k^* = v$. Also, since $v_1^* = s$, we must have $k_v \geq d_{\text{out}}(s) + 1$ for some $v \in N_{\text{out}}(s)$. Hence, for any $k \in \{1, \ldots, d_{\text{out}}(s)\}$, we can find some $v \in N_{\text{out}}(s)$ for which $k_v > k$, which implies $r_k^s(v) \geq r_1^s(v) = (1-\alpha)/d_{\text{out}}(s)$. Since also $d_{\text{out}}(s) \leq \max_{v \in V} d_{\text{out}}(v)$, the lower bound follows. For the upper bound, we use induction. For the base of induction, simply note

$$r_1^s(v) = \frac{1-\alpha}{d_{\text{out}}(s)} \leq \frac{1-\alpha}{\min_{v \in V} d_{\text{out}}(v)}\ \forall\ v \in N_{\text{out}}(s).$$

Now assuming the upper bound holds for $k-1$, we have for any $v \in V$,

$$\begin{aligned} r_k^s(v) &\leq r_{k-1}^s(v) + \frac{1-\alpha}{d_{\text{out}}(v_k^*)} r_{k-1}^s(v_k^*) \leq \left(1 + \frac{1-\alpha}{\min_{v \in V} d_{\text{out}}(v)}\right) \max_{v' \in V} r_{k-1}^s(v') \\ &\leq \left(1 + \frac{1-\alpha}{\min_{v \in V} d_{\text{out}}(v)}\right) \frac{1-\alpha}{\min_{v \in V} d_{\text{out}}(v)} \exp\left(\frac{(1-\alpha)(k-2)}{\min_{v \in V} d_{\text{out}}}(v)\right) \\ &\leq \frac{1-\alpha}{\min_{v \in V} d_{\text{out}}(v)} \exp\left(\frac{(1-\alpha)(k-1)}{\min_{v \in V} d_{\text{out}}(v)}\right), \end{aligned}$$

where the first inequality uses the iterative update in Algorithm 2.1, the second is immediate, the third uses the inductive hypothesis, and the fourth uses $1 + x \leq e^x$. □

We next state and prove a corollary of Lemma A.3, which translates the r_k^s bounds from Lemma A.3 to bounds regarding σ_k^s (the actual vector of interest in the theorem).

Corollary A.1. Let $G_n = (V_n = \{1, \ldots, n\}, E_n)$ be a graph satisfying

$$d_{\text{out}}(v) \in \left((1-\varepsilon)p\sqrt{n}, (1+\varepsilon)p\sqrt{n}\right)\ \forall\ v \in V_n \tag{A.25}$$

for some $p, \varepsilon \in (0,1)$. Then for any $k \in \{1, \ldots, \lfloor (1-\varepsilon)p\sqrt{n} \rfloor\}$ and any $s \in V_n$,

$$\begin{aligned} \frac{1-\alpha}{2\sqrt{n}} < \max_{v \in V_n} r_k^s(v) < \frac{e}{(1-\varepsilon)p\sqrt{n}}, \\ \frac{(1-\varepsilon)(1-\alpha)}{4\sqrt{n}} < r_k^s(v_{k+1}^*) < \frac{2e}{(1-\varepsilon)^2 p\sqrt{n}}, \\ (1-\alpha) - \frac{2e(k-1)}{(1-\varepsilon)^2 p\sqrt{n}} \leq \|r_k^s\|_1 \leq (1-\alpha) - \frac{(k-1)(1-\varepsilon)(1-\alpha)}{4\sqrt{n}}. \end{aligned}$$

Proof. Fix $k \in \{1, \ldots, \lfloor (1-\varepsilon)p\sqrt{n} \rfloor\}$ and $s \in V_n$. Then $k < d_{\text{out}}(s)$ by (A.25) and the choice

of k. We can then use the assumption (A.25), Lemma A.3, and the choice of k to obtain

$$\frac{1-\alpha}{(1+\varepsilon)p\sqrt{n}} < \frac{1-\alpha}{\max_{v\in V} d_{\text{out}}(v)} \leq \max_{v\in V} r_k^s(v)$$
$$\leq \frac{1-\alpha}{\min_{v\in V} d_{\text{out}}(v)} \exp\left(\frac{(1-\alpha)(k-1)}{\min_{v\in V} d_{\text{out}}(v)}\right) < \frac{(1-\alpha)e^{1-\alpha}}{(1-\varepsilon)p\sqrt{n}}.$$

Finally, $\varepsilon, p, \alpha \in (0,1)$ yields the first pair of inequalities. Next, by definition of v_{k+1}^*,

$$r_k^s(v_{k+1}^*) = d_{\text{out}}(v_{k+1}^*)\frac{r_k^s(v_{k+1}^*)}{d_{\text{out}}(v_{k+1}^*)} = d_{\text{out}}(v_{k+1}^*)\max_{v\in V_n}\frac{r_k^s(v)}{d_{\text{out}}(v)} = \max_{v\in V_n}\frac{d_{\text{out}}(v_{k+1}^*)}{d_{\text{out}}(v)} r_k^s(v). \quad \text{(A.26)}$$

On the other hand, by the assumption (A.25), and since $\varepsilon \in (0,1)$,

$$\frac{1-\varepsilon}{2} < \frac{1-\varepsilon}{1+\varepsilon} < \frac{d_{\text{out}}(v_{k+1}^*)}{d_{\text{out}}(v)} < \frac{1+\varepsilon}{1-\varepsilon} < \frac{2}{1-\varepsilon} \quad \text{(A.27)}$$

Combining (A.26) and (A.27), and using the first pair of inequalities, yields the second pair of inequalities. For the third pair of inequalities, we first assume $k > 1$ and use (A.13) from Appendix A.1.2 to obtain

$$\|r_k^s\|_1 = \|r_{k-1}^s\|_1 - \alpha r_{k-1}^s(v_k^*) = \cdots = \|r_0^s\|_1 - \alpha r_0^s(v_1^*) - \alpha\sum_{j=1}^{k-1} r_j^s(v_{j+1}^*) = 1-\alpha-\alpha\sum_{j=1}^{k-1} r_j^s(v_{j+1}^*),$$

where we also used $r_0^s = e_s, v_1^* = s$ by Algorithm 2.1. We can then use the second pair of inequalities to obtain the third pair of inequalities. If instead $k = 1$, we immediately have $\|r_k^s\|_1 = 1-\alpha$, which is precisely the third pair of inequalities in the case $k = 1$. □

We can now prove Lemma A.2. For the first bound, note the assumptions of Lemma A.2 are stronger than those of Corollary A.1, so we can use Corollary A.1 to obtain

$$\sigma_k^s(v) = \frac{r_k^s(v)}{\|r_k^s\|_1} < \frac{\frac{e}{(1-\varepsilon)p\sqrt{n}}}{(1-\alpha) - \frac{2e(k-1)}{(1-\varepsilon)^2 p\sqrt{n}}} = \frac{1}{\sqrt{n}}\frac{(1-\varepsilon)e}{(1-\alpha)(1-\varepsilon)^2 p - 2e(k-1)/\sqrt{n}}.$$

Using the trivial inequalities $1-\varepsilon < 1, k-1 < k$ then yields the first upper bound. (Note the assumed upper bound on k ensures the denominator is non-negative.)

For the second bound, let $S_n \subset V_n$ be a set containing s. We begin by showing

$$\sum_{v\in V_n\setminus S_n} r_k^s(v) < \frac{\sqrt{n}+2e(k-1)}{(1-\varepsilon)^3 p^2 n}\max_{s'\in S_n}|N_{\text{out}}(s')\setminus S_n|. \quad \text{(A.28)}$$

To prove (A.28), we use induction. For $k = 1$, the r^s update in Algorithm 2.1 implies

$$r_1^s(v) = \frac{1-\alpha}{d_{\text{out}}(s)}1(v\in N_{\text{out}}(s)) \Rightarrow \sum_{v\in V_n\setminus S_n} r_k^s(v) = \frac{1-\alpha}{d_{\text{out}}(s)}|N_{\text{out}}(s)\setminus S_n|,$$

which, using the assumption (A.22) and $\alpha, \varepsilon, p \in (0,1)$, can clearly be bounded as

$$\sum_{v \in V_n \setminus S_n} r_k^s(v) < \frac{1}{(1-\varepsilon)^3 p^2 \sqrt{n}} \max_{s' \in S_n} |N_{\text{out}}(s') \setminus S_n| ,$$

which proves (A.28) when $k = 1$. Now assume (A.28) holds for $k-1$ and consider two cases:

1. $v_k^* \notin S_n$: We can write the r^s update in Algorithm 2.1 as

$$r_k^s(v) = r_{k-1}^s(v) + \frac{1-\alpha}{d_{\text{out}}(v_k^*)} r_{k-1}(v_k^*) 1(v \in N_{\text{out}}(v_k^*)) - r_{k-1}(v_k^*) 1(v = v_k^*) \; \forall \, v \in V_n, \quad \text{(A.29)}$$

   where $1(A)$ is the indicator function of the event A. This clearly implies

$$\begin{aligned} \sum_{v \in V_n \setminus S_n} r_k^s(v) &= \sum_{v \in V_n \setminus S_n} r_{k-1}^s(v) + r_{k-1}(v_k^*) \left(\frac{1-\alpha}{d_{\text{out}}(v_k^*)} |N_{\text{out}}(v_k^*) \setminus S_n| - 1 \right) \\ &< \sum_{v \in V_n \setminus S_n} r_{k-1}^s(v), \end{aligned}$$

   from which the inductive hypothesis completes the proof.

2. $v_k^* \in S_n$: Again using (A.29), we observe

$$\sum_{v \in V_n \setminus S_n} r_k^s(v) = \sum_{v \in V_n \setminus S_n} r_{k-1}^s(v) + \frac{1-\alpha}{d_{\text{out}}(v_k^*)} r_{k-1}(v_k^*) |N_{\text{out}}(v_k^*) \setminus S_n| . \quad \text{(A.30)}$$

   (Note the final term in (A.29) does not appear in (A.30), since $\sum_{v \in V_n \setminus S_n} 1(v = v_k^*) = 0$ when $v_k^* \in S_n$.) For the second summand in (A.30), the second upper bound from Corollary A.1, the assumption (A.22), $\alpha \in (0,1)$, and the assumption $v_k^* \in S_n$ imply

$$\frac{1-\alpha}{d_{\text{out}}(v_k^*)} r_{k-1}(v_k^*) |N_{\text{out}}(v_k^*) \setminus S_n| < \frac{2e}{(1-\varepsilon)^3 p^2 n} \max_{s' \in S_n} |N_{\text{out}}(s') \setminus S_n| ,$$

   Substituting into (A.30) and using the inductive hypothesis yields

$$\sum_{v \in V_n \setminus S_n} r_k^s(v) < \left(\frac{\sqrt{n} + 2e(k-2)}{(1-\varepsilon)^3 p^2 n} + \frac{2e}{(1-\varepsilon)^3 p^2 n} \right) \max_{s' \in S} |N_{\text{out}}(s') \setminus S_n| ,$$

   which completes the proof.

Combining (A.28) with the lower bound for $\|r_k^s\|_1$ from Corollary A.1 gives

$$\begin{aligned} \sum_{v \in V_n \setminus S_n} \sigma_k^s(v) = \frac{\sum_{v \in V_n \setminus S_n} r_k^s(v)}{\|r_k^s\|_1} &< \frac{\frac{\sqrt{n}+2e(k-1)}{(1-\varepsilon)^3 p^2 n} \max_{s' \in S} |N_{\text{out}}(s') \setminus S_n|}{(1-\alpha) - \frac{2e(k-1)}{(1-\varepsilon)^2 p \sqrt{n}}} \\ &= \frac{\max_{s' \in S_n} |N_{\text{out}}(s') \setminus S_n|}{\sqrt{n}} \frac{1 + 2e(k-1)/\sqrt{n}}{(1-\varepsilon) p \left((1-\alpha)(1-\varepsilon)^2 p - 2e(k-1)/\sqrt{n} \right)}, \end{aligned}$$

from which the trivial bound $k-1<k$ completes the proof.

A.5 Proof of Proposition 2.1

First, assume `Merge` is used at each iteration for which $v^* = t_2$. By Algorithm 2.2, $\|p^{t_2}\|_1$ increases by at least $\alpha r^t_{\max}$ at each iteration for which $v^* \neq t_1$. By (2.9), $\|p^{t_2}\|_1$ increases by at least $\|p^{t_1}\|_1 r^t_{\max}$ at each iteration for which $v^* = t_1$. Let us define I_1 as the number of iterations for which $v^* \neq t_1$, I_2 as the number of iterations for which $v^* = t_1$, and $I = I_1 + I_2$ as the total number of iterations. Since $\|p^{t_2}\|_1 = 0$ at the start of Algorithm 2.2 and $\|p^{t_2}\|_1 \leq n\pi(t_2)$ by the invariant (2.2), we have

$$\frac{n\pi(t_2)}{r^t_{\max}} \geq \alpha I_1 + \|p^{t_1}\|_1 I_2 = \alpha I + (\|p^{t_1}\|_1 - \alpha) I_2. \tag{A.31}$$

Now at termination of Algorithm 2.2, $\|r^{t_2}\|_\infty \leq r^t_{\max}$, so by the invariant (2.2), $\pi_{t_1}(t_2) \leq p^{t_2}(t_1) + r^t_{\max}$ at termination. Therefore, if $\pi_{t_1}(t_2) > r^t_{\max}$, $p^{t_2}(t_1) > 0$ at termination, which can only occur if $v^* = t_1$ at some iteration. Hence, $\pi_{t_1}(t_2) > r^t_{\max} \Rightarrow I_2 \geq 1$. Finally, from Algorithm 2.2, $\|p^{t_1}\|_1 \geq \alpha$. Substituting into (A.31) gives $I \leq \frac{n\pi(t_2)}{\alpha r^t_{\max}} - \frac{(\|p^{t_1}\|_1 - \alpha)}{\alpha}$.

If instead `Merge` is not used, $\|p^{t_2}\|_1$ increases by at least $\alpha r^t_{\max}$ at *every* iteration. Hence, the same argument establishes that the total number of iterations is bounded by $\frac{n\pi(t_2)}{\alpha r^t_{\max}}$.

A.6 Proof of Theorem 2.3

We will use Corollary 6.2.1 from [42], which (applied to our setting) states the following. Assume $\{X_i\}_{i=1}^w$ are independent random matrices satisfying $\mathbb{E}[X_i] = R_S^\mathsf{T} \Pi R_T$. Let M be s.t. $\|X_i\|_2 \leq M$ a.s., and let $m_2(X_i) = \max\{\|\mathbb{E}[X_i X_i^\mathsf{T}]\|_2, \|\mathbb{E}[X_i^\mathsf{T} X_i]\|_2\}$. Then $\forall\ \eta > 0$,

$$\mathbb{P}\left[\left\| R_S^\mathsf{T} \Pi R_t - \frac{1}{w}\sum_{i=1}^w X_i \right\|_2 > \eta\right] \leq 2l \exp\left(\frac{-3w\eta^2}{6m_2(X_i) + 4M\eta}\right).$$

We have verified the independence and $\mathbb{E}[X_i] = R_S^\mathsf{T}\Pi R_T$ assumptions in the main text. Also, from (2.15) and Algorithm 2.5, $\Pi(S,T) - \hat{\Pi}(S,T) = R_S^\mathsf{T}\Pi R_t - \frac{1}{w}\sum_{i=1}^w X_i$. Thus,

$$\begin{aligned}
&\mathbb{P}\left[\left\|\Pi(S,T) - \hat{\Pi}(S,T)\right\|_2 > \varepsilon \max\{\|\Pi(S,T)\|_2, 1\}\right] \\
&\quad\leq 2l \exp\left(\frac{-3w\left(\varepsilon \max\{\|\Pi(S,T)\|_2, 1\}\right)^2}{6m_2(X_i) + 4M\varepsilon \max\{\|\Pi(S,T)\|_2, 1\}}\right) \\
&\quad\leq 2l \exp\left(\frac{-3w\varepsilon^2}{6\frac{m_2(X_i)}{\max\{\|\Pi(S,T)\|_2,1\}} + 4M\varepsilon}\right) \leq 2l \exp\left(\frac{-3w\varepsilon^2}{6\frac{m_2(X_i)}{\|\Pi(S,T)\|_2} + 4M\varepsilon}\right).
\end{aligned} \tag{A.32}$$

where we have also used $\max\{\|\Pi(S,T)\|_2, 1\} \geq 1$, $\max\{\|\Pi(S,T)\|_2, 1\} \geq \|\Pi(S,T)\|_2$.

Now to prove the theorem, we aim to find M s.t. $\|X_i\|_2 \leq M$ a.s. and to compute $m_2(X_i)$ such that (A.32) is bounded by p_{fail}, in each of the following cases:

$$\text{(Case 1)} \quad \sigma = \sigma_{\text{avg}}, \quad w \geq \frac{l^2\sqrt{\text{srank}(\Pi(S,T))}\log(2l/p_{\text{fail}}) r^s_{\max} r^t_{\max}(6+4\varepsilon)}{3\varepsilon^2}. \tag{A.33}$$

$$\text{(Case 2)} \quad \sigma = \sigma_{\max}, \quad w \geq \frac{l^{3/2}\|\Sigma\|_{\infty,1}\log(2l/p_{\text{fail}})r^s_{\max}r^t_{\max}(6+4\varepsilon)}{3\varepsilon^2}. \tag{A.34}$$

We begin with Case 1. By Lemma A.4, we may take $M = l^{3/2}r^s_{\max}r^t_{\max}$, and by Lemma A.5, we have $m_2(X_i) \leq l^2 r^s_{\max}r^t_{\max}\|\Pi(S,T)\|_F$. We can then write

$$\begin{aligned} 6\frac{m_2(X_i)}{\|\Pi(S,T)\|_2} + 4M\varepsilon &\leq l^2 r^s_{\max}r^t_{\max}\left(6\frac{\|\Pi(S,T)\|_F}{\|\Pi(S,T)\|_2} + \frac{4}{\sqrt{l}}\varepsilon\right) \\ &= l^2 r^s_{\max}r^t_{\max}\left(6\sqrt{\text{srank}(\Pi(S,T))} + \frac{4}{\sqrt{l}}\varepsilon\right) \\ &\leq l^2 r^s_{\max}r^t_{\max}\sqrt{\text{srank}(\Pi(S,T))}(6+4\varepsilon) \leq \frac{3w\varepsilon^2}{\log(2l/p_{\text{fail}})}, \end{aligned} \tag{A.35}$$

where the penultimate inequality holds since $l, \text{srank}(\Pi(S,T)) \geq 1$, and the final inequality is by (A.33). Substituting (A.35) into (A.32) establishes the desired result.

For Case 2, we take $M = l^{3/2}\|\Sigma\|_{\infty,1}r^s_{\max}r^t_{\max}$ (Lemma A.4), and by Lemma A.5 we have

$$m_2(X_i) \leq l\|\Sigma\|_{\infty,1}r^s_{\max}r^t_{\max}\max\{\|\Pi(S,T)\|_\infty, \|\Pi(S,T)\|_1\}.$$

We then obtain

$$\begin{aligned} 6\frac{m_2(X_i)}{\|\Pi(S,T)\|_2} + 4M\varepsilon &\leq l\|\Sigma\|_{\infty,1}r^s_{\max}r^t_{\max}\left(6\frac{\max\{\|\Pi(S,T)\|_\infty, \|\Pi(S,T)\|_1\}}{\|\Pi(S,T)\|_2} + 4\sqrt{l}\varepsilon\right) \\ &\leq l^{3/2}\|\Sigma\|_{\infty,1}r^s_{\max}r^t_{\max}(6+4\varepsilon) \leq \frac{3w\varepsilon^2}{\log(2l/p_{\text{fail}})} \end{aligned} \tag{A.36}$$

where the second inequality is $\|A\|_\infty, \|A\|_1 \leq \sqrt{l}\|A\|_2 \;\forall\; A \in \mathbb{R}^{l\times l}$), and the third inequality is by (A.34). Substituting (A.36) into (A.32) completes the proof.

Lemma A.4. If $\sigma = \sigma_{\text{avg}}$, $\|X_i\|_2 \leq l^{3/2}r^s_{\max}r^t_{\max}$; if $\sigma = \sigma_{\max}$, $\|X_i\|_2 \leq l^{3/2}\|\Sigma\|_{\infty,1}r^s_{\max}r^t_{\max}$.

Proof. Observe $X_i = a_i b_i^\mathsf{T}$, where $a_i, b_i \in \mathbb{R}^l$ with $a_i(j) = r^{s_j}(\mu_i)/\sigma(\mu_i), b_i(j) = r^{t_j}(\nu_i)$. X_i has rank 1, and we may write its singular value decomposition as

$$X_i = (\|a_i\|_2\|b_i\|_2)\left(\frac{a_i}{\|a_i\|_2}\right)\left(\frac{b_i}{\|b_i\|_2}\right)^\mathsf{T},$$

so the nonzero singular value of X_i is $\|a_i\|_2\|b_i\|_2$. Using the well-known fact that a matrix's 2-norm equals its largest singular value, $\|X_i\|_2 = \|a_i\|_2\|b_i\|_2$, so we seek bounds on $\|a_i\|_2$ and $\|b_i\|_2$. First, if $\sigma = \sigma_{\text{avg}}$, we have

$$\sigma(\mu_i) = \frac{1}{l}\sum_{s\in S}\frac{r^s(\mu_i)}{\|r^s\|_1} \geq \frac{1}{lr^s_{\max}}\sum_{s\in S}r^s(\mu_i) \geq \frac{1}{lr^s_{\max}}\left(\sum_{s\in S}r^s(\mu_i)^2\right)^{1/2} = \frac{1}{lr^s_{\max}}\|a_i\|_2\sigma(\mu_i). \tag{A.37}$$

Here the first equality holds by definition (2.17), the first inequality uses the terminating condition of Algorithm 2.1 ($\|r^s\|_1 \leq r^s_{\max}$), and the second equality is by definition of a_i. We

conclude $\|a_i\|_2 \le l r^s_{\max}$. To bound $\|b_i\|_2$, we have

$$\|b_i\|_2 \le \sqrt{l}\|b_i\|_\infty \le \sqrt{l} r^t_{\max}, \tag{A.38}$$

where we have used a well-known vector norm inequality and the terminating condition of Algorithm 2.2 ($\|r^t\|_\infty \le r^t_{\max}$). Hence, $\|X_i\|_2 \le l^{3/2} r^s_{\max} r^t_{\max}$ follows. If instead $\sigma = \sigma_{\max}$,

$$\begin{aligned}\sigma(\mu_i) &= \frac{1}{\|\Sigma\|_{\infty,1}} \max_{s\in S} \frac{r^s(\mu_i)}{\|r^s\|_1} \ge \frac{1}{\|\Sigma\|_{\infty,1} r^s_{\max}} \max_{s\in S} r^s(\mu_i) \\ &\ge \frac{1}{l\|\Sigma\|_{\infty,1} r^s_{\max}} \sum_{s\in S} r^s(\mu_i) = \frac{1}{l\|\Sigma\|_{\infty,1} r^s_{\max}} \|a_i\| \sigma(\mu_i),\end{aligned} \tag{A.39}$$

which holds similar to (A.37). Combining with (A.38) gives $\|X_i\|_2 \le l^{3/2} \|\Sigma\|_{\infty,1} r^s_{\max} r^t_{\max}$. □

Lemma A.5. If $\sigma = \sigma_{\text{avg}}$, then $m_2(X_i) \le l^2 r^s_{\max} r^t_{\max} \|\Pi(S,T)\|_F$; if instead $\sigma = \sigma_{\max}$, then $m_2(X_i) \le l\|\Sigma\|_{\infty,1} r^s_{\max} r^t_{\max} \max\{\|\Pi(S,T)\|_\infty, \|\Pi(S,T)\|_1\}$.

Proof. We first assume $\sigma = \sigma_{\text{avg}}$. Using Jensen's inequality, and since $X_i = a_i b_i^\mathsf{T}$, we have $\|\mathbb{E}[X_i X_i^\mathsf{T}]\|_2 \le \mathbb{E}[\|X_i X_i^\mathsf{T}\|_2] = \mathbb{E}[\|a_i\|_2^2 \|b_i\|_2^2]$; similarly, $\|\mathbb{E}[X_i X_i^\mathsf{T}]\|_2 \le \mathbb{E}[\|a_i\|_2^2 \|b_i\|_2^2]$. Thus,

$$\begin{aligned} m_2(X_i) &\le \mathbb{E}[\|a_i\|_2^2 \|b_i\|_2^2] = \sum_{u,v\in V} \sigma(u)\pi_u(v) \left(\frac{1}{\sigma(u)^2} \sum_{s\in S} r^s(u)^2\right) \left(\sum_{t\in T} r^t(v)^2\right) \\ &\le r^t_{\max} \sum_{u,v\in V} \frac{\pi_u(v)}{\sigma(u)} \left(\sum_{s\in S} r^s(u)\right)^2 \sum_{t\in T} r^t(v) \\ &\le l r^s_{\max} r^t_{\max} \sum_{s\in S} \sum_{t\in T} \sum_{u,v\in V} r^s(u)\pi_u(v) r^t(v) \le l r^s_{\max} r^t_{\max} \sum_{s\in S} \sum_{t\in T} \pi_s(t), \end{aligned} \tag{A.40}$$

where the second inequality uses the terminating condition of Algorithm 2.2 ($r^t(v) \le r^t_{\max}$) and the nonnegativity of $r^s(u)$, the third follows from (A.37), and the fourth uses the invariant (2.3). Finally, letting $vec(\Pi(S,T))$ denote the l^2-length vector with entries $\{\pi_s(t)\}_{s\in S, t\in T}$,

$$\sum_{s\in S} \sum_{t\in T} \pi_s(t) = \|vec(\Pi(S,T))\|_1 \le l\|vec(\Pi(S,T))\|_2 = l\|\Pi(S,T)\|_F,$$

where the inequality is a standard norm inequality, and the second inequality is by definition of Frobenius norm. Substituting into (A.40) establishes the result.

We next assume $\sigma = \sigma_{\max}$ and bound $\|\mathbb{E}[X_i X_i^\mathsf{T}]\|_2$. We observe that by definition,

$$\begin{aligned} X_i X_i^\mathsf{T} &= \frac{\left(\sum_{t\in T} r^t(\nu_i)^2\right)}{\sigma(\mu_i)^2} \begin{bmatrix} r^{s_1}(\mu_i) & \cdots & r^{s_l}(\mu_i) \end{bmatrix}^\mathsf{T} \begin{bmatrix} r^{s_1}(\mu_i) & \cdots & r^{s_l}(\mu_i) \end{bmatrix} \\ \Rightarrow \mathbb{E}[X_i X_i^\mathsf{T}] &= \sum_{u,v\in V} \frac{\pi_u(v)}{\sigma(u)} \sum_{t\in T} r^t(v)^2 \begin{bmatrix} r^{s_1}(u) & \cdots & r^{s_l}(u) \end{bmatrix}^\mathsf{T} \begin{bmatrix} r^{s_1}(u) & \cdots & r^{s_l}(u) \end{bmatrix}. \end{aligned}$$

Letting 1_l denote the all ones vector of length l, we also have

$$\mathbb{E}[X_i X_i^\mathsf{T}] 1_l = \sum_{u,v \in V} \frac{\pi_u(v)}{\sigma(u)} \sum_{t \in T} r^t(v)^2 \sum_{s \in S} r^s(u) \begin{bmatrix} r^{s_1}(u) & \cdots & r^{s_l}(u) \end{bmatrix}^\mathsf{T}. \tag{A.41}$$

Now since $\mathbb{E}[X_i X_i^\mathsf{T}]$ is symmetric, its 2-norm is its largest eigenvalue; since it is nonnegative, this eigenvalue is bounded by its maximum row sum (Perron-Frobenius Theorem). Thus,

$$\begin{aligned}
\|\mathbb{E}[X_i X_i^\mathsf{T}]\|_2 &\leq \max_{j \in \{1,2,\ldots,l\}} \sum_{u,v \in V} \frac{\pi_u(v)}{\sigma(u)} \sum_{t \in T} r^t(v)^2 \sum_{s \in S} r^s(u) r^{s_j}(u) && \text{(A.42)} \\
&\leq l \|\Sigma\|_{\infty,1} r^s_{\max} r^t_{\max} \max_{j \in \{1,2,\ldots,l\}} \sum_{t \in T} \sum_{u,v \in V} r^{s_j}(u) \pi_u(v) r^t(v) && \text{(A.43)} \\
&\leq l \|\Sigma\|_{\infty,1} r^s_{\max} r^t_{\max} \max_{j \in \{1,2,\ldots,l\}} \sum_{t \in T} \pi_{s_j}(t) = l \|\Sigma\|_{\infty,1} r^s_{\max} r^t_{\max} \|\Pi(S,T)\|_\infty, && \text{(A.44)}
\end{aligned}$$

where (A.42) uses the row sums derived in (A.41), (A.43) uses (A.39) from the proof of Lemma A.4 and the terminating condition of Algorithm 2.2 ($\|r^t\|_\infty \leq r^t_{\max}$), and (A.44) uses the invariant (2.3). We can use the same idea to bound $\|\mathbb{E}[X_i^\mathsf{T} X_i]\|_2$. The steps to obtain the expression analogous to (A.42) follow the same approach so we omit them. We then have

$$\begin{aligned}
\|\mathbb{E}[X_i^\mathsf{T} X_i]\|_2 &\leq \max_{j \in \{1,2,\ldots,l\}} \sum_{u,v \in V} \frac{\pi_u(v)}{\sigma(u)} \sum_{s \in S} r^s(u)^2 \sum_{t \in T} r^t(v) r^{t_j}(v) \\
&\leq \sum_{u,v \in V} \frac{\pi_u(v)}{\sigma(u)} \sum_{s \in S} r^s(u) \max_{s' \in S} r^{s'}(u) \sum_{t \in T} r^t(v) r^{t_j}(v) \\
&\leq l \|\Sigma\|_{\infty,1} r^s_{\max} r^t_{\max} \max_{j \in \{1,2,\ldots,l\}} \sum_{s \in S} \sum_{u,v \in V} r^s(u) \pi_u(v) r^{t_j}(v) && \text{(A.45)} \\
&\leq l \|\Sigma\|_{\infty,1} r^s_{\max} r^t_{\max} \max_{j \in \{1,2,\ldots,l\}} \sum_{s \in S} \pi_s(t_j) = l \|\Sigma\|_{\infty,1} r^s_{\max} r^t_{\max} \|\Pi(S,T)\|_1, && \text{(A.46)}
\end{aligned}$$

where (A.45) uses (A.39) from the proof of Lemma A.4 and the terminating condition of Algorithm 2.2 ($\|r^t\|_\infty \leq r^t_{\max}$), and (A.46) uses (2.3). Thus, by (A.44) and (A.46),

$$\max\{\|\mathbb{E}[X_i^\mathsf{T} X_i]\|_2, \|\mathbb{E}[X_i X_i^\mathsf{T}]\|_2\} \leq l \|\Sigma\|_{\infty,1} r^s_{\max} r^t_{\max} \max\{\|\Pi(S,T)\|_\infty, \|\Pi(S,T)\|_1\}. \quad \square$$

A.7 Choosing order of targets in Algorithm 2.4

As mentioned at the end of Section 2.4.1.2, the performance of Algorithm 2.4 can significantly depend on the order in which the targets $t_1, t_2, \ldots, t_{|T|}$ are chosen. For instance, suppose there exists $t^* \in T$ such that $\pi_{t^*}(t') > r^t_{\max}\ \forall\ t' \in T$, but $\pi_t(t') \leq r^t_{\max}\ \forall\ t \in T \setminus \{t^*\}, t' \in T$. Then choosing $t_1 = t^*$ implies $c_T = |T| - 1$, while choosing $t_{|T|} = t^*$ implies $c_T = 0$. More generally, the algorithm is most efficient when any t satisfying $\pi_t(t') > r^t_{\max}$ for many $t' \in T$ is chosen "early" in the algorithm, i.e. $t_i = t$ for small i. However, because $\pi_t(t')$ is unknown, optimizing the order $t_1, t_2, \ldots, t_{|T|}$ at runtime is difficult. A possible workaround is to use $p^{t'}(t)$ as a proxy for $\pi_t(t')$, since $p^{t'}(t) \in [\pi_t(t') - r^t_{\max}, \pi_t(t')]$ by the invariant (2.2).

Table A.1: Datasets for real graph experiments.

Dataset	Description	n	m
`com-Amazon`	Amazon co-purchasing	334863	925872
`com-dblp`	Scientific co-authorship	317080	1049866
`roadNet-PA`	Roads in Pennsylvania	1087532	1541514
`Slashdot`	Friendships on technology news site	71307	912381
`web-BerkStan`	berkley.edu, stanford.edu web graph	334857	4523232
`web-Google`	Partial web crawl	434818	3419124
`Wiki-Talk`	Friendships among Wikipedia editors	111881	1477893

Unfortunately, even this proxy is difficult to utilize at runtime. This is because we would like to choose t_i such that $\pi_{t_j}(t_i)$ is large for many $j < i$, but the proxy $p^{t_i}(t_j)$ of $\pi_{t_j}(t_i)$ is only known *after* choosing t_i. (Loosely speaking, we have a "chicken and egg" scenario.) Hence, we do not suspect there is a provably optimal method, or even a simple heuristic but suboptimal method, for choosing the order of targets at runtime.

A.8 Experimental details

Here we provide some details on the experiments from Section 2.5.

Datasets: `Direct-ER` is a directed Erdős-Rényi graph with parameters $n = 2000, p = 0.005$ (edge $v \to u$ is present with probability p, independent of other edges, $\forall\ v, u \in V, v \neq u$). `Direct-SBM` is a directed stochastic block model; there are $n = 2000$ nodes partitioned into $k = 20$ disjoint communities, each of size $\frac{n}{k} = 100$; directed edges occur with probability $9/(\frac{n}{k} - 1)$ between distinct nodes in the same community and with probability $1/(n - \frac{n}{k})$ between nodes in different communities (so that each node has nine neighbors in its own community and one neighbor in another community, in expectation, yielding a highly modular graph). The real graphs used are available from the Stanford Network Analysis Platform (SNAP) [43]; see Table A.1 for further details.

Parameters: For the scalar estimation experiments in Sections 2.5.1.1 and 2.5.2.1, we use the algorithmic parameters shown in Table A.2. More specifically, `FW-BW-MCMC` uses Algorithm A.1 for forward DP with parameter $\tilde{r}^s_{\max}$ and samples $w\|\tilde{r}^s\|_1$ random walk starting node locations for each source s (as in Algorithm A.2), uses the walk sharing scheme from Section 2.4.1.1 to sample walks jointly across S, and uses Algorithm 2.4 with parameter $r^t_{\max}$ for the targets; for `Bidirectional-PPR`, we sample w walks separately for each source and run Algorithm 2.2 separately for each target. In practice, we find that w given by the accuracy guarantee (Theorem A.1) is overly pessimistic, so we instead set $w = \frac{cr^t_{\max}}{\delta}$ for both methods, with c given in the table. For the matrix experiments in Sections 2.5.1.2 and 2.5.2.2, we use the same $\tilde{r}^s_{\max}$ and $r^t_{\max}$ values. Furthermore, we set $w = l\frac{cr^t_{\max}}{\delta}$, $w = \|\Sigma\|_{\infty,1}\frac{cr^t_{\max}}{\delta}$, and $w = \sqrt{l\ \mathrm{srank}(P_T(S,:) + P_S^\intercal R_T)}\frac{cr^t_{\max}}{\delta}$ for the baseline, $\sigma_{\max}$, and σ_{avg} schemes, respectively.

Single pair performance: The parameters in Table A.2 were chosen so the primitives `FW-BW-MCMC- Practical` and `Bidirectional-PPR` offer similar accuracy in the single pair case and balance runtime between dynamic programming (DP) and Monte Carlo (MC). To demonstrate this, we show statistics in Table A.2. We obtained the statistics by averaging

Table A.2: Algorithmic parameters and single pair performance.

Graph	Algorithm	$\tilde{r}^s_{\max}$ $\times 10^3$	$r^t_{\max}$ $\times 10^3$	δ	c	DP time (ms)	MC time (ms)	Error
Direct-ER	FW-BW-MCMC-Prac	1.8	3.8	$1/n$	7	10.61	7.63	0.075
Direct-ER	Bidir-PPR	N/A	1.6	$1/n$	12	6.94	7.52	0.072
Direct-SBM	FW-BW-MCMC-Prac	1	4	$1/n$	7	15.43	7.08	0.052
Direct-SBM	Bidir-PPR	N/A	3	$1/n$	10	10.19	12.01	0.061
com-amazon	FW-BW-MCMC-Prac	3.6	18.2	$10/n$	12	22.55	22.54	0.12
com-amazon	Bidir-PPR	N/A	7.4	$10/n$	13	22.13	22.21	0.11
com-dblp	FW-BW-MCMC-Prac	2.9	14.3	$10/n$	13	20.27	20.31	0.12
com-dblp	Bidir-PPR	N/A	6	$10/n$	15	20.03	19.65	0.11
roadNet-PA	FW-BW-MCMC-Prac	15.1	34.8	$10/n$	6	55.04	56.58	0.11
roadNet-PA	Bidir-PPR	N/A	12.8	$10/n$	6	53.19	55.96	0.10
Slashdot	FW-BW-MCMC-Prac	2	12.2	$10/n$	7	3.08	3.38	0.10
Slashdot	Bidir-PPR	N/A	4.2	$10/n$	17	3.30	4.03	0.11
web-BerkStan	FW-BW-MCMC-Prac	6.9	23	$10/n$	3	11.13	11.02	0.12
web-BerkStan	Bidir-PPR	N/A	11.6	$10/n$	3	8.40	8.42	0.12
web-Google	FW-BW-MCMC-Prac	4.5	17.6	$10/n$	8	23.33	22.83	0.11
web-Google	Bidir-PPR	N/A	6.7	$10/n$	11	26.07	22.29	0.11
WikiTalk	FW-BW-MCMC-Prac	2.3	7.5	$10/n$	8	4.40	3.99	0.11
WikiTalk	Bidir-PPR	N/A	2.9	$10/n$	20	5.84	5.10	0.11

across 10^3 trials of the following procedure. First, we sample $t \in V$ uniformly. Next, we sample a "significant" source s (i.e. s satisfying $\pi_s(t) > \delta$) and an "insignificant" source s' (i.e. s' satisfying $\pi_{s'}(t) < \delta$). Since Theorem A.1 bounds relative and absolute error for significant and insignificant pairs, respectively, we compute relative and absolute error for the $\pi_s(t)$ and $\pi_{s'}(t)$ estimates, respectively. (We do not report absolute error statistics as no insignificant estimate violated the absolute error guarantee.) For real datasets, we cannot compute $\pi_s(t)$ to test error performance; instead, we run Algorithm 2.2 with $r^t_{\max}$ replaced by $\eta = \frac{1}{n}$, denote the output p^t_η, r^t_η, and bound relative error for significant pairs as

$$\frac{|\hat{\pi}_s(t) - \pi(t)|}{\pi_s(t)} \leq \frac{|\hat{\pi}_s(t) - p^t_\eta(s)| + \|r^t_\eta\|_\infty}{p^t_\eta(s)} < \frac{|\hat{\pi}_s(t) - p^t_\eta(s)|}{p^t_\eta(s)} + \frac{1}{10},$$

where we have used $p^t_\eta(s) \in [\pi_s(t) - \|r^t_\eta\|_\infty, \pi_s(t)]$ (which holds by (2.2)), $\|r^t_\eta\|_\infty < \eta = \frac{1}{n}$ (which holds by Algorithm 2.2), and $p^t_\eta(s) \geq \delta = \frac{10}{n}$ (which holds by choice of s, t). In the same manner, we can bound absolute error for insignificant pairs as $|\hat{\pi}_s(t) - \pi(t)| \leq |\hat{\pi}_s(t) - p^t_\eta(s)| + \frac{1}{n}$. (Note we choose significant pairs as those (s, t) satisfying $p^t_\eta(s) \geq \delta$, since then $\pi_s(t) \geq \delta$ by (2.2); similarly, we choose insignificant pairs as those (s', t) satisfying $p^t_\eta(s') < \delta - \eta$, since then $\pi_{s'}(t) < \delta$ by (2.2).)

Additional Erdős-Rényi results: We also ran the first experiment from Section 2.5.1.1 for Erdős-Rényi graphs with $n \in \{4000, 8000\}$, each with edge formation probability $10/n$.

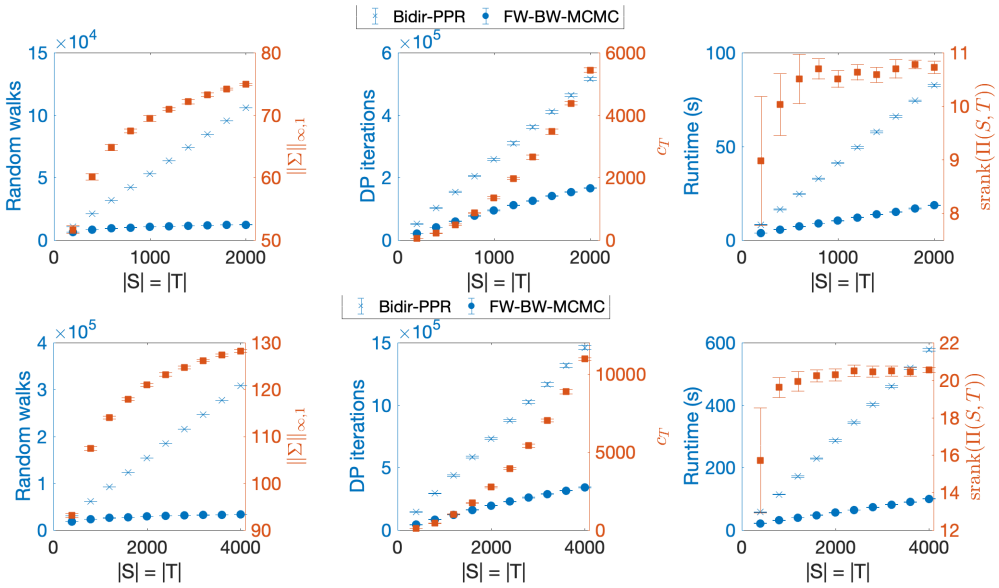


Figure A.1: Replicating Erdős-Rényi experiment from Section 2.5.1.1 with $n = 4000$ (top) and $n = 8000$ (bottom).

For `FW-BW-MCMC`, we used parameters $(\tilde{r}^s_{\max}, r^t_{\max}) = (1.5, 3.5) \times 10^{-3}$ when $n = 4000$ and $(\tilde{r}^s_{\max}, r^t_{\max}) = (1.2, 3.2) \times 10^{-3}$ when $n = 8000$ (choosing smaller parameters for larger n gave more balanced runtime than using the $n = 2000$ parameters from Table A.2). Similarly, for `Bidirectional-PPR`, we used $r^t_{\max} = 1.1 \times 10^{-3}$ when $n = 2000$ and $r^t_{\max} = 0.8 \times 10^{-3}$ when $n = 8000$. As in Table A.2, we ensured these parameters gave similar accuracy for both algorithms. Results are shown in Fig. A.1. As mentioned in Section 2.5.1.1, the plots are qualitatively similar across n; however, they improve slightly as n grows. For instance, in the extreme case $|S| = |T| = n/2$, `FW-BW-MCMC-Prac` was (on average) 2.9, 4.5, and 5.8 times faster than `Bidirectional-PPR` for $n = 2000$, $n = 4000$, and $n = 8000$, respectively.

Building clustered subsets: As mentioned in Section 2.5.2, we use a simple algorithm to randomly construct clustered subsets of nodes for experiments; see Algorithm A.3.

Algorithm A.3: $U =$ `Construct-Clustered-Set`(G, l)

Choose $u \in V$ uniformly at random, let $U = \{u\}$
for $i = 2$ **to** l **do**
  $w \sim (\cup_{u \in U} N_{\text{out}}(u)) \setminus U$ with prob. proportional to $\frac{\sum_{u \in U} 1(w \in N_{\text{out}}(u))}{N_{\text{in}}(w)}$; $U \leftarrow U \cup \{w\}$

A.9 Additional experiments for distributed setting

A.9.1 Matrix approximation, average sampling approach

In this section, we describe a scheme to use the σ_{avg} variant of Algorithm 2.5 in the distributed setting from Section 2.6. Our scheme is quite similar to that defined in Section 2.6 and proceeds as follows. First, we arbitrarily partition S into k subsets of size $|S|/k$, and we use the i-th machine to run forward DP (Algorithm 2.1) for each source s belonging to

the i-th subset. Next, we create another partition $\{S_i\}_{i=1}^k$ of S and use the i-th machine to sample random walks for S_i using the σ_{avg} variant of Algorithm 2.5. Finally, we construct the estimate $\hat{\Pi}(S,T)$ of $\Pi(S,T)$ as in Algorithm 2.5.

It remains to specify the construction of $\{S_i\}_{i=1}^k$. For this, we first use the output p^s of Algorithm 2.1 to define $\text{surr}_s = P_T(s,:) + (p^s)^\mathsf{T} R_T$ for each $s \in S$; here P_T and R_T are the matrices with columns $\{p^t\}_{t\in T}$ and $\{r^t\}_{t\in T}$, respectively (with each (p^t, r^t) computed offline via Algorithm 2.2 as in Section 2.6). Note that surr_s is a row of the surrogate matrix $P_T(S,:) + P_S^\mathsf{T} R_T$ discussed at the conclusion of Section 2.4.2. For $S' \subset S$, we also define $\text{surr}_{S'}$ be the matrix with rows $\{\text{surr}_s\}_{s\in S'}$. Now, as in Section 2.5.2.2, the number of walks sampled on the i-th machine will be set proportional to $\sqrt{|S_i|\text{srank}(\text{surr}_{S_i})}$; hence, our goal is to construct $\{S_i\}_{i=1}^k$ so as to minimize

$$\max_{i\in\{1,\ldots,k\}} \sqrt{|S_i|\text{srank}(\text{surr}_{S_i})}. \tag{A.47}$$

To approximate the solution of this minimization problem, we consider a heuristic method defined in Algorithm A.4. Note this is similar to Algorithm 2.6 in Section 2.6: we assign one source to each S_i (with surr_s vectors far apart), and then iteratively assign the remaining nodes to some S_i so as to minimize the cost of this assignment. In light of (A.47), we here define the cost of assigning s to S_i as $\tilde{d}(s,S_i) = \sqrt{(|S_i|+1)\text{srank}(\text{surr}_{S_i\cup\{s\}})}$.

Algorithm A.4: $\{S_i\}_{i=1}^k = \texttt{Source-Partition-}\sigma_{\text{avg}}\ (\{\text{surr}_s\}_{s\in S},\ k)$

Draw $s \sim S$ uniformly, set $S_1 = \{s\}$; set $S_i = \emptyset\ \forall\ i \in \{2,\ldots,k\}$
for $i = 2$ **to** k **do**
  Let $s \sim S$ with prob. proportional to $\min_{j\in\{1,\ldots,i-1\}} \|\text{surr}_s - \text{surr}_{S_j}\|_1$; set $S_i = \{s\}$
for $i = k+1$ **to** $|S|$ **do**
  Choose $s \in S \setminus (\cup_{j=1}^k S_j)$; compute $\tilde{d}(s,S_j)\ \forall\ j \in \{1,\ldots,k\}$
  Let $j^* \in \arg\min_j \tilde{d}(s,S_j)$, $S_{j^*} = S_{j^*} \cup \{s\}$.

Note Algorithm A.4 requires the singular value decomposition (SVD) of $\text{surr}_{S_j\cup\{s\}}$ to be computed, so that $\tilde{d}(s,S_j)$ can computed in the second for loop of Algorithm A.4. (In contrast, computing $d(s,S_j)$ in the $\sigma_{\max}$ partitioning scheme, Algorithm 2.6, only requires subtracting one vector from another.) Hence, we also propose an alternative partitioning method that avoids this SVD. This method is based on two observations. First, we have

$$\begin{aligned}
\|\text{surr}_{S_j\cup\{s\}}\|_2^2 &= \lambda_{\max}\left(\begin{bmatrix}\text{surr}_{S_j}^\mathsf{T} & \text{surr}_s^\mathsf{T}\end{bmatrix}\begin{bmatrix}\text{surr}_{S_j}\\ \text{surr}_s\end{bmatrix}\right) = \lambda_{\max}\left(\sum_{s'\in S_j}\text{surr}_{s'}^\mathsf{T}\text{surr}_{s'} + \text{surr}_s^\mathsf{T}\text{surr}_s\right)\\
&\le \max_{t\in T}\sum_{t'\in T}\left(\sum_{s'\in S_j}\text{surr}_{s'}^\mathsf{T}\text{surr}_{s'} + \text{surr}_s^\mathsf{T}\text{surr}_s\right)(t,t')\\
&= \max_{t\in T}\left(\sum_{s'\in S_j}\text{surr}_{s'}(t)\|\text{surr}_{s'}\|_1 + \text{surr}_s(t)\|\text{surr}_s\|_1\right),
\end{aligned}$$

where the first equality is a well-known result and the inequality follows from the Perron-Frobenius Theorem. Second, by definition of $\|\cdot\|_F$, we have

$$\|\text{surr}_{S_j \cup \{s\}}\|_F^2 = \sum_{s' \in S_j} \|\text{surr}_{s'}\|_2^2 + \|\text{surr}_s\|_2^2.$$

Combining these observations, we obtain

$$\tilde{d}(s, S_j) \geq \hat{d}(s, S_j) = \sqrt{\frac{(|S_j|+1)\left(\sum_{s' \in S_j} \|\text{surr}_{s'}\|_2^2 + \|\text{surr}_s\|_2^2\right)}{\max_{t \in T}\left(\sum_{s' \in S_j} \text{surr}_{s'}(t)\|\text{surr}_{s'}\|_1 + \text{surr}_s(t)\|\text{surr}_s\|_1\right)}}. \tag{A.48}$$

This expression allows us to estimate $\tilde{d}(s, S_j)$ more efficiently than it can be computed exactly. In Algorithm A.5, we give a partitioning scheme that leverages this insight. Note that the computation of $\hat{d}(s, S_j)$ in Algorithm A.5 can be performed as

$$\hat{d}(s, S_j) = \sqrt{\frac{(|S_j|+1)\,(x_j + \|\text{surr}_s\|_2^2)}{\max_{t \in T}\,(y_j(t) + \text{surr}_s(t)\|\text{surr}_s\|_1)}},$$

i.e. the terms $\sum_{s' \in S_j} \|\text{surr}_{s'}\|_2^2$ and $\sum_{s' \in S_j} \text{surr}_{s'}(t)\|\text{surr}_{s'}\|_1$ in (A.48) have already been computed as x_j and $y_j(t)$ when $\hat{d}(s, S_j)$ is computed; further, x_j and $y_j(t)$ are updated (rather than being computed in full) each time some s is added to S_j (last line of Algorithm A.5).

Algorithm A.5: $\{S_i\}_{i=1}^k$ = `Source-Partition-`σ_{avg}`-alt` $(\{\text{surr}_s\}_{s \in S}, k)$

Draw $s \sim S$ uniformly, set $S_1 = \{s\}$, $x_1 = \|\text{surr}_s\|_2^2$, $y_1(t) = \text{surr}_s(t)\|\text{surr}_s\|_1 \;\forall\; t \in T$
Set $S_i = \emptyset, x_i = y_i = 0 \;\forall\; i \in \{2, \ldots, k\}$
for $i = 2$ **to** k **do**
    Draw $s \sim S$ with probability proportional to $\min_{j \in \{1,\ldots,i-1\}} \|\text{surr}_s - \text{surr}_{S_j}\|_1$
    Set $S_i = \{s\}$, $x_i = \|\text{surr}_s\|_2^2$, $y_i(t) = \text{surr}_s(t)\|\text{surr}_s\|_1 \;\forall\; t \in T$
for $i = k+1$ **to** $|S|$ **do**
    Choose any $s \in S \setminus (\cup_{j=1}^k S_j)$; compute $\hat{d}(s, S_j) \;\forall\; j \in \{1, \ldots, k\}$
    Let $j^* \in \arg\min_j \hat{d}(s, S_j)$
    Set $x_{j^*} = x_{j^*} + \|\text{surr}_s\|_2^2$, $y_{j^*}(t) = y_{j^*}(t) + \text{surr}_s(t)\|\text{surr}_s\|_1 \;\forall\; t \in T$, $S_{j^*} = S_{j^*} \cup \{s\}$

In Fig. A.2, we present empirical results for the σ_{avg} matrix approximation scheme in the distributed setting. In particular, we show results for the scheme described above with the partition $\{S_i\}_{i=1}^k$ constructed via Algorithm A.4 ("Heuristic" in Fig. A.2) and via Algorithm A.5 ("Alt Heuristic" in Fig. A.2). For both schemes, we show the maximum forward DP and random walk sampling time across machines, the maximum number of walks sampled across machines, and the value of the objective function (A.47). The first two quantities are shown relative to the respective quantities for a baseline scheme, which arbitrarily partitions S into subsets of size $|S|/k$ and uses the i-th machine to run the baseline matrix approximation scheme from Section 2.5.2.2 for the i-th subset (recall no forward DP is used for this baseline

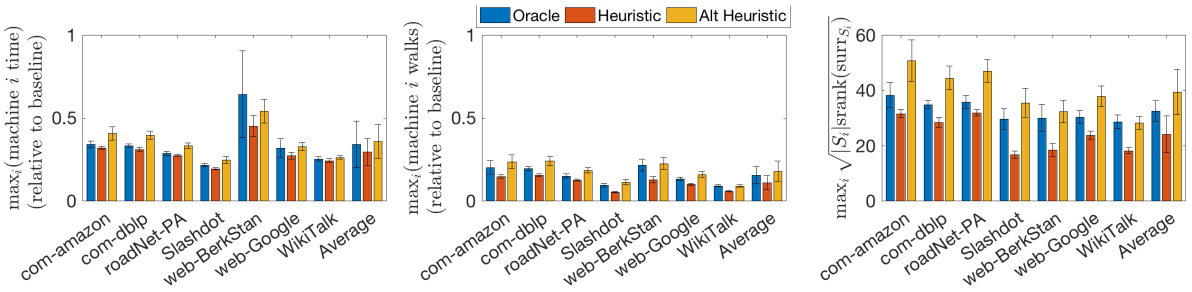


Figure A.2: The σ_{avg} matrix approximation scheme is typically 2-3 times faster than the baseline scheme in the distributed setting of Section 2.6, and our heuristic partitioning schemes (Algorithms A.4 and A.5) perform similar to the oracle method.

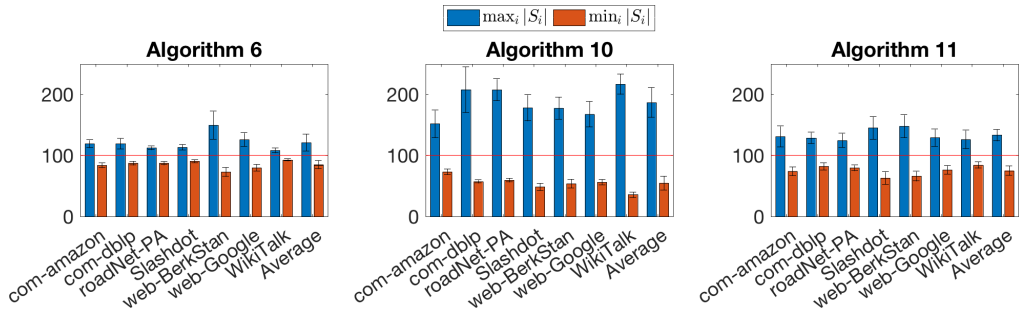


Figure A.3: Our source partitioning schemes produce partitions $\{S_i\}_{i=1}^k$ with $|S_i| \approx S|/k = 100 \ \forall \ i$ (where $|S|/k = 100$ is the case of perfectly balanced partition).

scheme, i.e. walks are not shared across sources). For this experiment, we let $S = \{\tilde{S}_i\}_{i=1}^k$, where $k = 10$ and each $\tilde{S}_i$ is a clustered subset satisfying $|\tilde{S}_i| = 100$; we also compare to an oracle scheme that sets $S_i = \tilde{S}_i$ (as in Section 2.6). In general, Fig. A.2 conveys the same message as Fig. 2.10 in Section 2.6: our methods perform similarly to the oracle method and noticeably outperform the baseline. Here we also note that the heuristic outperforms the oracle across graphs, while the oracle in turn outperforms the alternative heuristic. Nevertheless, the alternative heuristic offers similar performance as the other schemes, while avoiding the SVD computation of the heuristic (which may be prohibitive as S grows).

A.9.2 Other results for source partitioning schemes

As discussed at the conclusion of Section 2.6, it is crucial that our source partitioning schemes (Algorithms 2.6, A.4, and A.5) balance the number of sources assigned to each machine. We find this occurs in practice, despite the lack of explicit balance constraints in Algorithms 2.6, A.4, and A.5. To demonstrate this, we show the maximum and minimum number of sources assigned to machines for the three partitioning schemes in Fig. A.3. Averaged across graphs, Algorithms 2.6, A.4 and A.5 typically produce partitions with $|S_i| \in [85, 122]$, $|S_i| \in [55, 188]$, and $|S_i| \in [75, 134]$, respectively (the red line shows $|S|/k = 100$, i.e. a perfectly balanced partition). We also note that, while Algorithm A.4 typically produces the least balanced partition, its overall performance is similar to that for Algorithm A.5 (see Fig. A.2), which we have argued is more useful in practice for large S.

APPENDIX B

Proofs and Experimental Details for Chapter III

B.1 Proof of Lemma 3.1 (outline)

In this appendix, we outline the proof of Lemma 3.1. Our approach follows the outline described in Section 3.4.3. Specifically, we consider Steps 1-4 of the outline in Appendices B.1.1-B.1.4, respectively. In Appendix B.1.5, we combine the results to prove the lemma.

B.1.1 Error bound in local neighborhood

Our first goal is to bound the error term

$$\left\| \pi_s - \left(\alpha_n e_s^{\mathsf{T}} + \sum_{k \in K_n} \beta_s(k) \pi_k \right) \right\|_1 \tag{B.1}$$

for a particular choice of $\{\beta_s(k)\}_{k \in K_n}$. For this, we require an intermediate result; namely, (3.5) from Section 3.4.3, which we formalize as Lemma B.1 here. Recall from Section 3.4.3 that $\tilde{\pi}_s$ is the stationary distribution of the Markov chain with transition matrix $\tilde{P}_s = (1 - \alpha_n)\tilde{P} + (\alpha_n e_{V_n \setminus K_n} + e_{K_n}) e_s^{\mathsf{T}}$, where $\tilde{P}$ satisfies $\tilde{P}(i,j) = U_i P(i,j)$.

As mentioned in Section 3.4, Lemma B.1 is an alternate form of the Hubs Theorem from [16]. Conceptually, both formulations view $\pi_s(v)$ as the probability of paths from s to v and partition these paths into those that avoid K_n (which have probability proportional to $\tilde{\pi}_s(v)$) and those through K_n (which have probability proportional to $\tilde{\pi}_s(k)\pi_k(v)$). The difference between the formulations is that we explicitly construct a new Markov chain that does not include paths through K_n (i.e. the chain with transition matrix $\tilde{P}_s$), while [16] does not. Our formulation admits an intuitive probabilistic proof; the proof in [16] is linear algebraic.

Lemma B.1. If $U_s = 1$, we have for any realization of the DCM,

$$\pi_s(v) = \frac{\alpha_n U_v \tilde{\pi}_s(v) + \sum_{k \in K_n} \tilde{\pi}_s(k) \pi_k(v)}{\alpha_n + (1 - \alpha_n)\tilde{\pi}_s(K_n)} \ \forall \ v \in V_n.$$

Proof. See Appendix B.2.1. □

We next bound the error term (B.1) using a particular $\{\beta_s(k)\}_{k \in K_n}$: that suggested by Lemma B.1. Our bound leverages the fact that the transition matrix $\tilde{P}_s$ is the sum of two

matrices, one of which is rank one. This allows us to use the Sherman-Morrison-Woodbury formula (see e.g. Section 6.4 of [125]) to bound the error term in terms of the (row) vector

$$\mu_s^{(m)} = e_s^\mathsf{T} \sum_{j=0}^{m} (1-\alpha_n)^j \tilde{P}^j, \tag{B.2}$$

which clearly depends only on the m step neighborhood out of s.

Lemma B.2. Consider any realization of the DCM and assume $U_s = 1$. Define

$$\beta_s(k) = \frac{\tilde{\pi}_s(k)}{\alpha_n + (1-\alpha_n)\tilde{\pi}_s(K_n)} \;\forall\; k \in K_n.$$

Then for each $m \in \mathbb{N}$,

$$\left\| \pi_s - \left(\alpha_n e_s^\mathsf{T} + \sum_{k \in K_n} \beta_s(k) \pi_k \right) \right\|_1 \leq \alpha_n \left(\mu_s^{(m-1)}(V_n \setminus K_n) - 1 \right) + e_s^\mathsf{T} (1-\alpha_n)^m \tilde{P}^m e_{V_n \setminus K_n}.$$

Proof. See Appendix B.2.2. □

B.1.2 Coupling with branching process (Step 2)

Next, we show that the error bound in Lemma B.2 follows the same distribution as a related quantity defined in terms of a branching process. Before presenting this result, we formally define the DCM construction and the branching process.

We begin with the DCM. As described in Section 3.2.1, the basic idea is to randomly pair outgoing half-edges (which we call *outstubs*) with incoming half-edges (which we call *instubs*) in a breadth-first search fashion. We begin by sampling a node s uniformly at random from V_n. In the first iteration, for each outstub belonging to s, we sample an instub uniformly (resampling if the sampled instub has already been paired), and we pair the outstub and instub. We allow the possibility that the sampled instub belongs to s (in which case a self-loop is formed) or that multiple outstubs of s are paired with instubs belonging to the same node (in which case multiple edges are formed between s and that node).[1]

At the end of the first iteration, we denote by A_1 the subset of $V_n \setminus \{s\}$ containing those nodes that have had at least one instub paired with an outstub of s. In the second iteration, we pair all outstubs of all nodes in A_1 in the manner described previously. In general, we pair all outstubs of all nodes in A_{m-1} during the m-th iteration, where A_{m-1} is the set of nodes v at distance $m-1$ from s. In other words, paths out of s of length m are constructed during the m-th iteration. When all outstubs have been paired, the construction finishes.

To facilitate this construction and the coupling argument, we define labels for each instub e and for each node v, denoted $g(e)$ and $g(v)$. The instub label $g(e)$ is necessary because if e

[1]Because of this, the resulting graph will in general be a multi-graph. We note the authors of [18] prove that a simple graph (no self-loops or multi-edges) results with positive probability as $n \to \infty$; however, this requires stronger assumptions on the degree sequence than Assumption 3.1, which is all that we require.

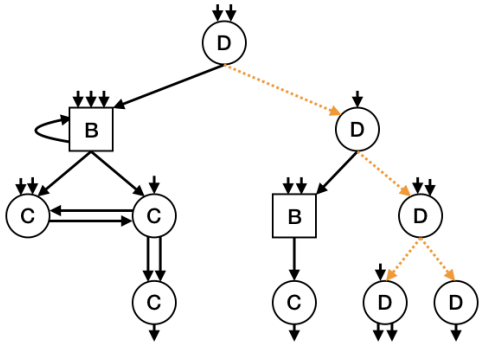


Figure B.1: Example DCM after three steps; $\mu_s^{(3)}(V_n \setminus K_n)$ depends only on dashed subgraph.

is sampled for pairing, we must check whether e has already been paired. Hence, we define

$$g(e) = \begin{cases} 1, & e \text{ is currently unpaired} \\ 0, & e \text{ is currently paired} \end{cases}.$$

The node label $g(v)$ is defined as

$$g(v) = \begin{cases} A, & v \text{ does not currently belong to graph} \\ B, & v \text{ belongs to graph, } U_v = 0 \\ C, & v \text{ belongs to graph, } U_v = 1, \text{ all paths from } s \text{ to } v \\ & \quad \text{visit some } w \in V_n \text{ s.t. } U_w = 0 \\ D, & v \text{ belongs to graph, } U_v = 1, \text{ some path from } s \text{ to } v \\ & \quad \text{avoids all } w \in V_n \text{ s.t. } U_w = 0 \end{cases}. \tag{B.3}$$

To illustrate these node labels, we show a graph after three iterations of the construction in Figure B.1. The node at the top of the figure is s. Circle and square nodes, respectively, depict those nodes v with $U_v = 1$ and $U_v = 0$, respectively (i.e., those belonging to $V_n \setminus K_n$ and K_n, respectively). Short arrows depict half-edges (i.e. unpaired instubs and outstubs), while longer arrows depict edges (i.e. instubs and outstubs that have been paired). Node labels, assigned according to (B.3), are displayed on each node.

Node labels will be useful in the coupling argument to come. In particular, the term $\mu_s^{(m)}(V_n \setminus K_n)$ in Lemma B.2 only depends on the subgraph containing label D nodes within m steps of s (see Figure B.1). This observation follows since $\mu_s^{(m)}(v)$ (by definition) is nonzero if and only if there exists a path from s to v of length at most m that avoids K_n.

The graph construction is defined in Algorithm B.1. We use three additional pieces of notation: I_n is the set of all instubs, $\{(v', j)\}_{j=1}^{D_{v'}}$ is the set of outstubs belonging to $v' \in V_n$ (ordered arbitrarily), and τ_G is a variable that tracks the first iteration at which certain events occur (these events relate to the coupling and will be discussed shortly). Before proceeding, we offer several comments to relate Algorithm B.1 to the preceding discussion:

- In Line 1-2, we initialize the algorithm. We sample the first node s, define the label $g(s)$ according to (B.3), and set $A_0 = \{s\}$ (i.e. the only node at distance zero from s is s itself). We then set $g(e) = 1$ for all instubs e (since no instubs have been paired) and

Algorithm B.1: Graph Construction

```
1  Choose s from V_n uniformly, set g(s) = D if U_s = 1 and g(s) = B if U_s = 0, set
     A_0 = {s}
2  Set g(e) = 1 ∀ e ∈ I_n, set g(v) = A ∀ v ∈ V_n \ {s}, set τ_G = ∞
3  for m = 1 to ∞ do
4      Set A_m = ∅
5      for v' ∈ A_{m-1} do
6          for j = 1 to D_{v'} do
7              // find instub for pairing
8              Uniformly sample instub e
9              if g(e) = 0, τ_G = ∞ then set τ_G = m
10             while g(e) = 0 do
11                 Uniformly sample instub e
12             Pair (v', j) with e, set g(e) = 0, denote instub node by v
13             if g(v) = A then set A_m = A_m ∪ {v}
14             if g(v') = D, g(v) ∈ {C, D}, τ_G = ∞ then set τ_G = m
15             // update label
16             if U_v = 0, g(v) = A then set g(v) = B
17             else if U_v = 1, g(v') = B, g(v) = A then set g(v) = C
18             else if U_v = 1, g(v') ∈ {C, D}, g(v) = A then set g(v) = g(v')
19             else if g(v') = D, g(v) = C then set g(v) = D, set g(w) = D ∀ w ∈ V_n
                 s.t. g(w) = C and v → w path avoiding all w' ∈ V_n s.t. U_{w'} = 0 exists
20             // termination
21             if g(e') = 0 ∀ e' ∈ I_n then return
```

$g(v) = A \ \forall \ v \neq s$ (since only s belongs to the graph at this stage of the algorithm).

- The remainder of the algorithm iterates over m (outer `for` loop), iterates over nodes v' at distance $m-1$ from s (middle `for` loop), and iterates over outstubs belonging to v' (inner `for` loop). For each such outstub, denoted (v', j), the occurs:
  - In Lines 8-11, we uniformly sample an instub e, resampling until an unpaired instub is found. (Line 9 relates to the coupling and will be discussed shortly.)
  - After sampling an unpaired instub e, we pair (v', j) with e and set $g(e) = 0$ to reflect the fact that e has been paired (Line 12). If the node v to which e belongs did not previously belong to the graph (i.e. if $g(v) = A$), then v is at distance m from s, so we add v to A_m (Line 13). (Line 14 relates to the coupling.)
  - In Lines 16-19, we update the label of v according to (B.3). Note that, if $g(v') = D$ and $g(v) = C$, (B.3) implies that a path from s to v avoiding K_n did not exist before (v', j) and e were paired, but now such a path does exist. Hence, if some node w s.t. $g(w) = C$ can be reached from v while avoiding K_n, a path from s to w avoiding K_n now exists as well. For this reason, we must change the label of such w from C to D (Line 19).
- If all instubs have been paired, the algorithm terminates (Line 21).

Our next goal is to define a branching process and a quantity related to the error bound in

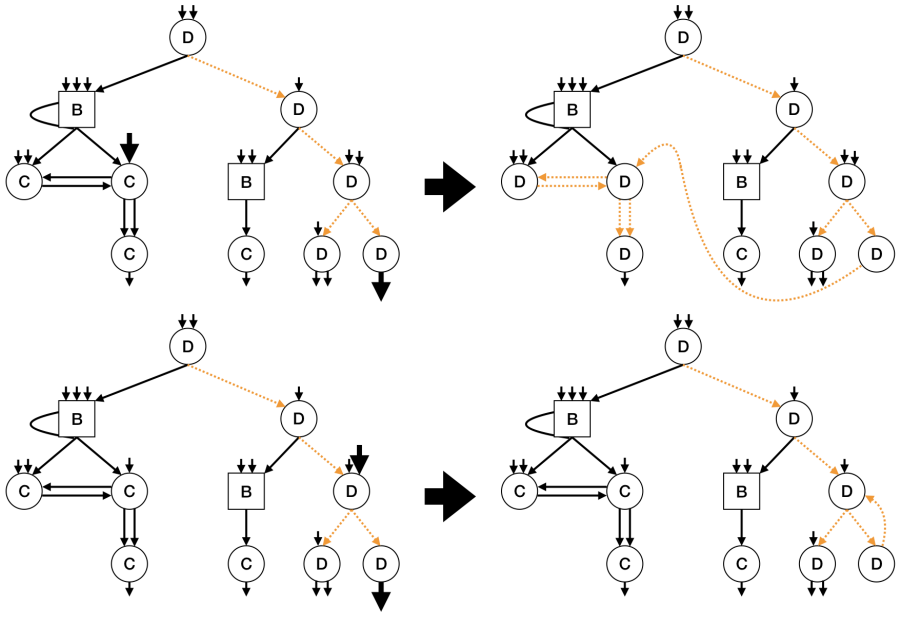


Figure B.2: Instub belonging to label C node (top) or label D node (bottom) is sampled for pairing with outstub of label D node (left of arrow). After labels are updated, orange dashed subgraph of label D nodes is no longer a tree (right of label).

Lemma B.2, so that this error bound can instead be analyzed on the tree resulting from the branching process. Before defining this tree construction, we offer some intuition, which also helps explain the variable τ_G in Algorithm B.1. First, recall the error bound in Lemma B.2 depends only on the m-step neighborhood out of s. Hence, a typical approach to analyzing the bound would be to argue that this neighborhood is treelike, and then to analyze the bound on a related tree. However, this is more than we require. To see this, we return to the example from Figure B.1. As argued previously, the error bound only depends on the orange dashed subgraph. Hence, the related tree we construct will (roughly speaking) only contain this subgraph, i.e. rather than require the entire m-step neighborhood to be treelike, we only require the m-step neighborhood of label D nodes to be treelike.

This discussion also helps explain the variable τ_G in Algorithm B.1. Note that we set $\tau_G = m$ if we pair an outstub of $v' \in A_{m-1}$ with an instub of v, where $g(v') = D$ and $g(v) \in \{C, D\}$ (Line 14 in Algorithm B.1). As shown in Figure B.2, these events (may) destroy the tree structure of the label D subgraph. We also set $\tau_G = m$ if we sample an instub that has already been paired while attempting to pair an outstub of $v' \in A_{m-1}$ (Line 9). This is to ensure nodes have *i.i.d.* attributes (N_v, D_v, U_v), as will nodes in the tree construction.

This intuition motivates our tree construction. We begin with a root node denoted by ϕ, and we assign attributes (N_ϕ, D_ϕ, U_ϕ). Here N_ϕ is the number of instubs of ϕ, all of which will remain unpaired for the duration of the algorithm (so that the tree structure is maintained); D_ϕ is the number of offspring of ϕ; and $U_\phi = 1$. To each offspring of ϕ, denoted $1, 2, \ldots, D_\phi$, we assign attributes (N_i, D_i, U_i). Here N_i denotes the number of instubs of i; one of these is paired with the i-th outstub of ϕ, while the other $N_i - 1$ remain unpaired

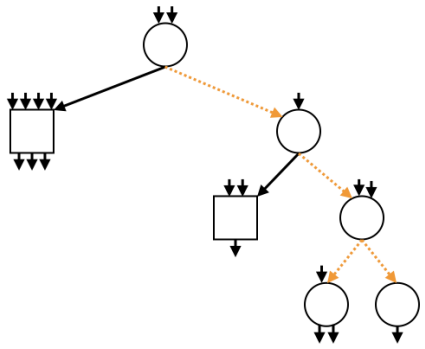


Figure B.3: Branching process after three generations, corresponding to the example graph from Figure B.1. In particular, the orange dashed subgraphs are identical.

(again, to preserve the tree structure). Furthermore, unlike the root node, node i receives D_i offspring *only if* $U_i = 1$; otherwise, the outstubs remain unpaired. This is explained by Figure B.1, since only the orange dashed subgraph affects the quantity of interest.

The set of nodes $1, 2, \ldots, D_\phi$ is denoted by $\hat{A}_1$. In general, we denote by $\hat{A}_m$ the m-th generation of the tree, i.e. the set of nodes at distance m from the root node. The generic node in $\hat{A}_m, m > 1$ is denoted by $\imath$, where $\imath = (i_1, i_2, \ldots, i_m)$ is an ordered list of natural numbers that traces the unique path from ϕ to $\imath$: specifically, this path is $\phi \in \hat{A}_0, i_1 \in \hat{A}_1, (i_1, i_2) \in \hat{A}_2, \ldots, \imath \in \hat{A}_m$. The offspring of $\imath$ (assuming $U_\imath = 1$) are denoted by $\{(\imath, j)\}_{j=1}^{D_\imath}$, where $(\imath, j) = (i_1, i_2, \ldots, i_m, j)$ is the concatenation operation.

To assign attributes, we define $f_n : \mathbb{N} \times \mathbb{N} \times \{0, 1\} \to [0, 1]$ and $f_n^* : \mathbb{N} \times \mathbb{N} \to [0, 1]$ (given the degree sequence) by (B.4). Note that f_n is the distribution of node attributes for nodes sampled proportional to in-degree, whereas f_n^* is the distribution of node attributes for nodes sampled uniformly at random from $V_n \setminus K_n$. Because non-root nodes are sampled proportional to in-degree in the graph construction (until an edge must be resampled, i.e. until we set $\tau_G = m$), non-root node attributes are sampled from f_n in the tree construction. Similarly, since the first node is sampled uniformly from $V_n \setminus K_n$ in the case of interest of the graph construction, root node attributes are sampled from f_n^* in the tree.

$$f_n(i, j, k) = \sum_{h=1}^{n} \frac{N_h}{L_n} 1(N_h = i, D_h = j, U_h = k), f_n^*(i, j) = \sum_{h=1}^{n} \frac{U_h}{\sum_{h'=1}^{n} U_{h'}} 1(N_h = i, D_h = j). \tag{B.4}$$

The tree construction is given formally in Algorithm B.2. We denote by $\hat{G}_n = (\hat{V}_n, \hat{E}_n)$ the resulting tree. Note the tree construction continues indefinitely, so the subscript n does not refer to the number of nodes in the tree; rather, it refers to the length of the sequence $\{N_h, D_h, U_h\}_{h=1}^{n}$ from which the distributions f_n, f_n^* are defined. Finally, in Figure B.3, we show an example of the tree construction, which corresponds to the graph construction of Figures B.1 (i.e. the dashed orange subgraph has the same structure).

Having defined the tree construction, we define the aforementioned quantity that follows the distribution of the error bound in Lemma B.2. Specifically, we define $\hat{\mu}_\phi$ recursively as

$$\hat{\mu}_\phi(\phi) = 1, \quad \hat{\mu}_\phi((\imath, j)) = \hat{\mu}_\phi(\imath) \frac{(1 - \alpha_n) U_\imath}{D_\imath}, (\imath, j) \in \hat{A}_l, l > 0, \tag{B.5}$$

Algorithm B.2: Tree Construction

Draw root attributes $(N_\phi, D_\phi) \sim f_n^*$, set $U_\phi = 1$, set $\hat{A}_0 = \{\phi\}$
for $m = 1$ **to** ∞ **do**
  Set $\hat{A}_m = \emptyset$
  for $\imath \in \hat{A}_{m-1}$ **do**
    if $U_\imath = 1$ **then**
      for $j = 1$ **to** $D_\imath$ **do**
        Add offspring $(\imath, j)$ to $\imath$, let $(N_{(\imath,j)}, D_{(\imath,j)}, U_{(\imath,j)}) \sim f_n$, set $\hat{A}_m = \hat{A}_m \cup \{(\imath, j)\}$

where (by convention), $\imath = \phi$ when $(\imath, j) = i_1 \in \mathbb{N}$, i.e. when $(\imath, j) \in \hat{A}_1$. Note that (B.5) is the same as (B.2) but computed on the tree $\hat{G}_n$; because there is a unique path from ϕ to $\imath$ for each $\imath \in \hat{V}_n$, this recursive definition will be more convenient.

We next state Lemma B.3, whose proof is in Appendix B.2.3. The proof formalizes the preceding intuition, that when $\tau_G > m$, the error bound from Lemma B.2 is computed on a treelike subgraph and therefore follows the distribution of the analogous tree quantity.

Lemma B.3. For any $m \in \mathbb{N}$,

$$\mu_s^{(m)}(V_n \setminus K_n) | \{\tau_G > m, U_s = 1\} \stackrel{\mathcal{D}}{=} \sum_{j=0}^{m} \sum_{\imath \in \hat{A}_j} U_\imath \hat{\mu}_\phi(\imath),$$

where $\stackrel{\mathcal{D}}{=}$ denotes equality in distribution.

Proof. See Appendix B.2.3. □

We can now explain the remainder of our approach to proving the lemma. Using Lemmas B.2 and B.3, and noting that, by definition, $\sum_{\imath \in \hat{A}_0} U_\imath \hat{\mu}_\phi(\imath) = U_\phi \hat{\mu}_\phi(\phi) = 1$, we have

$$\begin{aligned}
\mathbb{P}\left[B_s(K_n, \varepsilon) | U_s = 1\right] &\leq \mathbb{P}\Big[\alpha_n \left(\mu_s^{(m-1)}(V_n \setminus K_n) - 1\right) + e_s^\mathsf{T}(1-\alpha)^m \tilde{P}^m e_{V_n \setminus K_n} \geq \varepsilon \Big| U_s = 1\Big] \\
&\leq \mathbb{P}[\tau_G \leq m | U_s = 1] + \mathbb{P}\left[\alpha_n \sum_{j=1}^{m-1} \sum_{\imath \in \hat{A}_j} U_\imath \hat{\mu}_\phi(\imath) + \sum_{\imath \in \hat{A}_m} U_\imath \hat{\mu}_\phi(\imath) \geq \varepsilon\right] \quad \text{(B.6)}
\end{aligned}$$

Hence, our approach to bounding the probability of (3.2) will be to further bound the two summands in (B.6). Since (B.6) holds for any $m \in \mathbb{N}$, our final step will be to choose m to optimize the sum of these bounds. In particular, we will choose m to balance the two bounds. This is because the summands are increasing and decreasing in m, respectively.

B.1.3 Coupling failure (Step 3)

Our bound for the first summand in (B.6) is given in Lemma B.4. This result is similar to Lemma 5.4 of [48], and our proof follows a similar approach. However, Assumption 3.1 is different than the assumption required for the result in [48]. This difference arises because the result in [48] requires the entire m-step neighborhood to be treelike, while we only

require the m-step neighborhood of label D nodes to be treelike. This allows us to relax the assumption from [48], which requires $\sum_{h=1}^{n} N_h^2/n$ to converge; we only require $\sum_{h=1}^{n} N_h^2 U_h/n$ to converge. In fact, the example degree sequence presented in Section 3.7.3 satisfies

$$\mathbb{E}[N_h^2 U_h] = O(1), \quad \mathbb{E}[N_h^2] = O\left(n^{l_2}\right),$$

where $l_2 > 0$. Hence, there are sequences for which the lemma from [48] does not apply, but for which our version does apply. This is why we do not directly use the lemma from [48].

Lemma B.4. Given Assumption 3.1, for any $m_n \to \infty$ as $n \to \infty$ s.t. $m_n = O(n^\gamma)$, we have

$$\mathbb{P}[\tau_G \leq m_n | U_s = 1] = O\left(n^{-\delta} + \zeta^{m_n}/\sqrt{n}\right),$$

where γ, δ, ζ are defined in Assumption 3.1.

Proof. See Appendix B.2.4. □

B.1.4 Tail bound on branching process quantity (Step 4)

Our final step is to bound the second summand in (B.6). Our approach is to bound the probability that either $\alpha_n \sum_{j=1}^{m-1} \sum_{\imath \in \hat{A}_j} U_\imath \hat{\mu}_\phi(\imath)$ or $\sum_{\imath \in \hat{A}_m} U_\imath \hat{\mu}_\phi(\imath)$ exceeds $\varepsilon/2$. For the first term, the recursive definition of $\hat{\mu}_\phi$ yields a martingale structure that allows us to use an approach similar to the method of bounded differences. The second term arises from the tail of the m-step neighborhood approximation from Appendix B.1.1; hence, its expected value decays geometrically fast in m, so we simply use Markov's inequality.

Lemma B.5. Given Assumption 3.1, for any $\varepsilon > 0$, any $m_n \to \infty$ as $n \to \infty$ s.t. $m_n = O(n^\gamma)$, and any $\alpha_n \to 0$ as $n \to \infty$, we have

$$\mathbb{P}\left[\alpha_n \sum_{j=1}^{m-1} \sum_{\imath \in \hat{A}_j} U_\imath \hat{\mu}_\phi(\imath) + \sum_{\imath \in \hat{A}_m} U_\imath \hat{\mu}_\phi(\imath) \geq \varepsilon\right] = O\left(n^{-\delta} + p^{m_n} + e^{-((1-p)\varepsilon)^2/(2\alpha_n)}\right),$$

where p, δ are defined in Assumption 3.1.

Proof. See Appendix B.2.5. □

B.1.5 Completing the proof of Lemma 3.1

Finally, we can combine the results of this section to prove Lemma 3.1. First, we substitute the results of Lemmas B.4 and B.5 into (B.6) to obtain (when Assumption 3.1 holds)

$$\mathbb{P}\left[B_s(K_n, \varepsilon) | U_s = 1\right] = O\left(n^{-\delta} + \frac{\zeta^{m_n}}{\sqrt{n}} + p^{m_n} + e^{-((1-p)\varepsilon)^2/(2\alpha_n)}\right).$$

Next, choose $m_n = \frac{\log n}{2\log(\zeta/p)}$ to equate the middle two terms, i.e.

$$\frac{\zeta^{m_n}}{\sqrt{n}} = p^{m_n} = n^{-\log(1/p)/(2\log(\zeta/p))}.$$

For the third term, take $\alpha_n = \rho \log(1/\tau) \log \zeta / \log n$ as in Proposition 3.1 to obtain

$$\exp\left(-\frac{((1-p)\varepsilon)^2}{2\alpha_n}\right) = n^{-((1-p)\varepsilon)^2/(2\rho \log(1/\tau)\log\zeta)}.$$

Hence, we ultimately obtain

$$\mathbb{P}\left[B_s(K_n, \varepsilon) | U_s = 1\right] = O\left(n^{-c(\varepsilon)}\right),$$

where $c(\varepsilon)$ is defined as in the statement of the lemma.

B.2 Proof of Lemma 3.1 (details)

B.2.1 Proof of Lemma B.1

The lemma relates the stationary distributions of several Markov chains: those with transition matrices P_s, $\tilde{P}_s$, and $P_k, k \in K_n$, where P_s and P_k are defined in Section 3.2.2 and $\tilde{P}_s$ is defined in (3.4). We will denote these chains by $\{X_i^s\}_{i=0}^{\infty}$, $\{\tilde{X}_i^s\}_{i=0}^{\infty}$, and $\{X_i^k\}_{i=0}^{\infty}$, $k \in K_n$, respectively, in this proof. Our basic approach will be to relate the stationary distributions indirectly via a renewal-reward interpretation of PPR. Hence, we begin by defining this interpretation in Appendix B.2.1.1. We then prove the lemma in Appendix B.2.1.2. Recall from the main text that $\mathbb{P}_{G_n}[\cdot]$ and $\mathbb{E}_{G_n}[\cdot]$ denote probability and expectation with the DCM fixed (as in the statement of the lemma).

B.2.1.1 Renewal-reward interpretation of PPR

From the dynamics of $\{X_i^s\}_{i=0}^{\infty}$ described in Section 3.2.2, we can view the time instances of jumps to s as forming a Bernoulli process with parameter α_n, independent of the random walk. Furthermore, for each $v \in V_n$, we can define a reward function $1(X_i^s = v)$. Then, letting L_s denote the time of the first jump to s, we define

$$\tau_s(v) = \sum_{i=0}^{L_s - 1} 1(X_i^s = v), \tag{B.7}$$

which, when $X_0^s = s$, gives the accumulated reward during the first inter-renewal interval. From the renewal-reward theorem (see, for example, Section 5.4 of [126]), it follows that

$$\lim_{t\to\infty} \frac{1}{t} \sum_{i=0}^{t-1} 1(X_i^s = v) = \alpha_n \mathbb{E}_{G_n}[\tau_s(v) | X_0^s = s], \tag{B.8}$$

where we have also used the fact that $L_s \sim$ geometric(α_n). On the other hand, assuming P_s is irreducible (which we will return to argue is without loss of generality), we have

$$\pi_s(v) = \lim_{t\to\infty} \frac{1}{t} \sum_{i=0}^{t-1} 1(X_i^s = v). \tag{B.9}$$

Hence, combining (B.8) and (B.9) yields

$$\pi_s(v) = \alpha_n \mathbb{E}_{G_n}[\tau_s(v)|X_0^s = s] \ \forall \ v \in V_n. \tag{B.10}$$

Similarly, for $k \in K_n$, $\pi_k(v) = \alpha_n \mathbb{E}_{G_n}[\tau_k(v)|X_0^k = k]$, where $\tau_k(v)$ is defined as in (B.7).

For the chain $\{\tilde{X}_i^s\}_{i=0}^{\infty}$, we have a similar (though subtler) renewal-reward interpretation. Recall the dynamics of this chain are as follows: from $v \in V_n \setminus K_n$, follow the random walk with probability $1 - \alpha_n$ and jump to s with probability α_n; from $k \in K_n$, jump to s with probability 1. Hence, while the time instances of jumps to s do not form a Bernoulli process, they still form a renewal process: inter-renewal intervals are independent (due to the Markov property) and identically-distributed (due to the time invariance of the Markov chain). Also, assuming $\tilde{X}_0^s = s$, the first renewal occurs at $\min\{\tilde{L}_s, \tilde{H}+1\}$, where $\tilde{L}_s \sim \text{geometric}(\alpha_n)$ and $\tilde{H} = \inf\{i \in \mathbb{Z}_+ : \tilde{X}_i^s \in K_n\}$ is the hitting time of K_n. It follows that

$$\tilde{\pi}_s(v) = \frac{\mathbb{E}_{G_n}[\tilde{\tau}_s(v)|\tilde{X}_0^s = s]}{\mathbb{E}_{G_n}[\min\{\tilde{L}_s, \tilde{H}+1\}|\tilde{X}_0^s = s]} \ \forall \ v \in V_n,$$

where $\tilde{\tau}_s(v) = \sum_{i=0}^{\min\{\tilde{L}_s - 1, \tilde{H}\}} 1(\tilde{X}_i^s = v)$.

Before proceeding, we argue irreducibility is without loss of generality for the Markov chains at hand. Consider, for example, $\{X_i^s\}_{i=0}^{\infty}$. If this chain is not irreducible, we can define $V_{n,s} \subset V_n$ as the states for which a path of positive probability from s to v exists. Then the Markov chain restricted to states $V_{n,s}$ is irreducible: for any $v, w \in V_{n,s}$, we can jump from v to s and then reach w from s. We can then compute the stationary distribution $\{\pi_s(v)\}_{v \in V_{n,s}}$ for this irreducible chain and set $\pi_s(v) = 0 \ \forall \ v \in V_n \setminus V_{n,s}$ (intuitively, v is unimportant to s if s cannot reach v, so its PPR should be zero). Note this is consistent with the derivation above. In particular, (B.8) and (B.9) hold for the chain restricted to states $V_{n,s}$, so (B.10) holds for $v \in V_{n,s}$; conversely, both sides of (B.10) are zero for $v \notin V_{n,s}$.

B.2.1.2 Proof of the lemma

Equipped with this renewal-reward interpretation, we will relate π_s, $\tilde{\pi}_s$, and $\pi_k, k \in K_n$ by relating $\mathbb{E}_{G_n}[\tau_s(v)|X_0^s = s]$, $\mathbb{E}_{G_n}[\tilde{\tau}_s(v)|\tilde{X}_0^s = s]$, and $\mathbb{E}_{G_n}[\tau_k(v)|X_0^k = k]$. For this, we define $H = \inf\{i \in \mathbb{Z}_+ : X_i^s \in K_n\}$, the quantity analogous to $\tilde{H}$ instead defined on $\{X_i^s\}_{i=0}^{\infty}$.

Because the dynamics of $\{X_i^s\}_{i=0}^{\infty}$ and $\{\tilde{X}_i^s\}_{i=0}^{\infty}$ only differ when K_n is reached, we can immediately obtain several relationship between the quantities computed on these chains. In particular, if K_n is *not* reached before the first renewal (i.e. if $L_s \leq H$, $\tilde{L}_s \leq \tilde{H}$), the chains have identical dynamics. Therefore, we have $\forall \ v \in V_n$,

$$\mathbb{E}_{G_n}[\tau_s(v)|L_s \leq H, X_0^s = s] = \mathbb{E}_{G_n}[\tilde{\tau}_s(v)|\tilde{L}_s \leq \tilde{H}, \tilde{X}_0^s = s].$$

Furthermore, $\tilde{\tau}_s(v) = 0$ when $v \in K_n$ and $\tilde{L}_s \leq \tilde{H}$ (i.e. when K_n is not reached before the first renewal), so we may rewrite this as

$$\mathbb{E}_{G_n}[\tau_s(v)|L_s \leq H, X_0^s = s] = U_v \mathbb{E}_{G_n}[\tilde{\tau}_s(v)|\tilde{L}_s \leq \tilde{H}, \tilde{X}_0^s = s]. \tag{B.11}$$

By a similar argument, if K_n *is* reached before the first renewal ($L_s > H$, $\tilde{L}_s > \tilde{H}$), the

dynamics of the chains differ after $H, \tilde{H}$, but remain the until $H, \tilde{H}$. Hence, $\forall\ k \in K_n$,

$$\mathbb{P}_{G_n}[X_H^s = k, L_s > H | X_0^s = s] = \mathbb{P}_{G_n}[\tilde{X}_{\tilde{H}}^s = k, \tilde{L}_s > \tilde{H} | \tilde{X}_0^s = s], \tag{B.12}$$

which also implies

$$\mathbb{P}_{G_n}[L_s \leq H | X_0^s = s] = \mathbb{P}_{G_n}[\tilde{L}_s \leq \tilde{H} | \tilde{X}_0^s = s]. \tag{B.13}$$

We can obtain another expression for the right side of (B.12). Since jumps from k to s occur with probability 1 on the $\{\tilde{X}_i^s\}_{i=0}^{\infty}$ chain, k is visited at most one time before the first renewal, i.e. $\tilde{\tau}_s(k) \in \{0, 1\}$. Also, $\tilde{\tau}_s(k) = 1$ if and only if $\tilde{L}_s > \tilde{H}$ and $\tilde{X}_{\tilde{H}}^s = k$. Hence,

$$\mathbb{P}_{G_n}[X_H^s = k, L_s > H | X_0^s = s] = \mathbb{E}_{G_n}[\tilde{\tau}_s(k) | \tilde{X}_0^s = s]\ \forall\ k \in K_n. \tag{B.14}$$

If instead K_n is reached, the dynamics of $\{X_i^s\}_{i=0}^{\infty}$ and $\{\tilde{X}_i^s\}_{i=0}^{\infty}$ differ. In this case, we claim

$$\begin{aligned}\mathbb{E}_{G_n}[\tau_s(v) | X_H^s = k, L_s > H, X_0^s = s] = U_v \mathbb{E}_{G_n}[\tilde{\tau}_s(v) | \tilde{X}_{\tilde{H}}^s = k, \tilde{L}_s > \tilde{H}, \tilde{X}_0^s = s] \\ + \mathbb{E}_{G_n}[\tau_k(v) | X_0^k = k],\end{aligned} \tag{B.15}$$

which we will return to prove shortly. (In essence, (B.15) counts the number visits to v before and after reaching k using the $\{\tilde{X}_i^s\}_{i=0}^{\infty}$ and $\{X_i^k\}_{i=1}^{k}$ chains, respectively.)

By (B.11), (B.12), (B.13), (B.14), and (B.15), and the law of total expectation,

$$\mathbb{E}_{G_n}[\tau_s(v) | X_0^s = s] = U_v \mathbb{E}_{G_n}[\tilde{\tau}_s(v) | \tilde{X}_0^s = s] + \sum_{k \in K_n} \mathbb{E}_{G_n}[\tilde{\tau}_s(k) | \tilde{X}_0^s = s] \mathbb{E}_{G_n}[\tau_k(v) | X_0^k = k].$$

We then use the renewal-reward interpretation from Appendix B.2.1.1 to translate this equation back to stationary distributions. Specifically, multiplying by α_n on both sides, and multiplying and dividing by $\mathbb{E}_{G_n}[\min\{\tilde{L}_s, \tilde{H} + 1\} | \tilde{X}_0^s = s]$ on the right side, gives

$$\pi_s(v) = \mathbb{E}_{G_n}[\min\{\tilde{L}_s, \tilde{H} + 1\} | \tilde{X}_0^s = s] \left(\alpha_n U_v \tilde{\pi}_s(v) + \sum_{k \in K_n} \tilde{\pi}_s(k) \pi_k(v) \right). \tag{B.16}$$

Then, summing over $v \in V_n$ (assuming stationary distributions are normalized to sum to 1),

$$\begin{aligned}1 &= \mathbb{E}_{G_n}[\min\{\tilde{L}_s, \tilde{H} + 1\} | \tilde{X}_0^s = s] \left(\alpha_n \tilde{\pi}_s(V_n \setminus K_n) + \tilde{\pi}_s(K_n) \right) \\ \Rightarrow \mathbb{E}_{G_n}[\min\{\tilde{L}_s, \tilde{H} + 1\} | \tilde{X}_0^s = s] &= \frac{1}{\alpha_n + (1 - \alpha_n) \tilde{\pi}_s(K_n)}.\end{aligned} \tag{B.17}$$

Finally, combining (B.16) and (B.17) completes the proof.

We now return to prove (B.15). First, by definition of $\tau_s(v)$,

$$\begin{aligned}\mathbb{E}_{G_n}[\tau_s(v) | X_H^s = k, L_s > H, X_0^s = s] = \mathbb{E}_{G_n}\left[\sum_{i=0}^{H-1} 1(X_i^s = v) \middle| X_H^s = k, L_s > H, X_0^s = s \right] \\ + \mathbb{E}_{G_n}\left[\sum_{i=H}^{L_s - 1} 1(X_i^s = v) \middle| X_H^s = k, L_s > H, X_0^s = s \right].\end{aligned} \tag{B.18}$$

Now by the preceding arguments, $\{X_i^s\}_{i=0}^{\infty}$, $\{\tilde{X}_i^s\}_{i=0}^{\infty}$ have the same dynamics before $H, \tilde{H}$, so we can replace $H, X_i^s, X_H^s, L_s, X_0^s$ by $\tilde{H}, \tilde{X}_i^s, \tilde{X}_{\tilde{H}}^s, \tilde{L}_s, \tilde{X}_0^s$ in the first summand in (B.18). Moreover, for $v \in V_n \setminus K_n$ (i.e. $U_v = 1$), we can write

$$\begin{aligned}
&\mathbb{E}_{G_n}\left[\sum_{i=0}^{\tilde{H}-1} 1(\tilde{X}_i^s = v)\middle| \tilde{X}_{\tilde{H}}^s = k, \tilde{L}_s > \tilde{H}, \tilde{X}_0^s = s\right] \\
&= \mathbb{E}_{G_n}\left[\sum_{i=0}^{\tilde{H}} 1(\tilde{X}_i^s = v)\middle| \tilde{X}_{\tilde{H}}^s = k, \tilde{L}_s > \tilde{H}, \tilde{X}_0^s = s\right] \\
&= \mathbb{E}_{G_n}\left[\sum_{i=0}^{\min\{\tilde{L}_s - 1, \tilde{H}\}} 1(\tilde{X}_i^s = v)\middle| \tilde{X}_{\tilde{H}}^s = k, \tilde{L}_s > \tilde{H}, \tilde{X}_0^s = s\right] \\
&= \mathbb{E}_{G_n}\left[\tilde{\tau}_s(v)\middle| \tilde{X}_{\tilde{H}}^s = k, \tilde{L}_s > \tilde{H}, \tilde{X}_0^s = s\right],
\end{aligned}$$

where the first equality holds since $v \in V_n \setminus K_n$ and by conditioning on $\{\tilde{X}_{\tilde{H}}^s = k\}$ $(k \in K_n)$, the second holds by conditioning on $\{\tilde{L}_s > \tilde{H}\}$, and the third holds by definition of $\tilde{\tau}_s(v)$. Note also that if $v \in K_n$ (i.e. $U_v = 0$), we simply have

$$\mathbb{E}_{G_n}\left[\sum_{i=0}^{\tilde{H}-1} 1(\tilde{X}_i^s = v)\middle| \tilde{X}_{\tilde{H}}^s = k, \tilde{L}_s > \tilde{H}, \tilde{X}_0^s = s\right] = 0,$$

which holds by definition of $\tilde{H}$. To summarize, we have shown

$$\mathbb{E}_{G_n}\left[\sum_{i=0}^{H-1} 1(X_i^s = v)\middle| X_H^s = k, L_s > H, X_0^s = s\right] = U_v \mathbb{E}_{G_n}\left[\tilde{\tau}_s(v)\middle| \tilde{X}_{\tilde{H}}^s = k, \tilde{L}_s > \tilde{H}, \tilde{X}_0^s = s\right]. \tag{B.19}$$

Next, consider the second summand in (B.18). We rewrite this term as

$$\frac{\mathbb{E}_{G_n}[\sum_{i=H}^{L_s-1} 1(X_i^s = v, X_H^s = k, X_0^s = s) 1(L_s > H)]}{\mathbb{P}_{G_n}[X_H^s = k, L_s > H, X_0^s = s]}, \tag{B.20}$$

and we focus on the numerator. First, we note $1(L_s > H) = \sum_{l>h} 1(L_s = l, H = h)$, where the sum is taken over $\{(l,h) \in \mathbb{Z}_+ \times \mathbb{Z}_+ : l > h\}$. Substituting and using linearity gives

$$\begin{aligned}
&\sum_{l>h} \mathbb{E}_{G_n}\left[\sum_{i=H}^{L_s-1} 1(X_i^s = v, X_H^s = k, X_0^s = s) 1(L_s = l, H = h)\right] \\
&\quad = \sum_{l>h} \mathbb{E}_{G_n}\left[\sum_{i=h}^{l-1} 1(X_i^s = v, X_h^s = k, X_0^s = s) 1(L_s = l, H = h)\right]
\end{aligned}$$

$$= \sum_{l>h} \sum_{i=h}^{l-1} \mathbb{P}_{G_n} \left[X_i^s = v, X_h^s = k, X_0^s = s, L_s = l, H = h \right] \tag{B.21}$$

Rewriting the summand in (B.21) as

$$\mathbb{P}_{G_n} \left[X_i^s = v, X_0^s = s, L_s = l, H = h | X_h^s = k \right] \mathbb{P}_{G_n} \left[X_h^s = k \right],$$

we next aim to apply the Markov property to the conditional probability above. For this, we write $\{L_s = l\} = A_{s,l} \cap (\cap_{j=0}^{l-1} A_{s,j}^C)$, where $A_{s,j}$ denotes the event that a jump to s occurs at step j of the random walk. We then have

$$\left\{ X_i^s = v, X_0^s = s, L_s = l, H = h \right\} = \left\{ X_i^s = v, A_{s,l}, \cap_{j=h+1}^{l-1} A_{s,j}^C \right\} \cap \left\{ H = h, \cap_{j=0}^{h} A_{s,j}^C, X_0^s = s \right\}$$

where on the right side, the first event is the future and the second event is the past, when h is viewed as the present. Hence, the Markov property implies

$$\begin{aligned} \mathbb{P}_{G_n} \left[X_i^s = v, X_0^s = s, L_s = l, H = h | X_h^s = k \right] = \mathbb{P}_{G_n} \left[X_i^s = v, A_{s,l}, \cap_{j=h}^{l-1} A_{s,j}^C \middle| X_h^s = k \right] \\ \times \mathbb{P}_{G_n} \left[H = h, \cap_{j=0}^{h-1} A_{s,j}^C, X_0^s = s \middle| X_h^s = k \right]. \end{aligned} \tag{B.22}$$

Furthermore, by the time invariance of the Markov chain,

$$\begin{aligned} \mathbb{P}_{G_n} \left[X_i^s = v, A_{s,l}, \cap_{j=h}^{l-1} A_{s,j}^C \middle| X_h^s = k \right] &= \mathbb{P}_{G_n} \left[X_{i-h}^s = v, A_{s,l-h}, \cap_{j=0}^{l-h-1} A_{s,j}^C \middle| X_0^s = k \right] \\ &= \mathbb{P}_{G_n} \left[X_{i-h}^s = v, L_s = l - h \middle| X_0^s = k \right]. \end{aligned} \tag{B.23}$$

Finally, by definition of $A_{s,j}$, we have

$$\mathbb{P}_{G_n} \left[H = h, \cap_{j=0}^{h-1} A_{s,j}^C, X_0^s = s \middle| X_h^s = k \right] = \mathbb{P}_{G_n} \left[H = h, L_s > h, X_0^s = s | X_h^s = k \right]. \tag{B.24}$$

Combining (B.21), (B.22), (B.23), and (B.24) then yields

$$\sum_{l>h} \mathbb{E}_{G_n} \left[\sum_{i=H}^{L_s - 1} 1(X_i^s = v, X_H^s = k, X_0^s = s) 1(L_s = l, H = h) \right] \tag{B.25}$$

$$\begin{aligned} &= \sum_{l>h} \sum_{i=h}^{l-1} \mathbb{P}_{G_n} \left[X_{i-h}^s = v, L_s = l - h \middle| X_0^s = k \right] \mathbb{P}_{G_n} \left[H = h, L_s > h, X_0^s = s, X_h^s = k \right] \\ &= \sum_{h \in \mathbb{Z}_+} \mathbb{P}_{G_n} \left[H = h, L_s > h, X_0^s = s, X_h^s = k \right] \sum_{l=h+1}^{\infty} \sum_{i=0}^{l-h-1} \mathbb{P}_{G_n} \left[X_i^s = v, L_s = l - h | X_0^s = k \right], \end{aligned}$$

where in the second equality we have simply rearranged terms and rewritten indices. For

the inner double summation, we have

$$\begin{aligned}
&\sum_{l=h+1}^{\infty}\sum_{i=0}^{l-h-1}\mathbb{P}_{G_n}\left[X_i^s=v, L_s=l-h|X_0^s=k\right]\\
&\quad=\frac{\sum_{l=h+1}^{\infty}\mathbb{E}_{G_n}\left[\sum_{i=0}^{l-h-1}1(X_i^s=v)1(L_s=l-h,X_0^s=k)\right]}{\mathbb{P}_{G_n}[X_0^s=k]}\\
&\quad=\frac{\sum_{l=h+1}^{\infty}\mathbb{E}_{G_n}\left[\sum_{i=0}^{L_s-1}1(X_i^s=v)1(L_s=l-h,X_0^s=k)\right]}{\mathbb{P}_{G_n}[X_0^s=k]}\\
&\quad=\frac{\mathbb{E}_{G_n}\left[\sum_{i=0}^{L_s-1}1(X_i^s=v)1(X_0^s=k)\sum_{l=h+1}^{\infty}1(L_s=l-h)\right]}{\mathbb{P}_{G_n}[X_0^s=k]}\\
&\quad=\mathbb{E}_{G_n}\left[\sum_{i=0}^{L_s-1}1(X_i^s=v)\middle|X_0^s=k\right]=\mathbb{E}_{G_n}[\tau_s(v)|X_0^s=k]=\mathbb{E}_{G_n}[\tau_k(v)|X_0^k=k],
\end{aligned}$$

where the first three steps are straightforward, the fourth step uses the fact that L_s is integer-valued and *a.s.* finite, and the fifth step follows by definition. The final inequality follows because $\tau_s(v)$ and $\tau_k(v)$ count the number of visits to v on the $\{X_i^s\}_{i=0}^{\infty}$ and $\{X_i^k\}_{i=0}^{\infty}$ chains before jumps occur, and before jumps occur, these chains have the same dynamics (since they only differ in jump locations, s versus k). Substituting into (B.25) gives

$$\begin{aligned}
&\sum_{l>h}\mathbb{E}_{G_n}\left[\sum_{i=H}^{L_s-1}1(X_i^s=v,X_H^s=k,X_0^s=s)1(L_s=l,H=h)\right]\\
&\quad=\mathbb{E}_{G_n}[\tau_k(v)|X_0^k=k]\sum_{h\in\mathbb{Z}_+}\mathbb{P}_{G_n}\left[H=h,L_s>h,X_0^s=s,X_h^s=k\right]\\
&\quad=\mathbb{E}_{G_n}[\tau_k(v)|X_0^k=k]\mathbb{P}_{G_n}\left[L_s>H,X_0^s=s,X_H^s=k\right]. \qquad \text{(B.26)}
\end{aligned}$$

Hence, combining (B.20) and (B.26) yields

$$\mathbb{E}_{G_n}\left[\sum_{i=H}^{L_s-1}1(X_i^s=v)\middle|X_H^s=k,L_s>H,X_0^s=s\right]=\mathbb{E}_{G_n}[\tau_k(v)|X_0^k=k]. \qquad \text{(B.27)}$$

Finally, (B.18), (B.19), and (B.27) complete the proof of (B.15).

B.2.2 Proof of Lemma B.2

We aim to bound $\|\pi_s-(\alpha_n e_s^{\mathsf{T}}+\sum_{k\in K_n}\beta_s(k)\pi_k)\|_1$, where

$$\beta_s(k)=\frac{\tilde{\pi}_s(k)}{\alpha_n+(1-\alpha_n)\tilde{\pi}_s(K_n)}\ \forall\ k\in K_n.$$

Using Lemma B.1, we can write

$$\left\|\pi_s - \left(\alpha_n e_s^\mathsf{T} + \sum_{k\in K_n} \beta_s(k)\pi_k\right)\right\|_1 = \sum_{v\in V_n} \left|\frac{\alpha_n U_v \tilde{\pi}_s(v)}{\alpha_n + (1-\alpha_n)\tilde{\pi}_s(K_n)} - \alpha_n 1(v=s)\right|.$$

We next claim that the terms in the absolute values are nonnegative. This is obvious for $v \neq s$. For $v = s$, since $U_s = 1$, we aim to show

$$\tilde{\pi}_s(s) \geq \alpha_n + (1-\alpha_n)\tilde{\pi}_s(K_n). \tag{B.28}$$

To this end, first note that by $\tilde{\pi}_s = \tilde{\pi}_s \tilde{P}_s$ and $\tilde{\pi}_s 1_n = 1$,

$$\tilde{\pi}_s = (1-\alpha_n)\tilde{\pi}_s\left(\tilde{P} + e_{K_n} e_s^\mathsf{T}\right) + \alpha_n e_s^\mathsf{T},$$

which implies

$$\tilde{\pi}_s = \alpha_n e_s^\mathsf{T}\left(I - (1-\alpha_n)\left(\tilde{P} + e_{K_n} e_s^\mathsf{T}\right)\right)^{-1} = \alpha_n e_s^\mathsf{T} \sum_{i=0}^{\infty} (1-\alpha_n)^i \left(\tilde{P} + e_{K_n} e_s^\mathsf{T}\right)^i. \tag{B.29}$$

Using (B.29), we have

$$\begin{aligned}
\tilde{\pi}_s(s) &= \alpha_n e_s^\mathsf{T} \sum_{i=0}^{\infty} (1-\alpha_n)^i \left(\tilde{P} + e_{K_n} e_s^\mathsf{T}\right)^i e_s = \alpha_n + \alpha_n e_s^\mathsf{T} \sum_{i=1}^{\infty} (1-\alpha_n)^i \left(\tilde{P} + e_{K_n} e_s^\mathsf{T}\right)^i e_s \\
&= \alpha_n + \alpha_n(1-\alpha_n) e_s^\mathsf{T} \sum_{i=0}^{\infty} (1-\alpha_n)^i \left(\tilde{P} + e_{K_n} e_s^\mathsf{T}\right)^i e_{K_n} e_s^\mathsf{T} e_s \\
&\quad + \alpha_n(1-\alpha_n) e_s^\mathsf{T} \sum_{i=0}^{\infty} (1-\alpha_n)^i \left(\tilde{P} + e_{K_n} e_s^\mathsf{T}\right)^i \tilde{P} e_s^\mathsf{T},
\end{aligned}$$

and so, discarding a nonnegative term, we obtain

$$\tilde{\pi}_s(s) \geq \alpha_n + \alpha_n(1-\alpha_n) e_s^\mathsf{T} \sum_{i=0}^{\infty} (1-\alpha_n)^i \left(\tilde{P} + e_{K_n} e_s^\mathsf{T}\right)^i e_{K_n} e_s^\mathsf{T} e_s = \alpha_n + (1-\alpha_n)\tilde{\pi}_s e_{K_n}.$$

This establishes (B.28), since $\tilde{\pi}_s e_{K_n} = \tilde{\pi}_s(K_n)$. Hence, we have shown

$$\left\|\pi_s - \left(\alpha_n e_s^\mathsf{T} + \sum_{k\in K_n} \beta_s(k)\pi_k\right)\right\|_1 = \alpha_n \left(\frac{\tilde{\pi}_s(V_n \setminus K_n)}{\alpha_n + (1-\alpha_n)\tilde{\pi}_s(K_n)} - 1\right). \tag{B.30}$$

(We pause to note that since $\tilde{\pi}_s(V_n \setminus K_n) \leq 1$ and $\tilde{\pi}_s(K_n) \geq 0$,

$$\left\|\pi_s - \left(\alpha_n e_s^\mathsf{T} + \sum_{k\in K_n} \beta_s(k)\pi_k\right)\right\|_1 \leq \alpha_n \left(\frac{1}{\alpha_n} - 1\right) = 1 - \alpha_n, \tag{B.31}$$

i.e. π_s is at l_1 distance at most $1-\alpha_n$ from a linear combination of e_s^T and $\{\pi_k\}_{k\in K_n}$.) We next bound the right side of (B.30) in terms of $\mu_s^{(m)}$, as in the statement of the lemma. We begin by establishing a relationship between $\tilde{\pi}_s$ and μ_s, where

$$\mu_s = \lim_{m\to\infty} \mu_s^{(m)} = e_s^\mathsf{T} \sum_{i=0}^{\infty} (1-\alpha_n)^i \tilde{P}^i = e_s^\mathsf{T} \left(I - (1-\alpha_n)\tilde{P}\right)^{-1}. \tag{B.32}$$

To this end, consider the matrix inversion in (B.29). By the Sherman-Morrison-Woodbury formula (see, for example, Section 6.4 of [125]),

$$\begin{aligned}\left(I - (1-\alpha_n)\left(\tilde{P} + e_{K_n} e_s^\mathsf{T}\right)\right)^{-1} &= \left(\left(I - (1-\alpha_n)\tilde{P}\right) - (1-\alpha_n) e_{K_n} e_s^\mathsf{T}\right)^{-1} \\ &= \left(I - (1-\alpha_n)\tilde{P}\right)^{-1} + \frac{\left(I - (1-\alpha_n)\tilde{P}\right)^{-1} (1-\alpha_n) e_{K_n} e_s^\mathsf{T} \left(I - (1-\alpha_n)\tilde{P}\right)^{-1}}{1 - e_s^\mathsf{T} \left(I - (1-\alpha_n)\tilde{P}\right)^{-1} (1-\alpha_n) e_{K_n}}. \end{aligned} \tag{B.33}$$

It follows that, for each $v \in V_n$,

$$\begin{aligned}\tilde{\pi}_s(v) &= \alpha_n e_s^\mathsf{T} \left(I - (1-\alpha_n)\left(\tilde{P} + e_{K_n} e_s^\mathsf{T}\right)\right)^{-1} e_v \\ &= \alpha_n e_s^\mathsf{T} \left(I - (1-\alpha_n)\tilde{P}\right)^{-1} e_v \\ &\qquad + \alpha_n e_s^\mathsf{T} \frac{\left(I - (1-\alpha_n)\tilde{P}\right)^{-1} (1-\alpha_n) e_{K_n} e_s^\mathsf{T} \left(I - (1-\alpha_n)\tilde{P}\right)^{-1}}{1 - e_s^\mathsf{T} \left(I - (1-\alpha_n)\tilde{P}\right)^{-1} (1-\alpha_n) e_{K_n}} e_v \\ &= \alpha_n \mu_s(v) \left(1 + \frac{(1-\alpha_n)\mu_s(K_n)}{1 - (1-\alpha_n)\mu_s(K_n)}\right) = \frac{\alpha_n \mu_s(v)}{1 - (1-\alpha_n)\mu_s(K_n)}, \end{aligned} \tag{B.34}$$

where the first three equalities follow from (B.29), (B.33), and (B.32), respectively, and the fourth involves simple manipulations. We can then combine (B.30) and (B.34) to obtain

$$\left\| \pi_s - \left(\alpha_n e_s^\mathsf{T} + \sum_{k\in K} \beta_s(k)\pi_k\right) \right\|_1 = \alpha_n \left(\mu_s(V_n \setminus K_n) - 1\right). \tag{B.35}$$

Next, we observe

$$\begin{aligned}\mu_s(V_n \setminus K_n) &= \mu_s^{(m)}(V_n \setminus K_n) + e_s^\mathsf{T} \sum_{i=m+1}^{\infty} (1-\alpha_n)^i \tilde{P}^i e_{V_n \setminus K_n} \\ &= \mu_s^{(m)}(V_n \setminus K_n) + e_s^\mathsf{T} (1-\alpha_n)^m \tilde{P}^m \sum_{i=1}^{\infty} (1-\alpha_n)^i \tilde{P}^i e_{V_n \setminus K_n} \\ &= \mu_s^{(m)}(V_n \setminus K_n) + \left(\mu_s^{(m)} - \mu_s^{(m-1)}\right) \sum_{i=1}^{\infty} (1-\alpha_n)^i \tilde{P}^i e_{V_n \setminus K_n}, \end{aligned} \tag{B.36}$$

where we have used (B.2) and (B.32). We next claim $\tilde{P}e_{V_n\setminus K_n} \le e_{V_n\setminus K_n}$ componentwise. To prove this, let $(\tilde{P}e_{V_n\setminus K_n})(i)$ denote the i-th component of $\tilde{P}e_{V_n\setminus K_n}$. Then

$$(\tilde{P}e_{V_n\setminus K_n})(i) = U_i \sum_{j=1}^{n} P(i,j) e_{V_n\setminus K_n}(j) \le U_i \sum_{j=1}^{n} P(i,j) = U_i = e_{V_n\setminus K_n}(i),$$

where the first equality uses the definition of $\tilde{P}$, the second equality holds because P is row stochastic, and the remaining steps are straightforward. It follows that

$$\sum_{i=1}^{\infty} (1-\alpha_n)^i \tilde{P}^i e_{V_n\setminus K_n} \le \left(\sum_{i=1}^{\infty} (1-\alpha_n)^i \right) e_{V_n\setminus K_n} = \left(\frac{1-\alpha_n}{\alpha_n} \right) e_{V_n\setminus K_n}, \tag{B.37}$$

where the inequality is componentwise. Combining (B.36) and (B.37),

$$\begin{aligned} \mu_s(V_n \setminus K_n) &\le \mu_s^{(m)}(V_n \setminus K_n) + \left(\mu_s^{(m)} - \mu_s^{(m-1)}\right) \left(\frac{1-\alpha_n}{\alpha_n} \right) e_{V_n\setminus K_n} \\ &= \frac{1}{\alpha_n} e_s^{\mathsf{T}} (1-\alpha_n)^m \tilde{P}^m e_{V_n\setminus K_n} + \mu_s^{(m-1)}(V_n \setminus K_n), \end{aligned} \tag{B.38}$$

where we also used $\mu_s^{(m)} \ge \mu_s^{(m-1)}$ (componentwise). Finally, (B.35) and (B.38) imply

$$\left\| \pi_s - \left(\alpha_n e_s^{\mathsf{T}} + \sum_{k\in K} \beta_s(k) \pi_k \right) \right\|_1 \le \alpha_n \left(\mu_s^{(m-1)}(V_n \setminus K_n) - 1 \right) + e_s^{\mathsf{T}} (1-\alpha_n)^m \tilde{P}^m e_{V_n\setminus K_n}.$$

B.2.3 Proof of Lemma B.3

We use Algorithm B.3 in Appendix B.2.6, which simultaneously constructs a graph and a tree. We let H_n and $\hat{H}_n$ denote the graph and tree, respectively. From H_n, we define

$$\nu_s^{(m)} = e_s^{\mathsf{T}} \sum_{j=0}^{m} (1-\alpha_n)^j \tilde{Q}^j, \tag{B.39}$$

where $\tilde{Q}(i,j) = U_i Q(i,j)$ and Q is the adjacency matrix of H_n, normalized to be row stochastic. Note this is simply (B.2), i.e. the definition as $\mu_s^{(m)}$, but computed on H_n (while $\mu_s^{(m)}$ is computed on G_n). Similarly, using $\hat{H}_n$, recursively define

$$\hat{\nu}_\phi(\phi) = 1, \quad \hat{\nu}_\phi((\imath, j)) = \hat{\nu}_\phi(\imath) \frac{(1-\alpha_n) U_\imath}{D_\imath}, (\imath, j) \in \hat{A}_l, l > 0, \tag{B.40}$$

which is (B.5) but computed on $\hat{H}_n$ instead of $\hat{G}_n$. With this notation in place, we will show

$$\mu_s^{(m)}(V_n \setminus K_n) | \{\tau_G > m, U_s = 1\} \overset{\mathcal{D}}{=} \nu_s^{(m)}(V_n \setminus K_n) | \{\tau_S > m\}, \tag{B.41}$$

$$\nu_s^{(m)}(V_n \setminus K_n) = \sum_{j=0}^{m} \sum_{\imath \in \hat{A}_j} U_\imath \hat{\nu}_\phi(\imath) \text{ when } \tau_S > m, \tag{B.42}$$

$$\sum_{j=0}^{m} \sum_{\imath \in \hat{A}_j} U_\imath \hat{\nu}_\phi(\imath) | \{\tau_S > m\} \stackrel{\mathcal{D}}{=} \sum_{j=0}^{m} \sum_{\imath \in \hat{A}_j} U_\imath \hat{\mu}_\phi(\imath), \tag{B.43}$$

which, taken together, establish the lemma. (We remind the reader that τ_G and τ_S, respectively, denote the first iteration at which certain events occur in Algorithm B.1 and Algorithm B.3, respectively. Specifically, these events are the following: an instub belonging to v with label $g(v) \in \{C, D\}$ is sampled for pairing to an oustub of v' with label $g(v') = D$, or an instub e with label $g(e) = 0$ is sampled for pairing with *any* outstub.)

We begin with (B.41). First, observe that by definition $\mu_s^{(m)}(V_n \setminus K_n)$ and $\nu_s^{(m)}(V_n \setminus K_n)$ depend only the m-step neighborhood out of s (i.e. the subgraph with nodes $\cup_{j=0}^m A_j$) in G_n and H_n, respectively. When $\tau_G > m, U_s = 1$ in Algorithm B.1 and $\tau_S > m$ in Algorithm B.3, these neighborhoods are constructed by the same procedure. Thus, (B.41) follows.

We next consider (B.43). The left and right sides of (B.43) depend on the first m generations of $\hat{G}_n$ and $\hat{H}_n$, respectively. In Algorithm B.2, these first m generations of $\hat{G}_n$ are constructed as follows: the root node ϕ has attributes $(N_\phi, D_\phi) \sim f_n^*$ and $U_\phi = 1$, non-root nodes $\imath$ have attributes $(N_\imath, D_\imath, U_\imath) \sim f_n$, and $D_\imath$ offspring are born to $\imath$ if and only if $U_\imath = 1$. In Algorithm B.3, the root node in $\hat{H}_n$ also has attributes has attributes $(N_\phi, D_\phi) \sim f_n^*$ and $U_\phi = 1$; furthermore, with $\tau_S > m$, non-root nodes $\imath$ have attributes $(N_\imath, D_\imath, U_\imath) \sim f_n$ and $D_\imath$ offspring are born for either value of $U_\imath$. Hence, when $\tau_S > m$, modifying the construction of the first m generations of $\hat{H}_n$ such that offspring are born only when $U_\imath = 1$ yields the construction of the first m generations of $\hat{G}_n$. But, by (B.40), the left side of (B.43) remains unchanged when this modification occurs. (B.43) follows.

It remains to prove (B.42). For this, we begin with two lemmas. These lemmas use the mapping Φ from graph nodes to tree nodes defined in Algorithm B.3 in Appendix B.2.6. Lemma B.6 states that tree nodes that do not map to graph nodes do not contribute to the right side of (B.42). Lemma B.7 states that a tree node that does map to a graph node contributes to the right side of (B.42) the same value that the corresponding graph node contributes to the left side of (B.42). Together, these lemmas will allow us to prove (B.42).

Lemma B.6. If $\tau_S > m$, $\imath \in \hat{A}_j$ for some $j \in \{0, \ldots, m\}$, and $\Phi^{-1}(\imath) = \emptyset$, then $U_\imath \hat{\nu}_\phi(\imath) = 0$.

Proof. We will denote $\imath$ by $\imath = (i_1, i_2, \ldots, i_j)$, and for $l \leq j$, we let $\imath|l = (i_1, i_2, \ldots, i_l)$, with $\imath|0 = \phi$ by convention. Define $l^* = \max\{l \in \{0, 1, \ldots, j\} : \Phi^{-1}(\imath|l) \neq \emptyset\}$. Note the set over which the maximum is taken is nonempty, since $\Phi^{-1}(\imath|0) = \Phi^{-1}(\phi) = s$; furthermore, since $\Phi^{-1}(\imath|j) = \Phi^{-1}(\imath) = \emptyset$ by assumption, $l^* < j$. In words, $\imath|l^*$ is the youngest ancestor of $\imath$ that maps to a node in the tree; we let $v' = \Phi^{-1}(\imath|l^*)$ denote this node.

We observe $\Phi^{-1}(\imath|l) \neq \emptyset \ \forall \ l \in \{0, 1, \ldots, l^* - 1\}$. To see this, suppose instead that $\Phi^{-1}(\imath|l) = \emptyset$ for some such l. Then, from the second inner for loop in Algorithm B.3, the offspring $\imath|(l+1)$ was born without adding a node to the graph, which implies $\Phi^{-1}(\imath|(l+1)) = \emptyset$. Repeating this argument eventually gives $\Phi^{-1}(\imath|l^*) = \emptyset$, a contradiction.

Now suppose $U_\imath \hat{\nu}_\phi(\imath) > 0$; we seek a contradiction. First, by (B.40), $U_\imath \hat{\nu}_\phi(\imath) > 0$ implies

$$U_{\imath|0} = U_{\imath|1} = \cdots = U_\imath = 1 \tag{B.44}$$

which further implies $U_{\Phi^{-1}(\imath|l)} = U_{\imath|l} = 1 \ \forall \ l \in \{0, 1, \ldots, l^*\}$, i.e. the graph H_n contains a path of length l^* from $s = \Phi^{-1}(\imath|0)$ to $v' = \Phi^{-1}(\imath|l^*)$ that avoids K_n.

Next, note that $\Phi^{-1}(\imath|l^*) \neq \emptyset$, $\Phi^{-1}(\imath|(l^*+1)) = \emptyset$ implies that, during the (l^*+1)-th iteration of Algorithm B.3, an outstub of v' was paired with an instub of some $v \in V_n$ that already belonged to the graph, and so a copy of v (namely, $\imath|(l^*+1)$) was added to the tree. Consider the following cases for the labels of these nodes at the moment of pairing:

- If $g(v') = A$ or $g(v) = A$, we have a contradiction, since by assumption, both v' and v already belonged to the graph at the moment of pairing.
- If $g(v') = B$ or $g(v) = B$, $U_{\imath|l^*} = U_{v'} = 0$ or $U_{\imath|(l^*+1)} = U_v = 0$, contradicting (B.44).
- If $g(v') = D$, $g(v) \in \{C, D\}$, then $\tau_S = l^* \leq m$ in Algorithm B.3, a contradiction.

The remaining case is $g(v') = C$ at the moment of pairing. This contradicts the statement that the graph contains a path from s to v' of length l^* that avoids K_n (since this path was present at start of the (l^*+1)-th iteration, it was present at the moment of pairing). □

Lemma B.7. If $\tau_S > m$, then $U_v \nu_s^{(m)}(v) = U_{\Phi(v)} \hat{\nu}_\phi(\Phi(v)) \ \forall \ v \in \cup_{j=0}^m A_j$.

Proof. We proceed by induction. For the base of induction, we note $A_0 = \{s\}$, so the statement only needs to be verified for $v = s$. But this is immediate, since $\Phi(s) = \phi$ and $U_s = U_\phi = 1$ in Algorithm B.3, and since $\nu_s^{(0)}(s) = \hat{\nu}_\phi(\phi) = 1$ by (B.39) and (B.40).

Now assume $\tau_S > m$ and let $v \in \cup_{j=0}^m A_j$. We consider two cases. First, if $v \in A_j$ for some $j \in \{0, 1, \ldots, m-1\}$, we can use the inductive hypothesis to write

$$U_v \nu_s^{(m)}(v) = U_v \left(\nu_s^{(m)}(v) - \nu_s^{(m-1)}(v)\right) + U_v \nu_s^{(m-1)}(v) = U_v e_s^\mathsf{T} (1-\alpha_n)^m \tilde{Q}^m e_v + U_{\Phi(v)} \hat{\nu}_\phi(\Phi(v)),$$

and so it suffices to show $U_v e_s^\mathsf{T} \tilde{Q}^m e_v = 0$. Clearly, this holds when $U_v = 0$. If instead $U_v = 1$, suppose $e_s^\mathsf{T} \tilde{Q}^m e_v > 0$. First, note that $U_v = 1$ and $v \in A_j, j < m$ imply $g(v) \in \{C, D\}$ at the start of the m-th iteration of Algorithm B.3. Furthermore, $e_s^\mathsf{T} \tilde{Q}^m e_v > 0$ implies there exists a path of length m from s to v, with every node w along the path satisfying $U_w = 1$. Let v' be the node immediately preceding v on this path, so that an outstub of v' was paired with instub of v during the m-th iteration. Then we have $e_s^\mathsf{T} \tilde{Q}^{m-1} e_{v'} > 0$, which implies $g(v') = D$ at the start of the m-th iteration of Algorithm B.3. But $g(v') = D, g(v) \in \{C, D\}$ contradicts $\tau_S > m$ in Algorithm B.3. Therefore, we must have $e_s^\mathsf{T} \tilde{Q}^m e_v = 0$.

Now suppose $v \in A_m$. Then $U_v \nu_s^{(m-1)}(v) = 0$ (else, v is at most $m-1$ steps from s, contradicting $v \in A_m$), so we aim to show $U_v e_s^\mathsf{T} (1-\alpha_n)^m \tilde{Q}^m = U_{\Phi(v)} \hat{\nu}_\phi(\Phi(v))$. Since $U_v = U_{\Phi(v)}$ in Algorithm B.3, this is trivial when $U_v = 0$; when $U_v = 1$, it suffices to show

$$e_s^\mathsf{T} (1-\alpha_n)^m \tilde{Q}^m = \hat{\nu}_\phi(\Phi(v)).$$

Towards this end, let $v' \in \cup_{j=0}^{m-1} A_j$ be the first node whose outstub was paired with an instub of v during the m-th iteration (which occurs by $v \in A_m$); by the inductive hypothesis,

$$U_{v'} \nu_s^{(m-1)}(v') = U_{\Phi(v')} \hat{\nu}_\phi(\Phi(v')).$$

Now since $D_{v'} = D_{\Phi(v')}$, and since $\Phi(v)$ is an offspring of $\Phi(v')$, we can use (B.40) to obtain

$$\frac{(1-\alpha_n) U_{v'} \nu_s^{(m-1)}(v')}{D_{v'}} = \frac{(1-\alpha_n) U_{\Phi(v')} \hat{\nu}_\phi(\Phi(v'))}{D_{\Phi(v')}} = \hat{\nu}_\phi(\Phi(v)). \tag{B.45}$$

Next, observe the left side of (B.45) is at most $e_s^{\mathsf{T}}(1-\alpha_n)^m\tilde{Q}^m$ by (B.39), so we must show this inequality is actually an equality. Suppose instead that the inequality is strict. Then, later in the m-th iteration, we must have paired an outstub of some v'' s.t. $g(v'') = D$ with another instub of v. But $g(v) \in \{C, D\}$ after the v' outstub was paired with the v instub, and $g(v'') = D, g(v) \in \{C, D\}$ contradicts $\tau_S > m$ in Algorithm B.3. □

We now return to the proof of (B.42). Assume $\tau_S > m$. Note that by Lines 19-20 of Algorithm B.3, $\{\Phi(v) : v \in A_j\} \subset \hat{A}_j$, so the right side of (B.42) satisfies

$$\sum_{j=0}^{m}\sum_{\imath\in\hat{A}_j} U_\imath\hat{\nu}_\phi(\imath) = \sum_{j=0}^{m}\left(\sum_{\imath\in\hat{A}_j:\Phi^{-1}(\imath)=\emptyset} U_\imath\hat{\nu}_\phi(\imath) + \sum_{v\in A_j} U_{\Phi(v)}\hat{\nu}_\phi(\Phi(v))\right).$$

Now since $U_\imath\hat{\nu}_\phi(\imath) \geq 0$ by definition, Lemma B.6 implies

$$\sum_{\imath\in\hat{A}_j:\Phi^{-1}(\imath)=\emptyset} U_\imath\hat{\nu}_\phi(\imath) = 0 \;\forall\; j \in \{0, 1, \ldots, m\}.$$

Furthermore, since $\nu_s^{(m)}(v) = 0 \;\forall\; v \notin \cup_{j=0}^m A_j$ (which holds by (B.39)), Lemma B.7 implies

$$\nu_s^{(m)}(V_n \setminus K_n) = \sum_{j=0}^{m}\sum_{v\in A_j} U_v\nu_s^{(m)}(v) = \sum_{j=0}^{m}\sum_{v\in A_j} U_{\Phi(v)}\hat{\nu}_\phi(\Phi(v)).$$

Finally, combining the previous three equations yields (B.42).

B.2.4 Proof of Lemma B.4

We begin with some initial definitions that will be used throughout the proof. Specifically, let $\zeta_n = \mathbb{E}_n[D_\imath]$ and $\lambda_n = \mathbb{E}_n[N_\imath U_\imath]$, where $(N_\imath, D_\imath, U_\imath) \sim f_n$ are the attributes for a non-root node in the tree. Then, conditioned on Ω_n,

$$\zeta_n = \frac{1}{L_n}\sum_{h=1}^{n} N_h D_h = \frac{\eta_2(1+O(n^{-\gamma}))}{\eta_1(1+O(n^{-\gamma}))} = \zeta(1+O(n^{-\gamma})),$$
$$\lambda_n = \frac{1}{L_n}\sum_{h=1}^{n} N_h^2 U_h = \frac{\eta_3(1+O(n^{-\gamma}))}{\eta_1(1+O(n^{-\gamma}))} = \lambda(1+O(n^{-\gamma})).$$

Similarly, let $\zeta_n^* = \mathbb{E}_n[D_\phi]$ and $\lambda_n^* = \mathbb{E}_n[N_\phi]$, where $(N_\phi, D_\phi) \sim f_n^*$ are the attributes for the root node of the tree, so that given Ω_n,

$$\zeta_n^* = \frac{1}{\sum_{h=1}^n U_h}\sum_{h=1}^{n} D_h U_h = \zeta^*(1+O(n^{-\gamma})), \quad \lambda_n^* = \frac{1}{\sum_{h=1}^n U_h}\sum_{h=1}^{n} N_h U_h = \lambda^*(1+O(n^{-\gamma})).$$

We explain our approach for bounding $\mathbb{P}[\tau_G \leq m | U_s = 1]$. First, observe that, conditioned on $U_s = 1$, the graphs in Algorithms B.1 and B.3 (the graph and simultaneous constructions, respectively) are constructed by the same procedure until $\tau_G = m$ or $\tau_S = m$; further, τ_G is

assigned in Algorithm B.1 by the same procedure τ_S is assigned in Algorithm B.3. Thus,

$$\mathbb{P}[\tau_G \le m | U_s = 1] = \mathbb{P}[\tau_S \le m].$$

Next, for $i \in \{0, 1\}$, define

$$E_i = \{g(e) = i \text{ at the moment } \tau_S \text{ is assigned in Algorithm B.3}\}.$$

In other words, E_0 is the event that the coupling breaks because a paired instub was sampled, while E_1 is the event that the coupling breaks because an unpaired instub that forms an edge $v' \to v$ s.t. $g(v') = D, g(v) \in \{C, D\}$ was sampled. Also, for $l \in \{1, 2, \dots, m\}$, let

$$\hat{Z}_l = \sum_{\imath \in \hat{A}_{l-1}} D_\imath, \tag{B.46}$$

which is the total number of outstubs in generation $l-1$ of the tree; note $\hat{Z}_l = |\hat{A}_l|$. Finally, let $\{y_n : n \in \mathbb{N}\}$ be a sequence tending to infinity (which we will choose later), and let

$$F_m = \left\{ \max_{1 \le l \le m} \frac{\hat{Z}_l}{\zeta^{l-1}} \le \zeta^* y_n \right\}.$$

We can then use the previous four equations to write

$$\mathbb{P}[\tau_G \le m | U_s = 1] \le O(n^{-\delta}) + \mathbb{P}[F_m^C | \Omega_n] + \sum_{i=0}^{1} \sum_{l=1}^{m} \mathbb{P}[\tau_S = l, E_i, F_m | \Omega_n]. \tag{B.47}$$

where we also used $\mathbb{P}[\Omega_n^C] = O(n^{-\delta})$ by Assumption 3.1. We next bound each term in (B.47).

To bound $\mathbb{P}[F_m^C | \Omega_n]$, first note $\{D_\imath\}_{\imath \in \hat{A}_{l-1}}$ are identically distributed and independent of $\hat{Z}_{l-1} = |\hat{A}_{l-1}|$, so

$$\mathbb{E}_n[\hat{Z}_l] = \mathbb{E}_n[\mathbb{E}_n[\hat{Z}_l | \hat{Z}_{l-1}]] = \mathbb{E}_n[\hat{Z}_{l-1} \mathbb{E}_n[D_\imath | \hat{Z}_{l-1}]] = \mathbb{E}_n[\hat{Z}_{l-1}] \mathbb{E}_n[D_\imath] = \mathbb{E}_n[\hat{Z}_{l-1}] \zeta_n,$$

and so applying recursively gives

$$\mathbb{E}_n[\hat{Z}_l] = \mathbb{E}_n[\hat{Z}_1] \zeta_n^{l-1} = \mathbb{E}_n[D_\phi] \zeta_n^{l-1} = \zeta_n^* \zeta_n^{l-1}. \tag{B.48}$$

Now let $X_l = \hat{Z}_l / (\zeta_n^* \zeta_n^{l-1})$, so that $\mathbb{E}_n[X_l] = 1$. Furthermore, define

$$\mathcal{G}_l = \sigma(\{N_h, D_h, U_h : 1 \le h \le n\} \cup \{D_\imath : \imath \in \hat{A}_j, 0 \le j < l\}),$$

where by $\sigma(\cdot)$ we mean the generated σ-algebra. Then for $j > 0$,

$$\begin{aligned} \mathbb{E}[X_{l+j} | \mathcal{G}_l] &= \frac{\mathbb{E}[\hat{Z}_{l+j} | \mathcal{G}_l]}{\zeta_n^* \zeta_n^{l+j-1}} = \frac{\mathbb{E}[\hat{Z}_{l+j-1} | \mathcal{G}_l] \mathbb{E}[D_\imath | \mathcal{G}_l]}{\zeta_n^* \zeta_n^{l+j-1}} = \frac{\mathbb{E}[\hat{Z}_{l+j-1} | \mathcal{G}_l]}{\zeta_n^* \zeta_n^{l+j-2}} \\ &= \mathbb{E}[X_{l+j-1} | \mathcal{G}_l] = \cdots = \mathbb{E}[X_l | \mathcal{G}_l] = X_l, \end{aligned}$$

so $\{X_l : l \in \mathbb{N}\}$ is a martingale. This implies, by Doob's inequality,

$$\mathbb{P}_n\left[\max_{1\le l\le m} X_l > \frac{y_n}{(1+O(n^{-\gamma}))^m}\right] \le \frac{(1+O(n^{-\gamma}))^m}{y_n},$$

where we have used $\mathbb{E}_n[X_m] = 1$. Using this bound, we can obtain

$$\begin{aligned}
\mathbb{P}[F_m^C|\Omega_n] &= \mathbb{P}\left[\max_{1\le l\le m}\frac{\hat{Z}_l}{\zeta^{l-1}} > \zeta^* y_n\middle|\Omega_n\right] = \mathbb{P}\left[\max_{1\le l\le m}\frac{X_l\zeta_n^*\zeta_n^{l-1}}{\zeta^*\zeta^{l-1}} > y_n\middle|\Omega_n\right] \\
&= \mathbb{P}\left[\max_{1\le l\le m} X_l(1+O(n^{-\gamma}))^l > y_n\middle|\Omega_n\right] \le \mathbb{P}\left[\max_{1\le l\le m} X_l > \frac{y_n}{(1+O(n^{-\gamma}))^m}\middle|\Omega_n\right] \\
&= \frac{1}{\mathbb{P}[\Omega_n]}\mathbb{E}\left[1(\Omega_n)\mathbb{P}_n\left[\max_{1\le l\le m} X_l > \frac{y_n}{(1+O(n^{-\gamma}))^m}\right]\right] \\
&\le \frac{1}{\mathbb{P}[\Omega_n]}\mathbb{E}\left[1(\Omega_n)\frac{(1+O(n^{-\gamma}))^m}{y_n}\right] = \frac{(1+O(n^{-\gamma}))^m}{y_n} = O(y_n^{-1}),
\end{aligned}$$

where in the third line we used the tower property and the fact that $1(\Omega_n)$ is fixed given the degree sequence, and where the final equality holds by the assumption $m = O(n^\gamma)$ in the statement of the lemma, since then $(1+O(n^{-\gamma}))^m = (1+\frac{O(1)}{m})^m = e^{O(1)} = O(1)$.

To bound $\mathbb{P}[\tau_S = l, E_0, F_m|\Omega_n]$, we first write

$$\mathbb{P}[\tau_S = l, E_0, F_m|\Omega_n] = \mathbb{E}\left[1\left(F_m\right)\mathbb{P}_n\left[\tau_S = l, E_0\middle|\{\hat{Z}_j\}_{j=1}^{m+1}\right]\middle|\Omega_n\right] \tag{B.49}$$

which holds because $1(\Omega_n)$ and $1(F_m)$ are fixed given the degree sequence and $\{\hat{Z}_j\}_{j=1}^m$. Next, observe $\{\tau_S = l, E_0\}$ occurs if and only if, during iteration l, we sample an instub that has already been paired while attempting to pair an outstub belonging to a node $v' \in A_{l-1}$. We aim to bound the probability of this event. Consider any such outstub. Since we sample instubs uniformly from the set of all L_n instubs, the probability of sampling a paired instub is the fraction of paired instubs at the moment we attempt to pair the outstub under consideration. This fraction is clearly bounded by the fraction of paired instubs at the end of iteration l. Further, since each time we pair an instub of $v \in V$ in the graph, we also add a node to the tree with the same attributes as v, the numerator of this fraction is further bounded by the number of nodes in the tree at the end of iteration l, which is

$$\frac{1}{L_n}\sum_{j=1}^{l+1}\hat{Z}_j, \tag{B.50}$$

where (we recall) $L_n = \sum_{v\in V_n} N_v = \sum_{v\in V_n} D_v$. Now consider the number of such outstubs. By definition, this is $\sum_{v'\in A_{l-1}} D_{v'}$. Also, since each time we add a node to A_{l-1} in the graph, we also add a node with the same attributes to $\hat{A}_{l-1}$ in the tree, we have

$$\sum_{v'\in A_{l-1}} D_{v'} \le \sum_{\imath\in\hat{A}_{l-1}} D_\imath \triangleq \hat{Z}_l.$$

Combining these arguments, letting Bin denote a binomial random variable, and using Markov's inequality, we can write

$$\mathbb{P}_n\left[\tau_S = l, E_0 \middle| \{\hat{Z}_j\}_{j=1}^{m+1}\right] \leq \mathbb{P}_n\left[\text{Bin}\left(\hat{Z}_l, \frac{\sum_{j=1}^{l+1}\hat{Z}_j}{L_n}\right) \geq 1 \middle| \{\hat{Z}_j\}_{j=1}^{m+1}\right]$$
$$\leq \mathbb{E}_n\left[\text{Bin}\left(\hat{Z}_l, \frac{\sum_{j=1}^{l+1}\hat{Z}_j}{L_n}\right) \middle| \{\hat{Z}_j\}_{j=1}^{m+1}\right] = \hat{Z}_l\frac{\sum_{j=1}^{l+1}\hat{Z}_j}{L_n}. \quad \text{(B.51)}$$

Next, we recognize $1(F_m)\hat{Z}_l \leq \zeta^*\zeta^{l-1}y_n$ by definition of F_m, so combining (B.49) and (B.51),

$$\mathbb{P}[\tau_S = l, E_0, F_m|\Omega_n] \leq \mathbb{E}\left[1(F_m)\hat{Z}_l\frac{\sum_{j=1}^{l+1}\hat{Z}_j}{L_n}\middle|\Omega_n\right] \leq \zeta^*\zeta^{l-1}y_n\sum_{j=1}^{l+1}\mathbb{E}\left[\frac{\hat{Z}_j}{L_n}\middle|\Omega_n\right].$$

Furthermore, by definition of Ω_n, we have

$$\mathbb{E}\left[\frac{\hat{Z}_j}{L_n}\middle|\Omega_n\right] = \mathbb{E}\left[\frac{\mathbb{E}_n[\hat{Z}_j]}{L_n}\middle|\Omega_n\right] = \mathbb{E}\left[\frac{\zeta_n^*\zeta_n^{j-1}}{L_n}\middle|\Omega_n\right] = \frac{\zeta^*\zeta^{j-1}}{n\eta_1}(1+O(n^{-\gamma}))^j = O\left(\frac{\zeta^{j-1}}{n}\right) \quad \text{(B.52)}$$

where $(1+O(n^{-\gamma}))^j = O(1)$ again follows from $m = O(n^{-\gamma})$. We have therefore shown

$$\mathbb{P}[\tau_S = l, E_0, F_m|\Omega_n] = O\left(\frac{y_n}{n}\zeta^{l-1}\sum_{j=0}^{l}\zeta^j\right).$$

We will use the same approach to bound $\mathbb{P}[\tau_S = l, E_1, F_m|\Omega_n]$ as we used to bound $\mathbb{P}[\tau_S = l, E_0, F_m|\Omega_n]$. First, observe $\{\tau_S = l, E_1\}$ occurs if and only if, during iteration l, we sample an instub belonging to v s.t. $g(v) \in \{C, D\}$ while attempting to pair an outstub belonging to a node $v' \in A_{l-1}$ s.t. $g(v') = D$. The key step in the derivation will be bounding the number of such instubs and outstubs. First, the number of such outstubs is clearly bounded the number of *all* outstubs paired during iteration l. As we argued previously, this is further bounded by $\hat{Z}_l$. Next, for $j \in \{1, \ldots, m+1\}$, define $\hat{V}_j = \sum_{\imath \in \hat{A}_{j-1}} N_\imath U_\imath$. As in the previous argument, the number of such instubs while pairing any such outstub is bounded by the number of instubs belonging to $U_v = 1$ nodes in the graph at the end of iteration l. Since each time we add a node to the graph, we also add a node to the tree with the same attributes, the former quantity is bounded by the same quantity computed on the tree, i.e.

$$\sum_{j=1}^{l+1}\sum_{\imath \in \hat{A}_{l-1}} N_\imath U_\imath = \sum_{j=1}^{l+1}\hat{V}_j.$$

Hence, as in the analysis of $\mathbb{P}[\tau_S = l, E_0, F_m|\Omega_n]$,

$$\mathbb{P}[\tau_S = l, E_1, F_m|\Omega_n] = \mathbb{E}\left[1(F_m)\mathbb{P}_n\left[\tau_S = l, E_1 \middle| \{\hat{Z}_j\}_{j=1}^{m}, \{\hat{V}_j\}_{j=1}^{l+1}\right]\middle|\Omega_n\right]$$

$$\leq \mathbb{E}\left[1(F_m)\hat{Z}_l \frac{\sum_{j=1}^{l+1}\hat{V}_j}{L_n}\middle|\Omega_n\right] \leq \zeta^*\zeta^{l-1}y_n \sum_{j=1}^{l+1}\mathbb{E}\left[\frac{\mathbb{E}_n[\hat{V}_j]}{L_n}\middle|\Omega_n\right].$$

Our final step is to compute $\mathbb{E}_n[\hat{V}_j]$. For $j > 1$, we have

$$\mathbb{E}_n[\hat{V}_j] = \mathbb{E}_n[\hat{Z}_{j-1}]\mathbb{E}_n[N_\imath U_\imath] = \zeta_n^*\zeta_n^{j-2}\lambda_n,$$

where the first equality holds since $|\hat{A}_{j-1}| = \hat{Z}_{j-1}$ and since $\{N_\imath U_\imath : \imath \in \hat{A}_{l-1}\}$ are identically distributed and independent of $\hat{Z}_{j-1}$. Therefore,

$$\mathbb{E}\left[\frac{\mathbb{E}_n[\hat{V}_j]}{L_n}\middle|\Omega_n\right] = \mathbb{E}\left[\frac{\zeta_n^*\zeta_n^{j-2}\lambda_n}{L_n}\middle|\Omega_n\right] = \frac{\zeta^*\zeta^{j-2}\lambda}{n\eta_1}(1+O(n^{-\gamma}))^j = O\left(\frac{\zeta^{j-2}}{n}\right).$$

For $j = 1$, since $\hat{A}_0 = \{\phi\}$ with $U_\phi = 1$, we simply have $\mathbb{E}_n[\hat{V}_1] = \mathbb{E}_n[N_\phi] = \lambda_n^*$, so

$$\mathbb{E}\left[\frac{\mathbb{E}_n[\hat{V}_1]}{L_n}\middle|\Omega_n\right] = \mathbb{E}\left[\frac{\lambda_n^*}{L_n}\middle|\Omega_n\right] = \frac{\lambda^*}{n\eta_1}(1+O(n^{-\gamma})) = O\left(\frac{1}{n}\right).$$

Combining previous arguments, we obtain

$$\mathbb{P}[\tau_S = l, E_1, F_m|\Omega_n] = O\left(\frac{y_n}{n}\zeta^{l-1}\sum_{j=0}^{l-1}\zeta^j\right).$$

B.2.4.1 Overall bound

Combining the bounds from the previous sections, we obtain

$$\mathbb{P}[\tau_G \leq m|U_s = 1] = O\left(n^{-\delta} + y_n^{-1} + \frac{y_n}{n}\sum_{l=1}^{m}\zeta^{l-1}\sum_{j=0}^{l}\zeta^j\right).$$

By Assumption 3.1, we have $\zeta > 1$, which implies

$$\sum_{l=1}^{m}\zeta^{l-1}\sum_{j=0}^{l}\zeta^j = \sum_{l=1}^{m}\zeta^{l-1}\frac{\zeta^{l+1}-1}{\zeta-1} \leq \frac{1}{\zeta-1}\sum_{l=1}^{m}\zeta^{2l} = \frac{\zeta^2(\zeta^{2m}-1)}{(1-\zeta)^2} \leq \left(\frac{\zeta}{\zeta-1}\right)^2\zeta^{2m}.$$

We thus obtain

$$\mathbb{P}[\tau_G \leq m|U_s = 1] = O\left(n^{-\delta} + y_n^{-1} + y_n\zeta^{2m}/n\right)$$

Finally, we choose y_n to minimize the bound. This yields

$$\mathbb{P}[\tau_G \leq m|U_s = 1] = O\left(n^{-\delta} + \zeta^m/\sqrt{n}\right).$$

B.2.5 Proof of Lemma B.5

For $j \in \{1, 2, \ldots, m\}$, let $X_j = \sum_{\imath \in \hat{A}_j} U_\imath \hat{\mu}_\phi(\imath)$, and for $n \in \mathbb{N}$, let

$$\hat{p}_n = \frac{\sum_{h=1}^n U_h N_h}{L_n}.$$

Note that, by Assumption 3.1, $|\hat{p}_n - p| < n^{-\gamma}$ when Ω_n holds.

Before proceeding, we present some intermediate results required for our analysis.

Lemma B.8. $\forall\ i, j \in \mathbb{N}$ s.t. $j \geq i$, let $X^i = \{X_l\}_{l=1}^i$. Then $\mathbb{E}_n[X_j | X^i] = ((1-\alpha_n)\hat{p}_n)^{j-i} X_i$.

Proof. We first observe

$$X_j = \sum_{\imath \in \hat{A}_j} U_\imath \hat{\mu}_\phi(\imath) = \sum_{\imath \in \hat{A}_j} \prod_{l=0}^{j-1} \frac{(1-\alpha_n)U_{\imath|l}}{D_{\imath|l}} U_\imath = \sum_{\imath \in \hat{A}_{j-1}} \prod_{l=0}^{j-1} \frac{(1-\alpha_n)U_{\imath|l}}{D_{\imath|l}} \sum_{k=1}^{D_\imath} U_{(\imath,k)}, \tag{B.53}$$

where the first equality follows from (B.5) and the second follows since, by Algorithm B.2,

$$\hat{A}_j = \left\{(\imath, k) : \imath \in \hat{A}_{j-1}, U_\imath = 1, k \in \{1, 2, \ldots, D_\imath\}\right\}.$$

Next, let $\imath \in \hat{A}_{j-1}$ s.t. $U_\imath = 1$. For each $k \in \{1, 2, \ldots, D_\imath\}$, observe

$$\begin{aligned} &\mathbb{E}\left[U_{(\imath,k)} \middle| \{N_h, D_h, U_h : 1 \leq h \leq n\} \cup \{U_{\imath'}, D_{\imath'} : \imath' \in \hat{A}_s, s < j\}\right] \\ &\quad = \mathbb{E}\left[U_{(\imath,k)} \middle| \{N_h, D_h, U_h : 1 \leq h \leq n\}\right] = \frac{\sum_{h=1}^n U_h N_h}{L_n} = \hat{p}_n, \end{aligned} \tag{B.54}$$

which follows since in Algorithm B.2, $(N_{(\imath,k)}, D_{(\imath,k)}, U_{(\imath,k)})$ are sampled from f_n, independent of the attributes of nodes in previous generations. Combining (B.53) and (B.54) gives

$$\begin{aligned} &\mathbb{E}\left[X_j \middle| \{N_h, D_h, U_h : 1 \leq h \leq n\} \cup \{U_\imath, D_\imath : \imath \in \hat{A}_s, s < j\}\right] \\ &\quad = \sum_{\imath \in \hat{A}_{j-1}} \prod_{l=0}^{j-1} \frac{(1-\alpha_n)U_{\imath|l}}{D_{\imath|l}} \sum_{k=1}^{D_\imath} \hat{p}_n = \sum_{\imath \in \hat{A}_{j-1}} \prod_{l=0}^{j-2} \frac{(1-\alpha_n)U_{\imath|l}}{D_{\imath|l}} \frac{(1-\alpha_n)U_\imath}{D_\imath} (D_\imath \hat{p}_n) \\ &\quad = (1-\alpha_n)\hat{p}_n \sum_{\imath \in \hat{A}_{j-1}} \prod_{l=0}^{j-2} \frac{(1-\alpha_n)U_{\imath|l}}{D_{\imath|l}} U_\imath = (1-\alpha_n)\hat{p}_n \sum_{\imath \in \hat{A}_{j-1}} \hat{\mu}_\phi(\imath) U_\imath = (1-\alpha_n)\hat{p}_n X_{j-1}. \end{aligned}$$

Note that X^i is a function of $\{U_\imath, D_\imath : \imath \in \hat{A}_s, s < j\}$, so we can also write

$$\mathbb{E}\left[X_j \middle| \{N_h, D_h, U_h : 1 \leq h \leq n\} \cup \{U_\imath, D_\imath : \imath \in \hat{A}_s, s < j\} \cup X^i\right] = (1-\alpha_n)\hat{p}_n X_{j-1}.$$

Then, taking conditional expectation with respect to $\{N_h, D_h, U_h : 1 \leq h \leq n\} \cup X^i$,

$$\mathbb{E}_n\left[X_j \middle| X^i\right] = (1-\alpha_n)\hat{p}_n \mathbb{E}_n\left[X_{j-1} | X^i\right],$$

and so applying recursively gives

$$\mathbb{E}_n[X_j|X^i] = ((1-\alpha_n)\hat{p}_n)^{j-i}X_i,$$

which completes the proof. □

Lemma B.9. Let Z be a random variable satisfying $\mathbb{E}[Z] = 0$ and $a \leq Z \leq b$ a.s. Then

$$\mathbb{E}\left[e^{\lambda Z}\right] \leq e^{\lambda^2(b-a)^2/8} \ \forall \ \lambda > 0.$$

Proof. See, for example, Lemma 5.1 in [124]. □

Lemma B.10. For any $j \in \mathbb{N}$ and any $c_j > 0$, define $Y_j = c_j(X_j - (1-\alpha_n)\hat{p}_n X_{j-1}))$. Then

$$\mathbb{E}_n[\exp(\lambda Y_j)|X_{j-1}] \leq \exp\left(\frac{\lambda^2}{8}(c_j(1-\alpha_n)^j)^2\right).$$

Proof. Note $\mathbb{E}_n[Y_j|X_{j-1}] = 0$ by Lemma B.8. Also, $X_j \in [0, (1-\alpha_n)X_{j-1}]$ by (B.5), so

$$Y_j \leq c_j(1-\alpha_n)(1-\hat{p}_n)X_{j-1} \triangleq b_j, \quad Y_j \geq -c_j(1-\alpha_n)\hat{p}_n X_{j-1} \triangleq a_j.$$

Therefore, applying Lemma B.9 gives

$$\mathbb{E}_n[\exp(\lambda Y_j)|X_{j-1}] \leq \exp\left(\frac{\lambda^2}{8}(c_j(1-\alpha_n)X_{j-1})^2\right),$$

and using $X_{j-1} \leq (1-\alpha_n)^{j-1}$ (which again follows from (B.5)) completes the proof. □

We now turn to the proof of the lemma. First, we write

$$\mathbb{P}\left[\alpha_n \sum_{j=1}^{m-1} X_j + X_m \geq \varepsilon\right] \leq \mathbb{P}\left[\alpha_n \sum_{j=1}^{m-1} X_j + X_m \geq \varepsilon \middle| \Omega_n\right] + \mathbb{P}[\Omega_n^C]. \tag{B.55}$$

Recall $\mathbb{P}[\Omega_n^C] = O(n^{-\delta})$ by Assumption 3.1, so it remains to bound the first summand. First,

$$\mathbb{P}\left[\alpha_n \sum_{j=1}^{m-1} X_j + X_m \geq \varepsilon \middle| \Omega_n\right] = \frac{1}{\mathbb{P}[\Omega_n]}\mathbb{E}\left[1(\Omega_n)\,\mathbb{P}_n\left[\alpha_n \sum_{j=1}^{m-1} X_j + X_m \geq \varepsilon\right]\right]. \tag{B.56}$$

For the term inside the expectation, we have

$$\mathbb{P}_n\left[\alpha_n \sum_{j=1}^{m-1} X_j + X_m > \varepsilon\right] \leq \mathbb{P}_n\left[\alpha_n \sum_{j=1}^{m-1} X_j > \frac{\varepsilon}{2}\right] + \mathbb{P}_n\left[X_m > \frac{\varepsilon}{2}\right]. \tag{B.57}$$

For the second summand, we use Markov's inequality to write

$$\mathbb{P}_n\left[X_m > \frac{\varepsilon}{2}\right] \leq \frac{2\mathbb{E}_n[X_m]}{\varepsilon} = \frac{2(1-\alpha_n)^m \hat{p}_n^m}{\varepsilon} < \frac{2\hat{p}_n^m}{\varepsilon}.$$

Recall that $\hat{p}_n \leq p + n^{-\gamma}$ when Ω_n holds. Therefore, by assumption $m = O(n^\gamma)$, we obtain

$$1(\Omega_n)\mathbb{P}_n\left[X_m > \frac{\varepsilon}{2}\right] < \frac{2(p+n^{-\gamma})^m}{\varepsilon} = p^m \frac{2(1+\frac{1/p}{n^\gamma})^m}{\varepsilon} = O\left(p^m\right). \tag{B.58}$$

We next consider the first summand in (B.57). First, we use the Chernoff bound to write

$$\mathbb{P}_n\left[\alpha_n \sum_{j=1}^{m-1} X_j > \varepsilon\right] \leq \min_{\lambda>0} e^{-\lambda\varepsilon}\mathbb{E}_n\left[\prod_{j=1}^{m-1} \exp\left(\lambda\alpha_n X_j\right)\right]. \tag{B.59}$$

To analyze (B.59), we require a definition: for $j = 0, 1, \ldots, m-1$, let

$$c_j = \mathbb{E}_n\left[\alpha_n \sum_{i=0}^{m-j-1} X_i\right] = \alpha_n \sum_{i=0}^{m-j-1} ((1-\alpha_n)\hat{p}_n)^i = \frac{\alpha_n(1-((1-\alpha_n)\hat{p}_n)^{m-j})}{1-(1-\alpha_n)\hat{p}_n}, \tag{B.60}$$

where we have used Lemma B.8 and since, by definition,

$$X_0 = \sum_{\imath\in\hat{A}_0} U_\imath \mu_\phi(\imath) = U_\phi \mu_\phi(\phi) = 1.$$

From (B.60), it is straightforward to show

$$c_0 - \alpha_n = \mathbb{E}_n\left[\alpha_n \sum_{i=1}^{m-1} X_i\right], \quad c_{m-1} = \alpha_n, \quad c_j = \alpha_n + (1-\alpha_n)\hat{p}_n c_{j+1}. \tag{B.61}$$

Now for $j \in \{1, \ldots, m-1\}$, we use Lemma B.10 and (B.61) to obtain

$$\begin{aligned}
&\mathbb{E}_n[\exp(\lambda c_j X_j)|X^{j-1}] \\
&\quad = \mathbb{E}_n[\exp(\lambda c_j(X_j - (1-\alpha_n)\hat{p}_n X_{j-1}))|X^{j-1}]\exp(\lambda c_j(1-\alpha_n)\hat{p}_n X_{j-1}) \\
&\quad \leq \exp\left(\frac{\lambda^2}{8}(c_j(1-\alpha_n)^j)^2\right)\exp(\lambda(c_{j-1}-\alpha_n)X_{j-1}).
\end{aligned}$$

We can then apply to the expectation in (B.59), i.e.

$$\begin{aligned}
\mathbb{E}_n\left[\prod_{j=1}^{m-1} \exp\left(\lambda\alpha_n X_j\right)\right] &= \mathbb{E}_n\left[\prod_{j=1}^{m-2} \exp\left(\lambda\alpha_n X_j\right)\exp\left(\lambda c_{m-1} X_{m-1}\right)\right] \\
&= \mathbb{E}_n\left[\prod_{j=1}^{m-2} \exp\left(\lambda\alpha_n X_j\right)\mathbb{E}_n\left[\exp\left(\lambda c_{m-1} X_{m-1}\right)\middle| X^{m-2}\right]\right] \\
&\leq \mathbb{E}_n\left[\prod_{j=1}^{m-2} \exp\left(\lambda\alpha_n X_j\right)\exp(\lambda(c_{m-2}-\alpha_n)X_{m-2})\right]\exp\left(\frac{\lambda^2}{8}\left(c_{m-1}(1-\alpha_n)^{m-1}\right)^2\right)
\end{aligned}$$

and so applying recursively eventually gives

$$\mathbb{E}_n\left[\prod_{j=1}^{m-1}\exp\left(\lambda\alpha_n X_j\right)\right] \leq \exp\left(\lambda\mathbb{E}_n\left[\alpha_n\sum_{i=1}^{m-1}X_i\right] + \frac{\lambda^2}{8}\sum_{j=1}^{m-1}\left(c_j(1-\alpha_n)^j\right)^2\right),$$

where we have also used $X_0 = 1$ and (B.61). Substituting into (B.59),

$$\mathbb{P}_n\left[\alpha_n\sum_{j=1}^{m-1}X_j > \frac{\varepsilon}{2}\right] \leq \min_{\lambda>0}\exp\left(-\lambda\left(\frac{\varepsilon}{2} - \mathbb{E}_n\left[\alpha_n\sum_{i=1}^{m-1}X_i\right]\right) + \frac{\lambda^2}{8}\sum_{j=1}^{m-1}\left(c_j(1-\alpha_n)^j\right)^2\right). \tag{B.62}$$

It is straightforward to show the global minimizer of (B.62) is

$$\lambda^* = \frac{4\left(\frac{\varepsilon}{2} - \mathbb{E}_n\left[\alpha_n\sum_{i=1}^{m-1}X_i\right]\right)}{\sum_{j=1}^{m-1}\left(c_j(1-\alpha_n)^j\right)^2},$$

which is positive when $\mathbb{E}_n\left[\alpha_n\sum_{i=1}^{m-1}X_i\right] < \frac{\varepsilon}{2}$. Plugging into (B.62),

$$\mathbb{P}_n\left[\alpha_n\sum_{j=1}^{m-1}X_j > \frac{\varepsilon}{2}\right] \leq \exp\left(-\frac{2\left(\frac{\varepsilon}{2} - \mathbb{E}_n\left[\alpha_n\sum_{i=1}^{m-1}X_i\right]\right)^2}{\sum_{j=1}^{m-1}\left(c_j(1-\alpha_n)^j\right)^2}\right). \tag{B.63}$$

We now derive bounds for the denominator and numerator in the exponential in (B.63). To (coarsely) approximate the denominator,

$$\begin{aligned} &c_j < \frac{\alpha_n}{1-(1-\alpha_n)\hat{p}_n} < \frac{\alpha_n}{1-\hat{p}_n}, \quad \sum_{j=1}^{m-1}(1-\alpha_n)^{2j} < \sum_{j=0}^{\infty}(1-\alpha_n)^j = \frac{1}{\alpha_n} \\ &\Rightarrow \sum_{j=1}^{m-1}\left(c_j(1-\alpha_n)^j\right)^2 < \frac{\alpha_n}{(1-\hat{p}_n)^2} < \frac{\alpha_n}{(1-p-n^{-\gamma})^2}, \end{aligned} \tag{B.64}$$

where the final inequality holds assuming Ω_n and n is sufficiently large (so that $p+n^{-\gamma} < 1$). For the numerator, first observe that, when Ω_n holds, we have

$$\mathbb{E}_n\left[\alpha_n\sum_{i=1}^{m-1}X_i\right] = \frac{\alpha_n((1-\alpha_n)\hat{p}_n - ((1-\alpha_n)\hat{p}_n)^m)}{1-(1-\alpha_n)\hat{p}_n} < \frac{\alpha_n}{1-(p+n^{-\gamma})},$$

and so $\mathbb{E}_n[\alpha_n\sum_{i=1}^{m-1}X_i] < \varepsilon/2$ for n sufficiently large (as required), assuming $\alpha_n \to 0$ as $n \to \infty$. Therefore, when Ω_n holds, $\alpha_n \to 0$, and n is large,

$$\begin{aligned} \left(\frac{\varepsilon}{2} - \mathbb{E}_n\left[\alpha_n\sum_{i=1}^{m-1}X_i\right]\right)^2 &> \left(\frac{\varepsilon}{2} - \frac{\alpha_n}{1-(p+n^{-\gamma})}\right)^2 \\ &= \frac{\varepsilon^2}{4} - \frac{\alpha_n}{1-(p+n^{-\gamma})}\left(\varepsilon - \frac{\alpha_n}{1-(p+n^{-\gamma})}\right). \end{aligned} \tag{B.65}$$

Thus, under these assumptions, (B.64) and (B.65) give

$$\frac{2\left(\frac{\varepsilon}{2}-\mathbb{E}_n\left[\alpha_n\sum_{i=1}^{m-1}X_i\right]\right)^2}{\sum_{j=1}^{m-1}\left(c_j(1-\alpha_n)^j\right)^2} > \frac{(1-p-n^{-\gamma})^2\varepsilon^2}{2\alpha_n} - 2(1-p-n^{-\gamma})\left(\varepsilon - \frac{\alpha_n}{1-(p+n^{-\gamma})}\right).$$

To summarize, we have shown that for n sufficiently large, assuming $\alpha_n \to 0$ and Ω_n holds,

$$\begin{aligned}\mathbb{P}_n\left[\alpha_n\sum_{j=1}^{m-1}X_j > \frac{\varepsilon}{2}\right] &\le \exp\left(-\frac{(1-p-n^{-\gamma})^2\varepsilon^2}{2\alpha_n}\right) \\ &\quad\times\exp\left(2(1-p-n^{-\gamma})\left(\varepsilon-\frac{\alpha_n}{1-(p+n^{-\gamma})}\right)\right) \\ &= O\left(\exp\left(-\frac{((1-p)\varepsilon)^2}{2\alpha_n}\right)\right) \qquad \text{(B.66)}\end{aligned}$$

where the equality holds because the second exponential term is $O(1)$ for $p \in (0,1)$. Finally, we combine (B.55), (B.56), (B.57), (B.58), and (B.66) to obtain

$$\mathbb{P}\left[\alpha_n\sum_{j=1}^{m}X_j + X_m > \varepsilon\right] = O\left(n^{-\delta} + p^m + e^{-((1-p)\varepsilon)^2/(2\alpha_n)}\right),$$

which completes the proof.

B.2.6 Simultaneous construction of graph and tree

For the proofs of Lemmas B.3 and B.4, we use Algorithm B.3, which simultaneously constructs a graph and a tree. Algorithm B.3 uses similar notation as Algorithms B.1 and B.2 in Appendix B.1.2. However, there are some differences, which we explain first.

- In Algorithm B.1, we chose $s \sim V_n$ uniformly, which is the standard DCM construction. In Algorithm B.3, we instead choose $s \sim V_n \setminus K_n$ uniformly. This is because in the statement of Lemma B.3 involves $\mu_s^{(m)}(V_n \setminus K_n)$, conditioned on $U_s = 1$ (i.e. $s \in V_n \setminus K_n$); similarly, the statement of Lemma B.4 involves $\{\tau_G \le m\}$, conditioned on $U_s = 1$.
- Algorithm B.3 uses a function $\Phi : V_n \to \mathcal{U}$, where $\mathcal{U} = \cup_{j=0}^{\infty}\mathbb{N}^j$ and $\mathbb{N}^0 = \{\phi\}$ by convention. The function Φ will be used to map nodes in the graph (which have labels in the set V_n) to nodes in the tree (which have labels in the set $\mathcal{U}$).
- The variable τ_S in Algorithm B.3 denotes the first iteration at which events that break the coupling occur (analogous to τ_G in Algorithm B.1). Once these events occur, the simultaneous construction terminates, and the graph and tree constructions are continued separately using Algorithms B.1 and B.2, respectively.

For illustrative purposes, we include an example of the simultaneous construction in Figure B.4. The basic idea is as follows. Whenever a new node is added to the graph, (which occurs when outstub (v', j) is paired with an instub belonging to $v \in V_n$ s.t. $g(v) = A$) a new offspring (with the same attributes as v) is added to the tree, and a map between the graph node and tree offspring is defined. In particular, Figure B.4 has the following mapping:

$$\Phi(s) = \phi, \quad \Phi(1) = (1), \quad \Phi(2) = (2), \quad \Phi(3) = (1,2),$$

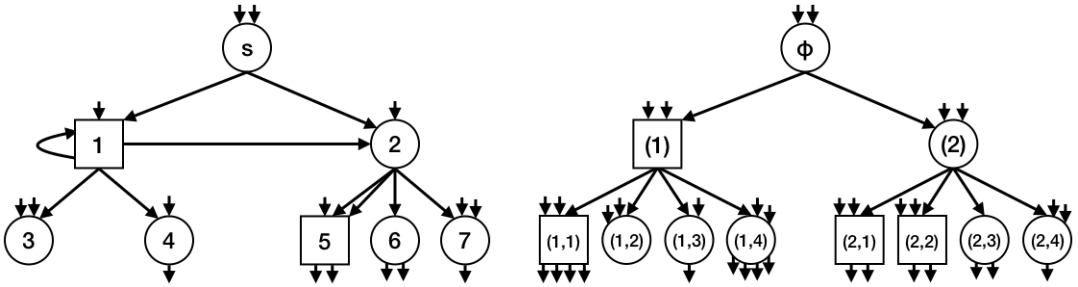


Figure B.4: Simultaneous construction of graph (left) and tree (right).

$$\Phi(4) = (1,3), \quad \Phi(5) = (2,1), \quad \Phi(6) = (2,3), \quad \Phi(7) = (2,4).$$

If an edge is added between two nodes already in the graph (which occurs when outstub (v', j) is paired with $v \in V_n$ s.t. $g(v) \in \{B, C, D\}$), a new offspring with the same attributes as v is added to the tree. This is illustrated by the following examples:

- Node 1 in the graph adds an edge to itself; $(1,1)$ in the tree has the attributes of 1
- Node 1 in the graph adds an edge to 2; $(1,4)$ in the tree has the attributes of 2
- Node 2 in the graph adds a multi-edge to 5; $(2,2)$ in the tree has the attributes of 5

These offspring can be thought of as copies of nodes already in the tree: $(1,1)$, $(1,4)$, and $(2,2)$ are copies of (1), (2), and $(2,1)$, respectively. Furthermore, note that for $\imath \in \{(1,1),(1,4),(2,2)\}$, $\Phi^{-1}(\imath) = \emptyset$. In other words, copies of nodes in the tree do not map to nodes in the graph. This implies that we may have more nodes in the tree than in the graph. For this reason, after pairing all outstubs belonging to all $v' \in A_{m-1}$ (which map to nodes in the tree), we must separately add offspring to nodes $\imath \in \hat{A}_{m-1}$ s.t. $\Phi^{-1}(\imath) = \emptyset$ (which do *not* map to nodes in the tree). This is done in Lines 27-29 in Algorithm B.3.

B.3 Proof of Theorem 3.1

First, we observe

$$\begin{aligned}
\mathbb{E}[\Delta(K_n, \varepsilon)] &= \mathbb{E}\left[|K_n| + |\{v \in V_n \setminus K_n : B_v(K_n, \varepsilon) \text{ holds}\}|\right] \\
&= O\left(n^{\kappa}\right) + \sum_{v \in V_n} \mathbb{E}\left[1(B_v(K_n, \varepsilon), U_v = 1)\right] = O\left(n^{\kappa}\right) + n\mathbb{E}\left[1(B_s(K_n, \varepsilon), U_s = 1)\right] \\
&\leq O\left(n^{\kappa}\right) + n\mathbb{P}[B_s(K_n, \varepsilon) | U_s = 1] = O\left(n^{\kappa}\right) + nO\left(n^{-c(\varepsilon)}\right) = O\left(n^{\max\{\kappa, 1-c(\varepsilon)\}}\right),
\end{aligned}$$

where the steps hold by definition of $\Delta(K_n, \varepsilon)$, by Assumption 3.2, since $1(B_v(K_n, \varepsilon), U_v = 1)$ are identically distributed before the degree sequence is realized, since $\mathbb{P}[U_v = 1] \leq 1$, and by Lemma 3.1, respectively. Hence, by Markov's inequality,

$$\mathbb{P}\Big[\Delta(K_n, \varepsilon) \geq Cn^{\bar{c}}\Big] \leq \frac{\mathbb{E}[\Delta(K_n, \varepsilon)]}{Cn^{\bar{c}}} = O\left(n^{\max\{\kappa, 1-c(\varepsilon)\}-\bar{c}}\right).$$

Algorithm B.3: Simultaneous Construction

Choose s from $V_n \setminus K_n$ uniformly, set $g(s) = D$, set $A_0 = \{s\}$
Set $g(e) = 1 \; \forall \; e \in S$, set $g(v) = A \; \forall \; v \in V_n \setminus \{s\}$
Set $(N_\phi, D_\phi, U_\phi) = (N_s, D_s, U_s)$, set $\hat{A}_0 = \{\phi\}$
Set $\Phi(s) = \phi$, set $\tau_S = \infty$
for $m = 1$ **to** ∞ **do**
    Set $A_m = \hat{A}_m = \emptyset$
    for $v' \in A_{m-1}$ **do**
        Let $\imath = \Phi(v')$
        for $j = 1$ **to** $D_{v'}$ **do**
            // find instub for pairing, check if failure has occurred
            Uniformly sample instub e, denote instub node by v
            if $g(e) = 0$ *or* $g(e) = 1, g(v') = D, g(v) \in \{C, D\}$ **then**
                Set $\tau_S = m$
                Continue constructing graph as in Algorithm B.1
                Continue constructing tree as in Algorithm B.2
                return
            // update graph, tree, and map
            Pair (v', j) with e, set $g(e) = 0$
            if $g(v) = A$ **then** set $A_m = A_m \cup \{v\}$, set $\Phi(v) = (\imath, j)$
            Add offspring $(\imath, j)$ to $\imath$, set $(N_{(\imath,j)}, D_{(\imath,j)}, U_{(\imath,j)}) = (N_v, D_v, U_v)$, set $\hat{A}_m = \hat{A}_m \cup \{(\imath, j)\}$
            // update node label in graph
            if $U_v = 0, g(v) = A$ **then** set $g(v) = B$
            else if $U_v = 1, g(v) = A, g(v') = B$ **then** set $g(v) = C$
            else if $U_v = 1, g(v) = A, g(v') \in \{C, D\}$ **then** set $g(v) = g(v')$
        if $g(e') = 0 \; \forall \; e' \in I_n$ **then return**
    // generate offspring for tree nodes not mapped to a graph node
    for $\imath \in \hat{A}_{m-1}$ *s.t.* $\Phi^{-1}(\imath) = \emptyset$ **do**
        for $j = 1$ **to** $D_\imath$ **do**
            Add offspring $(\imath, j)$ to $\imath$, sample $(N_{(\imath,j)}, D_{(\imath,j)}, U_{(\imath,j)})$ from f_n, set $\hat{A}_m = \hat{A}_m \cup \{(\imath, j)\}$

B.4 Proof of Theorem 3.2

B.4.1 Analysis of subroutines of Algorithm 3.1

We begin with analyses of `Approx-PageRank` and `Approx-Contributions`. Namely, Lemma B.11 gives accuracy and complexity guarantees for `Approx-PageRank`, while Lemma B.12 and Corollary B.1 provide guarantees for `Approx-Contributions`. We note that these results are essentially restatements of those found in [7, 32]; we have included the arguments because they need to be slightly modified and to state them using our notation. These arguments are also similar to those used in Appendices A.1-A.2 for our PPR algorithm and in Chapter IV for our policy evaluation algorithm.

Lemma B.11. For any G_n, $v \in V_n$, and $\varepsilon_1 \in (0,1)$, `Approx-PageRank`(v, ε_1) has complexity $O(L_n/(\alpha_n \varepsilon_1))$, and the output $\hat{\pi}_v$ satisfies $\|\pi_v - \hat{\pi}_v\|_1 \leq \varepsilon_1$, $\hat{\pi}_v(u) \leq \pi_v(u)\ \forall\ u \in V_n$.

Proof. We first claim that for each $u \in V_n$ and at each iteration of `Approx-PageRank`,

$$\pi_v(u) = \hat{\pi}_v(u) + \sum_{w \in V_n} r_v(w)\pi_w(u). \tag{B.67}$$

To prove (B.67), first note that since $\hat{\pi}_v$ and r_v are initialized to 0_n and e_v, respectively, it holds trivially at the beginning of the algorithm. Now assume (B.67) holds before $\hat{\pi}_v$ and r_v are updated at some iteration. Then after the update, we will have

$$\begin{aligned}
&(\hat{\pi}_v(u) + \alpha_n r_v(v^*)1(u = v^*)) + \sum_{w \in V_n} \left(r_v(w)1(w \neq v^*) + (1-\alpha_n)r_v(v^*)P(v^*, w)\right)\pi_w(u) \\
&\quad = (\hat{\pi}_v(u) + \alpha_n r_v(v^*)1(u = v^*)) \\
&\qquad + \sum_{w \in V_n \setminus \{v^*\}} r_v(w)\pi_w(u) + r_v(v^*)(1-\alpha_n)\sum_{w \in V_n} P(v^*, w)\pi_w(u) \\
&\quad = (\hat{\pi}_v(u) + \alpha_n r_v(v^*)1(u = v^*)) + \sum_{w \in V_n \setminus \{v^*\}} r_v(w)\pi_w(u) + r_v(v^*)\left(\pi_{v^*}(u) - \alpha_n 1(u = v^*)\right) \\
&\quad = \hat{\pi}_v(u) + \sum_{w \in V_n} r_v(w)\pi_w(u) = \pi_v(u),
\end{aligned}$$

where the final equality uses the assumption that (B.67) holds before the update, and where the second equality holds by (1.1). Next, observe that $\pi_w(u) \geq 0\ \forall\ w, u \in V_n$ by definition; further, $r_v(w) \geq 0\ \forall\ w \in V_n$ for the duration of the algorithm. Together with (B.67), this implies $\hat{\pi}_v(u) \leq \pi_v(u)\ \forall\ u \in V_n$, as claimed.

To show $\|\pi_v - \hat{\pi}_v\|_1 \leq \varepsilon_1$ at termination, observe

$$\begin{aligned}
\|\pi - \hat{\pi}_v\|_1 &= \sum_{u \in V_n} (\pi_v(u) - \hat{\pi}_v(u)) = \sum_{u \in V_n}\sum_{w \in V_n} r_v(w)\pi_w(u) = \sum_{w \in V_n} r_v(w)\sum_{u \in V_n}\pi_w(u) \\
&= \sum_{w \in V_n} r_v(w) = \sum_{w \in V_n} \frac{r_v(w)}{D_w} D_w \leq \frac{\varepsilon_1}{L_n}\sum_{w \in V_n} D_w = \varepsilon_1,
\end{aligned}$$

where the first equality holds by $\hat{\pi}_v(u) \leq \pi_v(u)$, the second holds by (B.67), and the fourth

uses the fact that π_w sums to 1 (the others are immediate); the inequality holds at termination of the algorithm via the terminating condition of the `while` loop.

For the complexity guarantee, let i^* denote the iteration at which the algorithm terminates, and let v_i be the node chosen as v^* during the i-th iteration. Then it is readily verified that $\|r_v\|_1$ decreases by $\alpha_n r_v(v_i) = \alpha_n \frac{r_v(v_i)}{D_{v_i}} D_{v_i} \geq \alpha_n \frac{\varepsilon_1}{L_n} D_{v_i}$ at the i-th iteration. Hence, because $\|r_v\|_1 = 1$ initially and is bounded below by zero,

$$1 \geq \sum_{i=1}^{i^*} \alpha_n r_v(v_i) \geq \alpha_n \frac{\varepsilon_1}{L_n} \sum_{i=1}^{i^*} D_{v_i} \Rightarrow \sum_{i=1}^{i^*} D_{v_i} \leq \frac{L_n}{\alpha_n \varepsilon_1}.$$

On the other hand, at most D_{v_i} elements of the r_v vector and one element of the $\hat{\pi}_v$ vectors are updated at iteration i, so the complexity of the algorithm scales with $\sum_{i=1}^{i^*} D_{v_i}$. Hence, the complexity is bounded by $L_n/(\alpha_n \varepsilon_1)$, as claimed. □

Lemma B.12. For G_n, $v \in V_n$, and $\varepsilon_2 \in (0,1)$, `Approx-Contributions`(v, ε_2) has complexity

$$O\left(\frac{1}{\varepsilon_2} \sum_{u \in V} \mu_u(v) N_u\right),$$

and the output $\{\hat{\mu}_u(v)\}_{u \in V_n}$ satisfies $|\mu_u(v) - \hat{\mu}_u(v)| \leq \varepsilon_2/\alpha_n$ and $\hat{\mu}_u(v) \leq \mu_u(v)\ \forall\ u \in V_n$, where $\mu_u(v)$ is the v-th element of the vector μ_u given by (B.32) in Appendix B.2.2.

Proof. We begin with a claim analogous to (B.67); namely, that for each $u \in V_n$ and at each iteration of `Approx-Contributions`,

$$\mu_u(v) = \hat{\mu}_u(v) + \sum_{w \in V_n} \mu_u(w) r_v(w). \tag{B.68}$$

As for (B.67), (B.68) is immediate at the start of the algorithm, and if it holds before $\{\hat{\mu}_u(v)\}_{u \in V_n}$ and r_v are updated, we have

$$\begin{aligned}
&(\hat{\mu}_u(v) + r_v(v^*)1(u = v^*)) + \sum_{w \in V_n} \mu_u(w) \left(r_v(w)1(w \neq v^*) + (1 - \alpha_n) r_v(v^*) \tilde{P}(w, v^*)\right) \\
&\quad = (\hat{\mu}_u(v) + r_v(v^*)1(u = v^*)) + \sum_{w \in V_n \setminus \{v^*\}} \mu_u(w) r_v(w) + r_v(v^*)(1 - \alpha_n) \sum_{w \in V_n} \mu_u(w) \tilde{P}(w, v^*) \\
&\quad = (\hat{\mu}_u(v) + r_v(v^*)1(u = v^*)) + \sum_{w \in V_n \setminus \{v^*\}} \mu_u(w) r_v(w) + r_v(v^*) \left(\mu_u(v^*) - 1(u = v^*)\right) \\
&\quad = \hat{\mu}_u(v) + \sum_{w \in V_n} \mu_u(w) r_v(w) = \mu_u(v),
\end{aligned}$$

where the final step is because (B.68) holds before the update by assumption, and the second equality holds by (B.32). From (B.68), the fact that $\mu_u(w) \geq 0\ \forall\ u, w \in V_n$ by definition, and the fact that $r_v(w) \geq 0\ \forall\ w \in V_n$ for the duration of the algorithm, we immediately obtain $\hat{\mu}_u(v) \leq \mu_u(v)\ \forall\ u \in V_n$, as claimed.

For the accuracy guarantee, note that at termination, we have

$$\mu_u(v) - \hat{\mu}_u(v) = \sum_{w \in V_n} \mu_u(w) r_v(w) \leq \varepsilon_2 \sum_{w \in V_n} \mu_u(w) \leq \varepsilon_2 / \alpha_n,$$

where the equality holds by (B.68), the first inequality is by the terminating condition of the `while` loop, and the final inequality holds by definition of μ_u and the fact that $\tilde{P}$ is nonnegative with row sums bounded by 1, together which imply

$$\sum_{w \in V_n} \mu_u(w) = \sum_{i=0}^{\infty} (1-\alpha_n)^i e_u^\intercal \tilde{P}^i 1_n \leq \sum_{i=0}^{\infty} (1-\alpha_n)^i = \frac{1}{\alpha_n}.$$

For the complexity guarantee, first note that $\hat{\mu}_u(v)$ increases by $r_v(u) > \varepsilon_2$ at each iteration for which $v^* = u$; hence, because $\hat{\mu}_u(v) \leq \mu_u(v)$, we can have $v^* = u$ for at most $\mu_u(v)/\varepsilon_2$ iterations. Also, the complexity of each such iteration scales with N_u (as in the argument in the proof of Lemma B.11). Hence, the complexity is bounded by $\sum_{u \in V_n} \mu_u(v) N_u / \varepsilon_2$. □

Corollary B.1. For any G_n and $\varepsilon_2 \in (0,1)$, running `Approx-Contributions`(k, ε_2) for each $k \in K_n$ produces output $\hat{\mu}_v(k)$ satisfying $|\mu_v(k) - \hat{\mu}_v(k)| \leq \varepsilon_2/\alpha_n$ and $\hat{\mu}_v(k) \leq \mu_v(k)$ for each $v \in V_n, k \in K_n$; also, if $L_n = O(n)$ and ε_2 depends on n, the complexity is $O(n/\varepsilon_2)$.

Proof. The accuracy guarantee follows from Lemma B.12. Also by Lemma B.12, we can bound the complexity as

$$\frac{1}{\varepsilon_2} \sum_{k \in K_n} \sum_{u \in V_n} \mu_u(k) N_u = \frac{1}{\varepsilon_2} \sum_{u \in V_n} N_u \mu_u(K_n).$$

Using (B.34), it is straightforward to show

$$\mu_u(K_n) = \frac{\tilde{\pi}_u(K_n)}{\alpha_n + (1-\alpha_n)\tilde{\pi}_u(K_n)} \leq 1 \ \forall \ u \in V_n,$$

where we have also used $\tilde{\pi}_u(K_n) \leq 1$. Combining the previous two equations gives complexity L_n/ε_2, from which the corollary follows by assumption $L_n = O(n)$. □

B.4.2 Proof of Theorem 3.2

With Lemmas B.11 and B.12 and Corollary B.1 in place, we turn to the proof of Theorem 3.2. We begin with the complexity guarantee. For this, we will proceed through each of the six computations undertaken by Algorithm 3.1 and bound the complexity of each.

First, let $C^{(1)}_{Alg3.1}$ denote the complexity of running `Approx-Page-Rank`$(k, \varepsilon_1) \ \forall \ k \in K_n$. By Lemma B.11, $C^{(1)}_{Alg3.1} = O(|K_n| L_n / (\alpha_n \varepsilon_1))$. Since $L_n = O(n)$ when Ω_n holds, $\alpha_n = \Theta(\frac{1}{\log n})$, and ε_1 is constant, we conclude

$$\mathbb{E}\left[C^{(1)}_{Alg3.1} \middle| \Omega_n \right] = O\left(\mathbb{E}[|K_n| | \Omega_n] n \log n \right).$$

Next, let $C^{(2)}_{Alg3.1}$ denote the complexity of running `Approx-Cont- ributions`(k, ε_2) for every

$k \in K_n$. By Corollary B.1, this has complexity $O(n/\varepsilon_2)$ when $L_n = O(n)$, which occurs when Ω_n holds. Since also $\varepsilon_2 = \frac{\alpha_n^2 g_n(\varepsilon/4)}{2|K_n|}$, this is $O(\mathbb{E}[|K_n||\Omega_n]n\alpha_n^2 g_n(\varepsilon/4))$ in expectation when Ω_n holds. Furthermore, since $\alpha_n \to 0$ as $n \to \infty$ and $\varepsilon \in (0,1)$ is constant, we have $1 - 2(\varepsilon/4) < 1 - \alpha_n - (\varepsilon/4)$ and $\varepsilon/4 > \alpha_n(2 - \alpha_n - (\varepsilon/4))$ for large enough n; for such n,

$$\frac{\alpha_n(1-2(\varepsilon/4))}{2(\varepsilon/4)} < \frac{\alpha_n(1-\alpha_n-(\varepsilon/4))}{(\varepsilon/4)+(2-\alpha_n-(\varepsilon/4))} = g_n(\varepsilon/4).$$

It is also immediate that $g_n(\varepsilon/4) \leq \frac{\alpha_n}{\varepsilon/4}$. Taken together,

$$g_n(\varepsilon/4) \in \left(\frac{\alpha_n(1-2(\varepsilon/4))}{2(\varepsilon/4)}, \frac{\alpha_n}{\varepsilon/4}\right) \Rightarrow g_n(\varepsilon/4) = \Theta(\alpha_n). \tag{B.69}$$

Hence, with $\alpha_n = \Theta(\frac{1}{\log n})$, we conclude

$$\mathbb{E}\left[C^{(2)}_{Alg3.1}\middle|\Omega_n\right] = O\left(\mathbb{E}[|K_n||\Omega_n]n(\log n)^3\right).$$

Next, let $C^{(3)}_{Alg3.1}$ denote the complexity of constructing $\hat{\pi}^l_k$. We claim that this can be done while running `Approx-PageRank`(k, ε_1) without increasing the order of the `Approx-PageRank` complexity. This argument is based on the fact that `Approx-PageRank`(k, ε_1) essentially completes a breadth-first-search out of k. First, we can set $V_{n,k}(0) = \{k\}$ at the initial iteration, and each time a new node is encountered, we can add it to $V_{n,k}(j+1)$ if its previously-encountered incoming neighbor belongs to $V_{n,k}(j)$ for some j. (By encountering a new node u, we mean incrementing $r_v(u)$ for the first time; note that $r_v(u)$ is incremented only if u's incoming neighbor w is chosen as v^*, which in turn occurs only if $r_v(w)$ is nonzero, which means w has been previously encountered.) Next, observe that $\hat{\pi}_k(u)$ is not updated until u is first encountered, at which point we can check if $u \in V_{n,k}(j)$ for some $j \leq l$; if it is, $\hat{\pi}^l_k(u)$ can be updated each time $\hat{\pi}_k(u)$ is updated. Adding nodes to $V_{n,k}(j+1)$ has complexity that scales with that of updating r_v, while updating $\hat{\pi}^l_k$ has complexity that scales with that of updating $\hat{\pi}_k$. Hence, constructing $\hat{\pi}^l_k$ has complexity bounded by the `Approx-PageRank` complexity. In other words, we have

$$\mathbb{E}\left[C^{(3)}_{Alg3.1}\middle|\Omega_n\right] = O\left(\mathbb{E}[|K_n||\Omega_n]n\log n\right).$$

Next, let $C^{(4)}_{Alg3.1}$ denote the complexity of computing $\hat{\tilde{\pi}}_v(k)\ \forall\ k \in K_n, v \in V_n \setminus K_n$. This has complexity $O(n|K_n|)$, i.e.

$$\mathbb{E}\left[C^{(4)}_{Alg3.1}\middle|\Omega_n\right] = O\left(\mathbb{E}[|K_n||\Omega_n]n\right). \tag{B.70}$$

Next, let $C^{(5)}_{Alg3.1}$ denote the complexity of running `Approx-Page-Rank`(v, ε_1) for any $v \in V_n \setminus K_n$ satisfying $\hat{\tilde{\pi}}_v(K_n) < g_n(\varepsilon/4)$. We first observe that, by (B.85), $\hat{\tilde{\pi}}_v(K_n) < g_n(\varepsilon/4)$ implies

$$\tilde{\pi}_v(K_n) < g_n(\varepsilon/4) + \frac{|K_n|\varepsilon_2}{\alpha_n^2} = \frac{3}{2}g_n(\varepsilon/4),$$

where we have also used ε_2 as given in the statement of the theorem. Next, as in the argument leading to (B.69), we have

$$\frac{3}{2}g_n(\varepsilon/4) < \frac{3}{2}\frac{\alpha_n}{\varepsilon/4} = \frac{6\alpha_n}{\varepsilon}, \quad g_n(\varepsilon/14) > \frac{\alpha_n(1-(\varepsilon/7))}{\varepsilon/7} = \frac{\alpha_n(7-\varepsilon)}{\varepsilon}.$$

Hence, by assumption $\varepsilon < 1$, we have $\frac{3}{2}g_n(\varepsilon/4) < g_n(\varepsilon/14)$. In other words, we have shown $\hat{\tilde{\pi}}_v(K_n) < g_n(\varepsilon/4) \Rightarrow \tilde{\pi}_v(K_n) < g_n(\varepsilon/14)$. Therefore, by the argument leading to (B.81),

$$\hat{\tilde{\pi}}_v(K_n) < g_n(\varepsilon/4) \Rightarrow \left\| \pi_v - \left(\alpha_n e_v^{\mathsf{T}} + \frac{\sum_{k\in K_n} \tilde{\pi}_v(k)\pi_k}{\alpha_n + (1-\alpha_n)\tilde{\pi}_v(K_n)} \right) \right\|_1 > \frac{\varepsilon}{14}. \tag{B.71}$$

Note that the right side of (B.71) is the event $B_v(K_n, \varepsilon/14)$ defined in (3.2). Hence, the number of $v \in V_n \setminus K_n$ for which `Approx-PageRank`(v, ε_1) is run in Algorithm 3.1 satisfies

$$\begin{aligned}\mathbb{E}\left[\left| \left\{ v \in V_n \setminus K_n : 1\left(\hat{\tilde{\pi}}_v(K_n) < g_n(\varepsilon/4) \right) \right\} \right| \Big| \Omega_n \right] \\ \leq \mathbb{E}\left[|\{v \in V_n \setminus K_n : B_v(K_n, \varepsilon/14) \text{ holds}\}| | \Omega_n \right].\end{aligned}$$

On the other hand, by the argument in the analysis of $C^{(1)}_{Alg3.1}$, the complexity of running `Approx- PageRank`(v, ε_1) is $O(n \log n)$ when Ω_n holds. Combining arguments, we obtain

$$\mathbb{E}\left[C^{(5)}_{Alg3.1} \Big| \Omega_n \right] = O\left(\mathbb{E}\left[|\{v \in V_n \setminus K_n : B_v(K_n, \varepsilon/14) \text{ holds}\}| | \Omega_n \right] n \log n \right). \tag{B.72}$$

Finally, let $C^{(6)}_{Alg3.1}$ denote the complexity of computing $\hat{\pi}_v$ for all $v \in V_n \setminus K_n$ s.t. $\hat{\tilde{\pi}}_v(K_n) \geq g_n(\varepsilon/4)$. Here we multiply two matrices: the first has dimension $O(n) \times |K_n|$ and contains

$$\left\{ \hat{\tilde{\pi}}_v(k) : v \in V_n \setminus K_n \text{ s.t. } \hat{\tilde{\pi}}_v(K_n) \geq g_n(\varepsilon/4), k \in K_n \right\},$$

and the second has dimension $|K_n| \times n$ and contains rows $\{\hat{\pi}^l_k\}_{k\in K_n}$. We may bound the complexity of this multiplication as n times the number of nonzero elements of the latter matrix. Towards this end, recall that $\hat{\pi}^l_k(u)$ is nonzero only if $u \in V_{n,k}(l)$ and $\hat{\pi}_k(u)$ is nonzero, so the number of nonzero elements of $\hat{\pi}^l_k$ is bounded by $|V_{n,k}(l)|$, which we can bound as

$$\begin{aligned}\mathbb{E}\left[|V_{n,k}(l)| | \Omega_n \right] &\leq \mathbb{E}\left[|V_{n,k}(l)| \Big| \max_{k\in K_n} D_k \leq D_{\max}, \Omega_n \right] + n\mathbb{P}\left[\max_{k\in K_n} D_k > D_{\max} \Big| \Omega_n \right] \\ &= \mathbb{E}\left[|V_{n,k}(l)| \Big| \max_{k\in K_n} D_k \leq D_{\max}, \Omega_n \right] + O\left(n^{1-\delta'} \right), \end{aligned} \tag{B.73}$$

where the equality holds by the assumption in the statement of the theorem. To further bound the remaining expectation, we use an argument from Appendix B.2.4 (we describe this briefly and refer the reader to Appendix B.2.4 for further details). The argument is as follows. After $k \in K_n$ is first encountered during the graph construction and $V_{n,k}(l)$ is being constructed, we can simultaneously construct a tree of l generations, adding a new node to this tree each time an instub is sampled for pairing with an outstub belonging to

some $u \in V_{n,k}(l)$. By construction, $|V_{n,k}(l)|$ will be upper bounded by the number of nodes in this tree. Also, the number of nodes in this tree will have the same distribution as a tree constructed via Algorithm B.2 in Appendix B.1, which, with slight modification of (B.48) in Appendix B.1, satisfies $D_{\max}\sum_{j=1}^{l}\zeta_n^j$ (here ζ_n is defined at the start of Appendix B.2.4). To summarize, we have argued $\mathbb{E}_n\left[|V_{n,k}(l)|\right] \leq D_{\max}\sum_{j=1}^{l}\zeta_n^j$, which further implies

$$\begin{aligned}
&\mathbb{E}\left[|V_{n,k}(l)|\middle|\max_{k\in K_n} D_k \leq D_{\max}, \Omega_n\right] \\
&\quad= \frac{1}{\mathbb{P}[\max_{k\in K_n} D_k \leq D_{\max}, \Omega_n]}\mathbb{E}\left[1\left(\max_{k\in K_n} D_k \leq D_{\max}, \Omega_n\right)\mathbb{E}_n\left[|V_{n,k}(l)|\right]\right] \\
&\quad\leq \frac{1}{\mathbb{P}[\max_{k\in K_n} D_k \leq D_{\max}, \Omega_n]}\mathbb{E}\left[1\left(\max_{k\in K_n} D_k \leq D_{\max}, \Omega_n\right)D_{\max}\sum_{j=1}^{l}\zeta_n^j\right] \\
&\quad= D_{\max}\sum_{j=1}^{l}\mathbb{E}\left[\zeta_n^j\middle|\max_{k\in K_n} D_k \leq D_{\max}, \Omega_n\right] = O\left(\zeta^l\right) = O\left(n^{1/\rho}\right), \qquad \text{(B.74)}
\end{aligned}$$

where the penultimate equality uses $D_{\max} = O(1)$, the argument of (B.52) in Appendix B.2.4, and $\zeta > 1$; the final equality uses (B.88) from Appendix B.5. Hence, by (B.73) and (B.74),

$$\mathbb{E}\left[|V_{n,k}(l)||\Omega_n\right] = O\left(n^{1/\rho} + n^{1-\delta'}\right) = O\left(n^{\max\{1/\rho, 1-\delta'\}}\right).$$

Recalling that $|V_{n,k}(l)|$ bounds the number of nonzeros of $\hat{\pi}_k^l$, that $C_{Alg3.1}^{(6)}$ is bounded by n times the number of nonzeros of $\{\hat{\pi}_k^l\}_{k\in K_n}$, we obtain

$$\mathbb{E}\left[C_{Alg3.1}^{(6)}\middle|\Omega_n\right] = O\left(\mathbb{E}[|K_n||\Omega_n]n^{1+\max\{1/\rho, 1-\delta'\}}\right).$$

Finally, since $\mathbb{E}[|K_n|] = O(n^\kappa)$ by Assumption 3.2 and $\mathbb{P}[\Omega_n^C] = O(n^{-\delta})$ by Assumption 3.1,

$$\mathbb{E}[|K_n||\Omega_n] = \frac{\mathbb{E}[K_n 1(\Omega_n)]}{\mathbb{P}[\Omega_n]} \leq \frac{\mathbb{E}[K_n]}{\mathbb{P}[\Omega_n]} = O\left(n^\kappa\right), \qquad \text{(B.75)}$$

and so we ultimately obtain

$$\mathbb{E}\left[C_{Alg3.1}^{(6)}\middle|\Omega_n\right] = O\left(n^{1+\kappa+\max\{1/\rho, 1-\delta'\}}\right). \qquad \text{(B.76)}$$

Now because $C_{Alg3.1} = \sum_{i=1}^{6} C_{Alg3.1}^{(i)}$, we have

$$\mathbb{E}\left[C_{Alg3.1}|\Omega_n\right] = \max\left\{\sum_{i=1}^{5}\mathbb{E}\left[C_{Alg3.1}^{(i)}\middle|\Omega_n\right], \mathbb{E}\left[C_{Alg3.1}^{(6)}\middle|\Omega_n\right]\right\} \qquad \text{(B.77)}$$

Using the bounds derived above for $\{\mathbb{E}[C^{(i)}_{Alg3.1}|\Omega_n]\}_{i=1}^5$, we have

$$\begin{aligned}\sum_{i=1}^{5}\mathbb{E}\left[C^{(i)}_{Alg3.1}\middle|\Omega_n\right] &= O\Big(\Big(\mathbb{E}\left[|\{v\in V_n\setminus K_n : B_v(K_n,\varepsilon/14)\text{ holds}\}||\Omega_n\right] \\ &\qquad + \mathbb{E}[|K_n||\Omega_n]\Big)n(\log n)^3\Big) \\ &= O\left(\mathbb{E}[\Delta(K_n,\varepsilon/14)|\Omega_n]n(\log n)^3\right) = O\left(\mathbb{E}[\Delta(K_n,\varepsilon/14)]n(\log n)^3\right),\end{aligned} \tag{B.78}$$

where the final line holds as in (B.75). (B.76), (B.77), and (B.78) complete the proof.

We now turn to the accuracy guarantee. For this, first note that $\hat{\pi}_v$ is computed via `Approx- PageRank`(v,ε_1) whenever $v\in K_n$ or $v\in V_n\setminus K_n$, $\hat{\tilde{\pi}}_v(K_n) < g_n(\varepsilon_1)$. In both cases, Lemma B.11 ensures $\|\hat{\pi}_v-\pi_v\|_1\le\varepsilon_1=\varepsilon/4<\varepsilon$. Thus, it only remains to show $\|\hat{\pi}_v-\pi_v\|_1\le\varepsilon$ when $v\in V_n\setminus K_n$ and $\hat{\tilde{\pi}}_v(K_n)\ge g_n(\varepsilon_1)$, in which case we instead compute $\hat{\pi}_v$ as

$$\hat{\pi}_v = \alpha_n e_v^{\mathsf{T}} + \frac{\sum_{k\in K_n}\hat{\tilde{\pi}}_v(k)\hat{\pi}^l_k}{\alpha_n+(1-\alpha_n)\hat{\tilde{\pi}}_v(K_n)}. \tag{B.79}$$

We first note that by (B.34) and the definition of $\hat{\tilde{\pi}}_v$, we have

$$\tilde{\pi}_v(K_n) = \frac{\alpha_n\mu_v(K_n)}{1-(1-\alpha_n)\mu_v(K_n)} \ge \frac{\alpha_n\hat{\mu}_v(K_n)}{1-(1-\alpha_n)\hat{\mu}_v(K_n)} = \hat{\tilde{\pi}}_v(K_n) \tag{B.80}$$

where the inequality holds because the left side is increasing in $\mu_v(K_n)$ and since $\mu_v(K_n)\ge\hat{\mu}_v(K_n)$ by Lemma B.12. Thus, $\hat{\tilde{\pi}}_v(K_n)\ge g_n(\varepsilon_1)$ implies $\tilde{\pi}_v(K_n)\ge g_n(\varepsilon_1)$ as well; furthermore, some simple algebra, along with (B.30) in Appendix B.2.2, shows

$$\tilde{\pi}_v(K_n)\ge g_n(\varepsilon_1) \Leftrightarrow \left\|\pi_v - \left(\alpha_n e_v^{\mathsf{T}} + \frac{\sum_{k\in K_n}\tilde{\pi}_v(k)\pi_k}{\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n)}\right)\right\|_1 \le \varepsilon_1 = \frac{\varepsilon}{4}, \tag{B.81}$$

where the equality holds by the statement of the theorem. Then

$$\begin{aligned}\|\pi_v-\hat{\pi}_v\|_1 &\le \left\|\pi_v - \left(\alpha_n e_v^{\mathsf{T}} + \frac{\sum_{k\in K_n}\tilde{\pi}_v(k)\pi_k}{\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n)}\right)\right\|_1 \\ &\quad + \left\|\frac{\sum_{k\in K_n}\tilde{\pi}_v(k)}{\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n)}\left(\pi_k-\hat{\pi}^l_k\right)\right\|_1 \\ &\quad + \left\|\frac{\sum_{k\in K_n}\hat{\pi}^l_k}{\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n)}\left(\tilde{\pi}_v(k)-\hat{\tilde{\pi}}_v(k)\right)\right\|_1 \\ &\quad + \left\|\sum_{k\in K_n}\hat{\tilde{\pi}}_v(k)\hat{\pi}^l_k\left(\frac{1}{\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n)}-\frac{1}{\alpha_n+(1-\alpha_n)\hat{\tilde{\pi}}_v(K_n)}\right)\right\|_1 \\ &\le \frac{\varepsilon}{4} + \max_{k\in K_n}\left\|\pi_k-\hat{\pi}^l_k\right\|_1 + \frac{\tilde{\pi}_v(K_n)-\hat{\tilde{\pi}}_v(K_n)}{\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n)}\end{aligned} \tag{B.82}$$

$$+\frac{(1-\alpha_n)\hat{\tilde{\pi}}_v(K_n)(\tilde{\pi}_v(K_n)-\hat{\tilde{\pi}}_v(K_n))}{(\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n))(\alpha_n+(1-\alpha_n)\hat{\tilde{\pi}}_v(K_n))},$$

where the first inequality is the triangle inequality, and the second uses (B.81) for the first term and the triangle inequality for the other terms (in bounding the final two terms, we have also used the fact that for each $k \in K_n$, $\|\hat{\pi}_k^l\|_1 \leq 1$ and $\tilde{\pi}_v(k) \geq \hat{\tilde{\pi}}_v(k)$). We next derive bounds on the final three terms in (B.82). For the second term, we observe

$$\begin{aligned}\max_{k\in K_n}\left\|\pi_k-\hat{\pi}_k^l\right\|_1 &\leq \max_{k\in K_n}\|\pi_k-\hat{\pi}_k\|_1+\max_{k\in K_n}\left\|\hat{\pi}_k-\hat{\pi}_k^l\right\|_1 \\ &\leq \varepsilon_1+\max_{k\in K_n}\pi_k\left(V_n\setminus V_{n,k}(l)\right)\leq \varepsilon_1+\tau,\end{aligned} \tag{B.83}$$

where the first inequality is the triangle inequality, the second uses Lemma B.11 and the fact that $\hat{\pi}_k(v)=\hat{\pi}_k^l(v)$ for $v \in V_{n,k}(l)$ and $\hat{\pi}_k(v) \leq \pi_k(v), \hat{\pi}_k^l(v)=0$ for other v, and the third follows by the argument leading to (B.87) in Appendix B.5. For the fourth term in (B.82), first note that by $\alpha_n > 0$ and $\hat{\tilde{\pi}}_v(K_n)) \leq 1$,

$$\frac{(1-\alpha_n)\hat{\tilde{\pi}}_v(K_n)}{(\alpha_n+(1-\alpha_n)\hat{\tilde{\pi}}_v(K_n))}<1,$$

and so the final two terms in (B.82) can be bounded as

$$\begin{aligned}&\frac{\tilde{\pi}_v(K_n)-\hat{\tilde{\pi}}_v(K_n)}{\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n)}+\frac{(1-\alpha_n)\hat{\tilde{\pi}}_v(K_n)(\tilde{\pi}_v(K_n)-\hat{\tilde{\pi}}_v(K_n))}{(\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n))(\alpha_n+(1-\alpha_n)\hat{\tilde{\pi}}_v(K_n))} \\ &\quad\leq \frac{2(\tilde{\pi}_v(K_n)-\hat{\tilde{\pi}}_v(K_n))}{\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n)}\leq\frac{2(\tilde{\pi}_v(K_n)-\hat{\tilde{\pi}}_v(K_n))}{g_n(\varepsilon/4)},\end{aligned} \tag{B.84}$$

where for the second inequality, we have used $\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n) \geq \tilde{\pi}_v(K_n) \geq g_n(\varepsilon/4)$ by assumption. Furthermore, we note

$$\begin{aligned}\tilde{\pi}_v(K_n)-\hat{\tilde{\pi}}_v(K_n)&=\frac{\alpha_n\mu_v(K_n)}{1-(1-\alpha_n)\mu_v(K_n)}-\frac{\alpha_n\hat{\mu}_v(K_n)}{1-(1-\alpha_n)\hat{\mu}_v(K_n)} \\ &=\frac{\alpha_n(\mu_v(K_n)-\hat{\mu}_v(K_n))}{(1-(1-\alpha_n)\mu_v(K_n))(1-(1-\alpha_n)\hat{\mu}_v(K_n))}\leq\frac{|K_n|\varepsilon_2}{\alpha_n^2},\end{aligned} \tag{B.85}$$

where we used (B.80), Lemma B.12, and the fact that, by (B.34),

$$\mu_v(K_n)=\frac{\tilde{\pi}_v(K_n)}{\alpha_n+(1-\alpha_n)\tilde{\pi}_v(K_n)}\leq 1,$$

and a similar argument implies $\hat{\mu}_v(K_n) \leq 1$. Combining (B.82), (B.83), (B.84), and (B.85), we have shown that when $v \in\notin K_n, \hat{\tilde{\pi}}_v(K_n) \geq g_n(\varepsilon_1)$, i.e. when $\hat{\pi}_v$ is computed via (B.79),

$$\|\pi_v-\hat{\pi}_v\|_1\leq\frac{\varepsilon}{4}+\varepsilon_1+\tau+\frac{2|K_n|\varepsilon_2}{\alpha_n^2 g_n(\varepsilon/4)}\leq\varepsilon,$$

where the final inequality holds by our assumptions on $\varepsilon_1, \varepsilon_2, \tau$. This completes the proof.

B.4.3 Precomputation variant of Algorithm 3.1

We begin by analyzing the precomputation variant's accuracy. For this, we first note that the estimate of $\pi_v, v \notin K_n$ is computed in the same manner as in Algorithm 3.1, so the accuracy guarantee of Theorem 3.2 holds for such v. However, for $k \in K_n$, the precomputation variant instead returns $\hat{\pi}_k^l$, so the accuracy guarantee does not apply. Nevertheless, by (B.83),

$$\left\|\pi_k - \hat{\pi}_k^l\right\|_1 \leq \varepsilon_1 + \tau \leq \varepsilon/2,$$

where the second inequality holds by the assumptions in Theorem 3.2. Hence, all estimates returned by the variant satisfy the accuracy guarantee claimed in the main text. We next consider the space complexity for storing $\{\hat{\pi}_k^l\}_{k\in K_n}$ and $\{\hat{\mu}_k(k)\}_{u\in V_n, k\in K_n}$ from the offline stage. Trivially, $nnz(\hat{\pi}_k^l) \leq n$ and $nnz(\{\hat{\mu}_k(k)\}_{u\in V_n}) \leq n \ \forall \ k \in K_n$, where $nnz(x)$ denote the number of nonzero elements of the vector x. Hence, the overall storage is at most $2n|K_n|$, which is $O(n^{1+\kappa})$ in expectation. Finally, we consider the complexity of running the online stage for $v^* \sim V_n$ uniformly. If $v^* \in K_n$, no computation is required, so this complexity is negligible. If $v^* \in V_n \setminus K_n$, this complexity is simply $1/|V_n \setminus K_n|$ times the complexity of running Lines 6-8 $\forall \ v \in V_n \setminus K_n$ in Algorithm 3.1. By the analysis in Appendix B.4.2 (specifically, by (B.70), (B.72), and (B.76)), this latter quantity is

$$\begin{aligned}
\sum_{i=4}^{6} \mathbb{E}\left[C_{Alg1}^{(i)} \middle| \Omega_n\right] &= O(\mathbb{E}[|K_n||\Omega_n]n) \\
&\quad + O\left(\mathbb{E}\left[|\{v \in V_n \setminus K_n : B_v(K_n, \varepsilon/14) \text{ holds}\}||\Omega_n\right] n \log n\right) \\
&\quad + O\left(n^{1+\kappa+\max\{1/\rho, 1-\delta'\}}\right) \\
&= O\left(\max\left\{\Delta(K_n, \varepsilon/14) n \log n, n^{1+\kappa+\max\{1/\rho,1-\delta'\}}\right\}\right).
\end{aligned}$$

Hence, with $|V_n \setminus K_n| = O(n)$ in expectation, the complexity the online stage for $v^* \sim V_n$ is

$$O\left(\max\left\{\Delta(K_n, \varepsilon/14) \log n, n^{\kappa+\max\{1/\rho,1-\delta'\}}\right\}\right).$$

B.5 Proof of Proposition 3.1

First, note that $\forall \ n \in \mathbb{N}$, $\forall \ l \in \mathbb{N}$, and $\forall \ l' \leq l$, we have (a.s.)

$$\pi_s(V_{n,s}(l)) \geq \alpha_n \sum_{j=0}^{l} (1-\alpha_n)^j \left(e_s^\mathsf{T} P^j 1_n\right) = \alpha_n \sum_{j=0}^{l} (1-\alpha_n)^j \geq 1 - (1-\alpha_n)^{l'}, \tag{B.86}$$

where the first inequality follows from (1.1) and by definition of $V_{n,s}(l)$, and the first equality holds since P^j is row stochastic (the remaining steps are simple manipulations). Therefore,

when $\alpha_n = \frac{\rho \log(1/\tau) \log \zeta}{\log n}$, we can define $c = \rho \log \zeta$ and use (B.86) to write

$$\liminf_{n\to\infty} \pi_s \left(V_{n,s} \left(\left\lceil \frac{\log(1/\tau)}{\alpha_n} \right\rceil \right) \right) \geq 1 - \lim_{n\to\infty} \left(1 + \frac{\log(\tau)}{\log(n)/c} \right)^{\log(n)/c} = 1 - \tau \ a.s.,$$

which is the desired bound. If instead $\alpha_n = \alpha$ is a constant, we have more simply

$$\liminf_{n\to\infty} \pi_s \left(V_{n,s} \left(\left\lceil \frac{\log(1/\tau)}{\log(1/(1-\alpha))} \right\rceil \right) \right) \geq 1 - (1-\alpha)^{\frac{\log(1/\tau)}{\log(1/(1-\alpha))}} = 1 - \tau \ a.s.. \tag{B.87}$$

Next, to bound the size of $V_{n,s}(l)$, we use the analysis of Appendix B.2.4. First, for $l \in \mathbb{N}$, the argument preceding (B.50) in Appendix B.2.4 implies $|V_{n,s}(l)| \leq \sum_{j=0}^{l} \hat{Z}_j$, where $\hat{Z}_j$ is defined in (B.46). Furthermore, by (B.52) in Appendix B.2.4, we have for $j \in \mathbb{N}$,

$$\mathbb{E}\left[\hat{Z}_j \middle| \Omega_n \right] = O\left(\zeta^{j-1} \right),$$

while $\hat{Z}_0 = 1$ by definition. Combining gives for $l \in \mathbb{N}$,

$$\mathbb{E}\left[V_{n,s}(l) | \Omega_n \right] = O\left(1 + \sum_{j=0}^{l-1} \zeta^j \right) = O\left(\zeta^l \right).$$

Therefore, when $\alpha_n = \frac{\rho \log(1/\tau) \log \zeta}{\log n}$, we have

$$\mathbb{E}\left[\left| V_{n,s} \left(\left\lceil \frac{\log(1/\tau)}{\alpha_n} \right\rceil \right) \right| \middle| \Omega_n \right] = O\left(\zeta^{\log(1/\tau)/\alpha_n} \right) = O\left(\zeta^{\log_\zeta(n^{1/\rho})} \right) = O\left(n^{1/\rho} \right). \tag{B.88}$$

Similarly, if $\alpha_n = \alpha$ is a constant,

$$\mathbb{E}\left[\left| V_{n,s} \left(\left\lceil \frac{\log(1/\tau)}{\log(1/(1-\alpha))} \right\rceil \right) \right| \middle| \Omega_n \right] = O\left(\zeta^{\log(1/\tau)/\log(1/(1-\alpha))} \right) = O(1).$$

B.6 Proof of Proposition 3.3

Assume $n \geq v$, so that π_v, P_v are defined as in Section 3.2.2. For such n, we claim that for any realization of G_n and any $i \in \mathbb{N}$,

$$P_v^i = \alpha_n 1_n e_v^\mathsf{T} \sum_{j=0}^{i-1} (1-\alpha_n)^j P^j + (1-\alpha_n)^i P^i. \tag{B.89}$$

We prove (B.89) inductively: it holds by definition for $i = 1$, and if it holds for general i,

$$\begin{aligned} P_v^{i+1} &= P_v \left(\alpha_n 1_n e_v^\mathsf{T} \sum_{j=0}^{i-1} (1-\alpha_n)^j P^j + (1-\alpha_n)^i P^i \right) \\ &= \alpha_n (P_v 1_n) e_v^\mathsf{T} \sum_{j=0}^{i-1} (1-\alpha_n)^j P^j + \left(\alpha_n 1_n e_v^\mathsf{T} + (1-\alpha_n) P \right) (1-\alpha_n)^i P^i \end{aligned}$$

$$= \alpha_n 1_n e_v^\mathsf{T} \sum_{j=0}^{i} (1-\alpha_n)^j P^j + (1-\alpha_n)^{i+1} P^{i+1},$$

where the first equality holds by the inductive hypothesis, the second uses the definition of P_v, and the third uses row stochasticity of P_v. We next write

$$\begin{aligned} \pi_v = \pi_v P_v^m &= \alpha_n e_v^\mathsf{T} \sum_{j=0}^{m-1} (1-\alpha_n)^j P^j + (1-\alpha_n)^m \pi_v P^m \\ &= \pi_v P^m + \alpha_n e_v^\mathsf{T} \sum_{j=0}^{m-1} (1-\alpha_n)^j P^j + ((1-\alpha_n)^m - 1)\pi_v P^m, \end{aligned} \tag{B.90}$$

where the equalities follow by global balance ($\pi_v = \pi_v P_v$), (B.89) and the fact π_v sums to 1, and adding/subtracting $\pi_v P^m$, respectively. Next, we have for any $w \in V_n$,

$$\begin{aligned} \|\pi - \pi_v\|_1 &\le \|\pi - e_w P^m\|_1 + \|e_w P^m - \pi_v P^m\|_1 + \|\pi_v P^m - \pi_v\|_1 \\ &= \|\pi - e_w P^m\|_1 + \|e_w P^m - \pi_v P^m\|_1 + 2(1 - (1-\alpha_n)^m), \end{aligned} \tag{B.91}$$

where the inequality is the triangle inequality, and the equality follows by (B.90) and the fact that P is row stochastic. Again using the triangle inequality, as well as the fact that π_v sums to 1 and convexity of $\|\cdot\|_1$, we can write

$$\begin{aligned} \|e_w P^m - \pi_v P^m\|_1 &\le \|e_w P^m - \pi\|_1 + \left\| \sum_{w' \in V_n} \pi_v(w') \left(e_{w'}^\mathsf{T} P^m - \pi \right) \right\|_1 \\ &\le \|e_w P^m - \pi\|_1 + \sum_{w' \in V_n} \pi_v(w') \left\| e_{w'}^\mathsf{T} P^m - \pi \right\|_1 \\ &= \|e_w P^m - \pi\|_1 + \max_{w' \in V_n} \left\| e_{w'}^\mathsf{T} P^m - \pi \right\|_1 \le 2 \max_{w' \in V_n} \left\| e_{w'}^\mathsf{T} P^m - \pi \right\|_1 . \end{aligned} \tag{B.92}$$

We may then combine (B.91) and (B.92) to obtain

$$\|\pi_v - \pi\|_1 \le 3 \max_{w \in V_n} \|e_w P^m - \pi\|_1 + 2(1 - (1-\alpha_n)^m). \tag{B.93}$$

Furthermore, by Bernoulli's inequality, $m = \Theta(\log n)$, and $\alpha_n = o(1/\log_n)$,

$$1 \ge (1-\alpha_n)^m \ge 1 - \alpha_n m \xrightarrow[n\to\infty]{} 1.$$

Thus, letting $n \to \infty$ in (B.93), $\|\pi_v - \pi\|_1 \to 0$ by assumption on $\max_{w \in V_n} \|e_w P^m - \pi\|_1$.

B.7 Experimental details

B.7.1 Dataset details

The following table shows details of the datasets used for experiments in Section 3.6. All datasets are available from the Stanford Network Analysis Platform [43]. The α_n values shown are used for all experiments conducted on the corresponding graph. We note that,

while these are smaller than α_n values typically used, they are the same order of magnitude ($\alpha_n = 0.15$ is a common choice in the literature). Finally, we note that the datasets with prefix web- are partial web crawls; those with prefix soc- are social networks.

Dataset	n	L_n	$\alpha_n = 1/\log n$
soc-LiveJournal1	4847571	68993773	0.065
soc-pokec	1632803	30622564	0.070
web-Google	875713	5105039	0.073
web-BerkStan	685230	7600595	0.074
web-Stanford	281903	2312497	0.080

B.7.2 Scheme to bound estimation error

To bound $\|\pi_v - (\alpha_n e_v^\mathsf{T} + \sum_{k\in K_n} \beta_v(k)\pi_k)\|_1$, where $\beta_v(k)$ are defined in (3.6), we employ a power iteration scheme: we initialize $x_v^{(0)} = e_v^\mathsf{T}$, and given $x_v^{(i-1)}$ for $i \geq 1$, we set

$$x_v^{(i)} = \alpha_n e_v^\mathsf{T} + (1-\alpha_n) x_v^{(i-1)} \tilde{P},$$

where $\tilde{P}$ is defined in (3.4). We claim

$$x_v^{(i)} = \alpha_n \mu_v^{(i-1)} + (1-\alpha_n)^i e_v^\mathsf{T} \tilde{P}^i, \tag{B.94}$$

where $\mu_v^{(i-1)} = e_v^\mathsf{T} \sum_{j=0}^{i-1} (1-\alpha_n)^j \tilde{P}^j$ (as in Appendix B.1-B.2). (B.94) is easily proven inductively: the base of induction holds by definition; assuming true for $i-1$, we have

$$\begin{aligned} x_v^{(i)} &= \alpha_n e_v^\mathsf{T} + (1-\alpha_n)\left(\alpha_n \mu_v^{(i-2)} + (1-\alpha_n)^{i-1} e_v^\mathsf{T} \tilde{P}^{i-1}\right)\tilde{P} \\ &= \alpha_n e_v^\mathsf{T} + \alpha_n \sum_{j=1}^{i-1} (1-\alpha_n)^j \tilde{P}^j + (1-\alpha_n)^i e_v^\mathsf{T} \tilde{P}^i = \alpha_n \mu_v^{(i-1)} + (1-\alpha_n)^i e_v^\mathsf{T} \tilde{P}^i \end{aligned}$$

as claimed. Now by Lemma B.2 in Appendix B.1, for any $i \in \mathbb{N}$ we obtain the following bound:

$$\left\| \pi_v - \left(\alpha_n e_v^\mathsf{T} + \sum_{k\in K_n} \beta_v(k)\pi_k \right) \right\|_1 \leq \alpha_n \mu_v^{(i-1)}(V_n \setminus K_n) + (1-\alpha_n)^i e_v^\mathsf{T} \tilde{P}^i e_{V_n \setminus K_n} - \alpha_n \tag{B.95}$$

$$= x_v^{(i)}(V_n \setminus K_n) - \alpha_n. \tag{B.96}$$

From this bound, we can prove two other claims from Section 3.5. First, we note

$$x_v^{(i)}(V_n \setminus K_n) = \alpha_n e_v^\mathsf{T} \sum_{j=0}^{i-1} (1-\alpha_n)^j \tilde{P}^j + (1-\alpha_n)^i e_v^\mathsf{T} \tilde{P}^i e_{V_n \setminus K_n} \leq = 1,$$

where the inequality follows since $\tilde{P}$ is nonnegative with row sums bounded by 1. Hence, from (B.96), the estimation error is bounded by $(1-\alpha_n)$ (as claimed in Section 3.5). Next, suppose $v \in V_{n,0}$, with $V_{n,0}$ given by (3.12). Then $e_v^\mathsf{T} \tilde{P}^j e_{V_n \setminus K_n} = 0$, so $x_v^{(i)}(V_n \setminus K_n) = \alpha_n$,

and the estimation error is zero (as claimed in Section 3.6). We can also bound the gap in the inequality (B.95): use (B.35) in Appendix B.2.2 and (B.95) to write

$$\begin{aligned}
&\left\|\pi_v - \left(\alpha_n e_v^\mathsf{T} + \sum_{k\in K_n}\beta_v(k)\pi_k\right)\right\|_1 - \left(x_v^{(i)}(V_n\setminus K_n) - \alpha_n\right) \\
&\quad= \left(\alpha_n e_v^\mathsf{T}\sum_{j=0}^{\infty}(1-\alpha_n)^i\tilde{P}^i e_{V_n\setminus K_n} - \alpha_n\right) - \left(x_v^{(i)}(V_n\setminus K_n) - \alpha_n\right) \\
&\quad= \alpha_n\mu_v^{(i-1)}(V_n\setminus K_n) + \alpha_n e_v^\mathsf{T}\sum_{j=i}^{\infty}(1-\alpha_n)^i\tilde{P}^i e_{V_n\setminus K_n} - x_v^{(i)}(V_n\setminus K_n) \\
&\quad= \alpha_n e_v^\mathsf{T}\sum_{j=i}^{\infty}(1-\alpha_n)^i\tilde{P}^i e_{V_n\setminus K_n} - (1-\alpha_n)^i e_v^\mathsf{T}\tilde{P}^i e_{V_n\setminus K_n} \geq -(1-\alpha_n)^i,
\end{aligned}$$

where the inequality holds by dropping a nonnegative term and since $e_v^\mathsf{T}\tilde{P}^i e_{V_n\setminus K_n} \leq 1$. Hence, if we let $i^* \geq \log_{(1-\alpha_n)}(tol)$ for some desired tolerance tol, the bound $x_v^{(i^*)}(V_n\setminus K_n) - \alpha_n$ is tight within additive error tol. (For all experiments, we set $tol = 0.05$.)

To bound average error across $V_n\setminus K_n$, we instead use

$$x_{V_n\setminus K_n}^{(0)} = \frac{e_{V_n\setminus K_n}^\mathsf{T}}{|V_n\setminus K_n|}, \quad x_{V_n\setminus K_n}^{(i)} = \alpha_n\frac{e_{V_n\setminus K_n}^\mathsf{T}}{|V_n\setminus K_n|} + (1-\alpha_n)x_{V_n\setminus K_n}^{(i-1)}\tilde{P}.$$

Note $x_{V_n\setminus K_n}^{(i)} = \frac{1}{|V_n\setminus K_n|}\sum_{v\in V_n\setminus K_n}x_v^{(i)}$ when $i=0$ by definition; assuming true for general $i-1$,

$$\begin{aligned}
x_{V_n\setminus K_n}^{(i)} &= \alpha_n\frac{e_{V_n\setminus K_n}^\mathsf{T}}{|V_n\setminus K_n|} + (1-\alpha_n)\left(\frac{1}{|V_n\setminus K_n|}\sum_{v\in V_n\setminus K_n}x_v^{(i-1)}\right)\tilde{P} \\
&= \frac{1}{|V_n\setminus K_n|}\sum_{v\in V_n\setminus K_n}\left(\alpha_n e_v^\mathsf{T} + (1-\alpha_n)x_v^{(i-1)}\tilde{P}\right) = \frac{1}{|V_n\setminus K_n|}\sum_{v\in V_n\setminus K_n}x_v^{(i)},
\end{aligned}$$

i.e. $x_{V_n\setminus K_n}^{(i)} = \frac{1}{|V_n\setminus K_n|}\sum_{v\in V_n\setminus K_n}x_v^{(i)}\ \forall\ i\in\mathbb{N}$. It follows from above that

$$\frac{1}{|V_n\setminus K_n|}\sum_{v\in V_n\setminus K_n}\left\|\pi_v - \left(\alpha_n e_v^\mathsf{T} + \sum_{k\in K_n}\beta_v(k)\pi_k\right)\right\|_1 \leq x_{V_n\setminus K_n}^{(i)}(V_n\setminus K_n) - \alpha_n,$$

which is the average error bound we compute for Figures 3.2a, 3.2b, and 3.3. The argument above also implies this bound is tight within tol when $i^* \geq \log_{(1-\alpha_n)}(tol)$.

B.7.3 Details on Figure 3.1 experiment

In addition to the histograms of l_1 error shown in Figure 3.1a, we include a more detailed set of plots for the same experiment. Specifically, we estimate the error $|\alpha_n e_v^\mathsf{T}(w) + \sum_{k\in K_n}\beta_v(k)\pi_k(w)|$ as $x_v^{(i)}(w) - \alpha_n 1(w=v)$ (where $x_v^{(i)}$ is defined in Appendix B.7.2), for each $w\in V_n\setminus K_n$, and for each v in a subset of $V_n\setminus K_n$ of size $\approx 10^4$. (These v were chosen

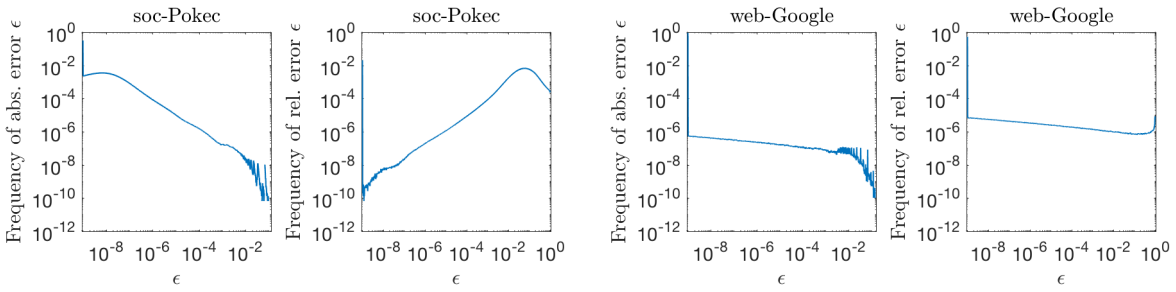


Figure B.5: Detailed error analyses.

uniformly from nodes with average error $\in (0.08, 0.25)$, which corresponds to the regime of linear decay in Figure 3.1.) We also estimate the relative error, i.e. the ratio of this absolute error to an estimate of $\pi_v(w)$, for the same set of (v, w). The estimate of π_v is computed using the same power iteration scheme in Appendix B.7.2, but replacing $\tilde{P}$ with P. Note this gives a lower bound on the true value of $\pi_v(w)$, thereby upper bounding relative error. Unfortunately, we cannot compute this relative error estimate when the estimate of $\pi_v(w)$ is zero; this occurred for only 10% of (v, w) pairs considered. Finally, for both absolute and relative error, we compute the number of error values lying in log-spaced bins and divide these values by n to estimate the frequency of each error value. (We add values lying beyond the first and last bin edges to the first and last bins, respectively.)

Results are shown for the soc-Pokec dataset at left in Figure B.5. (We note the spikes at left occur due to values lying beyond the first bin edge.) As an illustration for absolute error, the frequency of values above 10^{-3} was $\approx 10^{-5}$, i.e. the vast majority of nodes had estimated absolute error below 10^{-3}. To illustrate the relative error, the frequency of values above 0.2 was ≈ 0.09, i.e. over 90% of nodes had estimated relative error below 0.2. The results for web-Google are shown at right in Figure B.5. For absolute error, the frequency above 10^{-3} was again $\approx 10^{-5}$; for relative error, over 90% of nodes had error below 0.2.

B.7.4 Geometric interpretation of Theorem 3.1

In Figure B.6, we show a graphical representation of Theorem 3.1 similar to Figure 3.6 in Section 3.7.5 but using actual PPR vectors. For these plots, G_n is a DCM with in-degrees following a power law with exponent 2 and out-degrees generated as in Algorithm 3.4 from Section 3.7.3. The dots are projections of the n-dimensional vectors $\{\pi_v\}_{v \in V_n \setminus K_n}$ into 2D space; specifically, the v-th dot is at $(\pi_v(v_1), \pi_v(v_2))$, where v_1 is the node of highest in-degree and v_2 is the node of second-highest in-degree. Red and green, respectively, correspond to those v for which $B_v(K_n, \varepsilon)$ holds and fails, respectively, with K_n chosen as the $\sqrt{n}$ nodes of highest in-degree and $\varepsilon = 0.2$. Finally, the region outlined in blue is the convex hull of $\{(\pi_k(v_1), \pi_k(v_2))\}_{k \in K_n}$. Note that, as n grows, a larger fraction of dots fall near or within the blue outlined region, and the area of this region decreases as n grows, as in Figure 3.6. However, the dichotomy of green dots lying inside the region and red dots lying outside the region is much less clear than in Figure 3.6. This is in part because the projection $\pi_v \mapsto (\pi_v(v_1), \pi_v(v_2))$ is not l_1 distance-preserving. Instead, green and red dots exhibit a different distinction in Figure B.6: roughly speaking, red dots lie closer to the bottom left of each plot, while green dots lie closer to the top right. This is because $v_1, v_2 \in K_n$ by definition

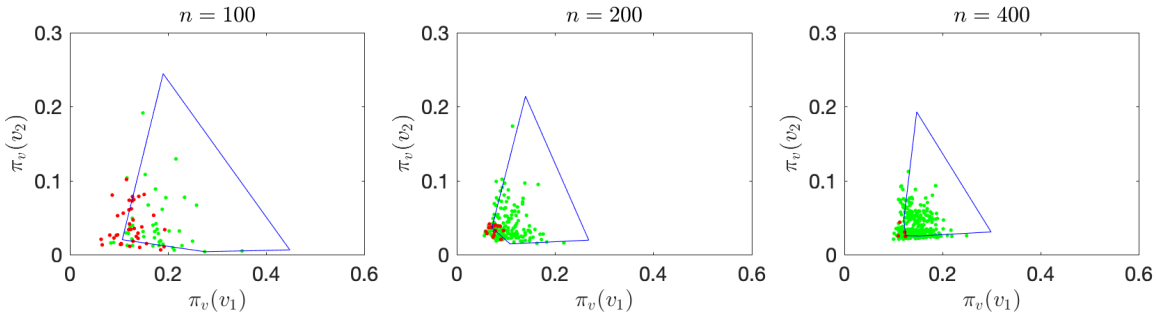


Figure B.6: An analogue Figure 3.6 in Section 3.7.5, but here using actual PPR vectors.

of v_1, v_2 and choice of K_n; hence, dots near the top right are "close" in the graph to at least two elements of K_n, which (at a high level) means their PPR vectors are well-approximated as linear combinations of $\{\pi_k\}_{k \in K_n}$.

B.8 Algorithmic comparisons

B.8.1 Nonnegative matrix factorization

As discussed in Section 3.5.2, our algorithm can be viewed as a variant of nonnegative matrix factorization (NMF). To explain this, we assume for simplicity that nodes are labeled such that $K_n = \{1, \dots, |K_n|\}$ and $B_v(K_n, \varepsilon/14)$ holds $\forall\ v \in \{|K_n| + 1, \dots, \Delta(K_n, \varepsilon/14)\}$. Thus, for $v \leq \Delta(K_n, \varepsilon/14)$, the estimate $\hat{\pi}_v$ is computed via `Approx-PageRank`, while for $v > \Delta(K_n, \varepsilon/14)$, $\hat{\pi}_v$ is computed as

$$\hat{\pi}_v = \alpha_n e_v^{\mathsf{T}} + \frac{\sum_{k \in K_n} \hat{\hat{\pi}}_v(k) \hat{\pi}_k}{\alpha_n + (1 - \alpha_n) \hat{\hat{\pi}}_v(K_n)}.$$

Thus, Algorithm 3.1 outputs $\alpha_n I + WH$, where $W \in \mathbb{R}^{n \times \Delta(K_n, \varepsilon/14)}$, $H \in \mathbb{R}^{\Delta(K_n, \varepsilon/14) \times n}$ satisfy

$$W(v,:) = \begin{cases} e_v^{\mathsf{T}}, & v \leq \Delta(K_n, \varepsilon/14) \\ \frac{[\hat{\hat{\pi}}_v(1) \ \cdots \ \hat{\hat{\pi}}_v(|K_n|) \ 0 \ \cdots \ 0]}{\alpha_n + (1-\alpha_n)\hat{\hat{\pi}}_v(K_n)}, & v > \Delta(K_n, \varepsilon/14) \end{cases},$$
$$H(v,:) = \hat{\pi}_v - \alpha_n e_v^{\mathsf{T}}.$$

In short, our algorithm computes matrices W and H such that

$$\|(\Pi_n - \alpha_n I) - WH\|_\infty < \varepsilon,$$

which is the NMF-like objective function discussed in Section 3.5.2.

As mentioned in Section 3.5.2, our algorithm offers several advantages over typical NMF algorithms. First, it is provably accurate (for general graphs) and provably efficient (for the DCM). Second, it is adaptive in terms of the dimensions of W and H: while standard NMF algorithms assume $W \in \mathbb{R}^{n \times r}, H \in \mathbb{R}^{r \times n}$ *a priori* for some r (typically $r \ll n$ to obtain a low rank estimate), our algorithm determines at runtime which $\pi_v, v \notin K_n$ cannot be approximated as linear combinations of $\{\pi_k\}_{k \in K_n}$ and adjusts r to account for this (ultimately yielding $r = \Delta(K_n, \varepsilon/14)$ as above). Additionally, off-the-shelf NMF algorithms

are difficult to adapt to our setting for several reasons. First, our objective function uses the non-differentiable norm $\|\cdot\|_\infty$, so we cannot compute the gradients needed for standard NMF. Second, even if our objective was differentiable, it involves the unknown matrix Π_n, again rendering gradient calculations impossible (typically, NMF aims to find W, H so as to minimize $\|X - WH\|$ for some *known* matrix X).

To overcome these issues, one could instead use the objective function

$$J(W,H) = \min_{W,H\geq 0} \tfrac{1}{2}\|\alpha_n I - WH(I-(1-\alpha_n)P)\|_F^2.$$

Here we have used the differentiable norm $\|\cdot\|_F$ and removed the unknown matrix Π_n from the objective. The form of this new objective function is motivated by (1.2), which shows $\Pi_n = \alpha_n(I-(1-\alpha_n)P)^{-1}$; hence, $J(W,H) = 0$ when $WH = \Pi_n$. With this objective function, the multiplicative update rule for NMF from [70] can be applied, which is

$$W \leftarrow W\nabla_W^-/\nabla_W^+, \quad H \leftarrow H\nabla_H^+/\nabla_H^-, \tag{B.97}$$

where the multiplication and division is elementwise, and where ∇_W^- and ∇_W^+ are the negative and positive parts of the gradient of J with respect to W (∇_H^-, ∇_H^+ are defined analogously). However, we claim that this method will perform worse than our algorithm. To see why, we define $Y = (I-(1-\alpha_n)P)$ and note

$$\begin{aligned}\nabla_W &= -\alpha_n Y^\mathsf{T} H^\mathsf{T} + WHYY^\mathsf{T}H^\mathsf{T},\\ \nabla_H &= -\alpha_n W^\mathsf{T} Y^\mathsf{T} + W^\mathsf{T}WHYY^\mathsf{T}.\end{aligned}$$

Here multiplying W^T by $WHYY^\mathsf{T}$ to compute ∇_H at the first iteration has complexity $O(n^2 r)$, assuming $W \in \mathbb{R}^{n\times r}$ and $H \in \mathbb{R}^{r\times n}$.[2] In short, the first iteration of this NMF scheme has higher complexity than our algorithm. We also note [127] provides similar algorithms but tailored to stochastic matrices (such as Π_n); however, these still involve computation of ∇_W, ∇_H and a multiplicative update, so the same issue remains.

B.8.2 Representation learning

In Section 3.5.2, we noted a connection between our algorithm and the representation learning scheme from [73]. At a high level, the latter proceeds in two stages. The first stage learns certain parameters for the second stage via stochastic gradient descent; the second stage (Algorithm 1 in [73]) is described informally as Algorithm B.4 here. Roughly, this second stage begins by representing each node with a given feature vector, then updates this representation based on features of neighbors (after one iteration), neighbors of neighbors (after two iterations), etc. There are some issues with applying Algorithm B.4 to our setting. First, it relies on given feature vectors, derived from e.g. text data pertaining to each node; our algorithm assumes only the graph structure is known. Second, it applies to undirected graphs; we have assumed a directed graph throughout the chapter. However, setting these

[2] This claimed complexity assumes W, H are initialized as dense matrices. We assumed this because, if instead they are initialized as sparse matrices, the resulting estimate WH could be far from Π_n. This latter claim follows from the update rule (B.97): entries of W, H initialized to zero will remain zero; hence, if $W(i,:)$ or $H(:,j)$ contain mostly zeros but $\Pi_n(i,j)$ is large, the estimate $W(i,:)H(:,j)$ could be far from $\Pi_n(i,j)$.

issues aside, we next discuss two more subtle issues with adapting this algorithm to PPR.

We first consider the most immediate adaptation of Algorithm B.4 to our setting, which we present as Algorithm B.5. Here we let feature vectors simply be point masses on each node (Line 1), we let the aggregate function be a simple average (Line 4), and we update h_v^k as a weighted average of this aggregated vector and e_v^T (Line 5). In essence, we have chosen the learned parameters for Algorithm B.4, rather than learning them. This choices guarantee

$$\begin{bmatrix} h_1^i \\ \vdots \\ h_n^i \end{bmatrix} = \alpha_n \sum_{j=0}^{i} (1-\alpha_n)^j P^j \xrightarrow[i\to\infty]{} \Pi_n, \tag{B.98}$$

where h_v^i is the representation of v after i iterations of Algorithm B.5, the equality holds by Proposition B.1 below, and the limit holds by (1.2). Hence, with these chosen parameters, running Algorithm B.4 is effectively the same as computing the power iteration in (B.98). However, as discussed in Section 3.5.1, `Approx-PageRank` and `Approx-Contributions` are refined versions of this power iteration (with stronger complexity and accuracy guarantees), and these methods have complexity $O(n^2 \log n)$. Hence, we strongly suspect that this immediate adaptation of Algorithm B.4 will have worse performance than our algorithm (which has complexity $O(n^{\bar{c}}), \bar{c} < 2$). Of course, the preceding paragraph only considers one choice of parameters for Algorithm B.4. We could also consider learning these parameters. However, [73] states that running Algorithm B.4 with learned parameters causes prohibitively long runtime when $I > 2$ (where I is the number of iterations for the algorithm). Hence, for feasible choices of I, each node's ultimate representation only depends on its two-step neighborhood. We believe this would lead to very poor accuracy in our setting. This is because, as described in Section 3.4.2, the set of nodes with large PPR grows with n when $\alpha_n \propto 1/\log n$. Hence, approximating a node's PPR vector while only accounting for its two-step neighborhood will give exceedingly poor accuracy as n grows.

Algorithm B.4: Stage 2 from [73]

1 $h_v^0 \leftarrow x_v \; \forall \; v \in V_n$, where x_v is a given feature vector
2 **for** $i = 1$ **to** I **do**
3 **for** $v \in V_n$ **do**
4 $h_{\mathcal{N}(v)}^k \leftarrow \texttt{AGGREGATE}_k(\{h_u^{k-1} : u \in \mathcal{N}(v)\})$, where $\mathcal{N}(v)$ are v's neighbors and $\texttt{AGGREGATE}_k$ is a para- meterized function with parameters learned in Stage 1
5 $h_v^k \leftarrow \sigma(W_k \texttt{CONCAT}(h_v^{k-1}, h_{\mathcal{N}(v)}^k))$, where σ is a nonlinearity, W_k is a matrix learned in Stage 1, and $\texttt{CONCAT}(x, y)$ concatenates vectors x and y

Proposition B.1. For any $v \in V_n$ and any iteration i in Algorithm B.5,

$$h_v^i = \alpha_n e_v^\mathsf{T} \sum_{j=0}^{i} (1-\alpha_n)^j P^j.$$

Proof. We use induction. For $i = 0$, the proposition is immediate by Line 1 of Algorithm

Algorithm B.5: Adapting Algorithm B.4 to PPR

1 $h_v^0 \leftarrow e_v^\mathsf{T} \; \forall \; v \in V_n$
2 **for** $i = 1$ **to** I **do**
3 **for** $v \in V_n$ **do**
4 $h_{\mathcal{N}(v)}^k = \frac{1}{D_v} \sum_{u \in N_{out}(v)} h_u^{k-1}$
5 $h_v^k \leftarrow \alpha_n e_v^\mathsf{T} + (1 - \alpha_n) h_{\mathcal{N}(v)}^k$

B.5. Assuming true for $i - 1$, we have

$$
\begin{aligned}
h_v^i &= \alpha_n e_v^\mathsf{T} + \frac{(1-\alpha_n)}{D_v} \sum_{u \in N_{out}(v)} h_u^{k-1} \\
&= \alpha_n e_v^\mathsf{T} + \frac{(1-\alpha_n)}{D_v} \sum_{u \in N_{out}(v)} \left(\alpha_n e_u^\mathsf{T} \sum_{j=0}^{i-1} (1-\alpha_n)^j P^j \right) \\
&= \alpha_n e_v^\mathsf{T} + \alpha_n (1-\alpha_n) \left(\frac{1}{D_v} \sum_{u \in N_{out}(v)} e_u^\mathsf{T} \right) \sum_{j=0}^{i-1} (1-\alpha_n)^j P^j \\
&= \alpha_n e_v^\mathsf{T} + \alpha_n (1-\alpha_n) \left(e_v^\mathsf{T} P \right) \sum_{j=0}^{i-1} (1-\alpha_n)^j P^j \\
&= \alpha_n e_v^\mathsf{T} + \alpha_n e_v^\mathsf{T} \sum_{j=1}^{i} (1-\alpha_n)^j P^j = \alpha_n e_v^\mathsf{T} \sum_{j=0}^{i} (1-\alpha_n)^j P^j,
\end{aligned}
$$

where the first equality holds by Lines 4 and 5 of Algorithm B.5, the second uses the inductive hypothesis, and the fourth holds since P is the row-normalized adjacency matrix. □

APPENDIX C

Proofs for Chapter IV

C.1 Proof of Theorem 4.1

First note that, as stated in the proof sketch,

$$\mathbb{P}(\|\hat{v}_{k_*} - v\|_\infty \geq 2\varepsilon) \leq \mathbb{P}(\|\overline{v} - v\|_\infty \geq \varepsilon). \tag{C.1}$$

We next derive a pointwise bound for $\|\overline{v} - v\|_\infty$. First, fix $T \in \mathbb{N}$ and observe

$$\|\overline{v} - v\|_\infty \leq (1-\alpha)\sum_{t=1}^{\infty} \alpha^t \|(\overline{Q}^t - Q^t)c\|_\infty \leq (1-\alpha)\sum_{t=1}^{T-1} \alpha^t \|(\overline{Q}^t - Q^t)c\|_\infty + 2\|c\|_\infty \alpha^T, \tag{C.2}$$

where the first inequality is convexity and the second holds since by row stochasticity,

$$(1-\alpha)\sum_{t=T}^{\infty} \alpha^t \|(\overline{Q}^t - Q^t)c\|_\infty \leq (1-\alpha)\sum_{t=T}^{\infty} \alpha^t (\|\overline{Q}^t c\|_\infty + \|Q^t c\|_\infty) \leq 2\|c\|_\infty \alpha^T. \tag{C.3}$$

Now for large enough T, the bound in (C.3) falls below $\varepsilon/2$; in particular,

$$T \geq \log(4\|c\|_\infty/\varepsilon)/(1-\alpha) \quad \Rightarrow \quad 2\|c\|_\infty \alpha^T \leq 2\|c\|_\infty e^{-(1-\alpha)T} \leq \varepsilon/2 \tag{C.4}$$

Furthermore, for the t-th summand in (C.2), we can use the triangle inequality to write

$$\|(\overline{Q}^t - Q^t)c\|_\infty \leq \|\overline{Q}(\overline{Q}^{t-1} - Q^{t-1})c\|_\infty + \|(\overline{Q} - Q)Q^{t-1}c\|_\infty. \tag{C.5}$$

For the first summand in (C.5), we have by convexity and row stochasticity,

$$\|\overline{Q}(\overline{Q}^{t-1} - Q^{t-1})c\|_\infty \leq \max_{s\in\mathcal{S}} \sum_{s'=1}^{S} \overline{Q}(s,s') |(\overline{Q}^{t-1}(s',\cdot) - Q^{t-1}(s',\cdot))c| \leq \|(\overline{Q}^{t-1} - Q^{t-1})c\|_\infty.$$

We can then combine the previous two inequalities and iterate to obtain

$$\|(\overline{Q}^t - Q^t)c\|_\infty \le \sum_{t=1}^{t} \|(\overline{Q} - Q)Q^{\tau-1}c\|_\infty \le t \max_{\tau\in[T]} \|(\overline{Q} - Q)Q^{\tau-1}c\|_\infty.$$

Since this holds uniformly in t, we obtain

$$(1-\alpha)\sum_{t=1}^{T-1} \alpha^t \|(\overline{Q}^t - Q^t)c\|_\infty \le \max_{\tau\in[T]} \|(\overline{Q} - Q)Q^{\tau-1}c\|_\infty \frac{\alpha}{1-\alpha}. \tag{C.6}$$

To summarize, for T as in (C.4) we have shown

$$\|\overline{v} - v\|_\infty \le \max_{\tau\in[T]} \|(\overline{Q} - Q)Q^{\tau-1}c\|_\infty \frac{\alpha}{1-\alpha} + \frac{\varepsilon}{2},$$

and so, by the union bound,

$$\mathbb{P}(\|\overline{v} - v\|_\infty \ge \varepsilon) \le \sum_{t=1}^{T} \mathbb{P}\left(\|(\overline{Q} - Q)Q^{t-1}c\|_\infty \ge \frac{\varepsilon(1-\alpha)}{2\alpha}\right). \tag{C.7}$$

Now consider the t-th summand in (C.7). Since $\overline{Q}$ and $\tilde{Q}$ have the same distribution,

$$\mathbb{P}\left(\|(\overline{Q} - Q)Q^{t-1}c\|_\infty \ge \frac{\varepsilon(1-\alpha)}{2\alpha}\right) = \mathbb{P}\left(\|(\tilde{Q} - Q)Q^{t-1}c\|_\infty \ge \frac{\varepsilon(1-\alpha)}{2\alpha}\right). \tag{C.8}$$

To bound the right side of (C.8), we first define $d_{t-1} = Q^{t-1}c$ and observe that for any $s \in \mathcal{S}$,

$$\tilde{Q}(s,\cdot)Q^{t-1}c = \sum_{s'=1}^{S} \tilde{Q}(s,s')d_{t-1}(s') = \sum_{s'=1}^{S}\left(\frac{1}{n}\sum_{i=1}^{n} 1(Y_{s,i} = s')\right) d_{t-1}(s') = \frac{1}{n}\sum_{i=1}^{n} d_{t-1}(Y_{s,i}).$$

Moreover, for any $s \in \mathcal{S}, i \in [n]$ we have

$$Q(s,\cdot)Q^{t-1}c = \sum_{s'=1}^{s} Q(s,s')d_{t-1}(s') = \sum_{s'=1}^{s} \mathbb{P}(Y_{s,i} = s')d_{t-1}(s') = \mathbb{E}d_{t-1}(Y_{s,i}).$$

Combining the previous two equations, and again the union bound, we obtain

$$\begin{aligned}
&\mathbb{P}\left(\|(\tilde{Q} - Q)Q^{t-1}c\|_\infty \ge \frac{\varepsilon(1-\alpha)}{2\alpha}\right) \qquad (C.9)\\
&\quad= \mathbb{P}\left(\max_{s\in\mathcal{S}} \left|\frac{1}{n}\sum_{i=1}^{n}(d_{t-1}(Y_{s,i}) - \mathbb{E}d_{t-1}(Y_{s,i}))\right| \ge \frac{\varepsilon(1-\alpha)}{2\alpha}\right)\\
&\quad\le \sum_{s=1}^{S} \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}(d_{t-1}(Y_{s,i}) - \mathbb{E}d_{t-1}(Y_{s,i}))\right| \ge \frac{\varepsilon(1-\alpha)}{2\alpha}\right).
\end{aligned}$$

Now fix $s \in \mathcal{S}$. Since $\{Y_{s,i}\}_{i=1}^n$ are independent, $\{d_{t-1}(Y_{s,i})\}_{i=1}^n$ are independent as well. Moreover, $d_{t-1}(Y_{s,i})$ takes values in $[0, \|c\|_\infty]$. Thus, by the Chernoff bound (C.24),

$$\begin{aligned}\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n (d_{t-1}(Y_{s,i}) - \mathbb{E}d_{t-1}(Y_{s,i}))\right| \geq \frac{\varepsilon(1-\alpha)}{2\alpha}\right) \\ = \mathbb{P}\left(\left|\sum_{i=1}^n \left(\frac{d_{t-1}(Y_{s,i})}{\|c\|_\infty} - \frac{\mathbb{E}d_{t-1}(Y_{s,i})}{\|c\|_\infty}\right)\right| \geq \frac{n\varepsilon(1-\alpha)}{2\|c\|_\infty \alpha}\right) \\ \leq 2\exp\left(-\frac{n\varepsilon^2(1-\alpha)^2}{2\|c\|_\infty^2\alpha^2}\right) \leq \frac{\delta}{ST},\end{aligned} \tag{C.10}$$

where the final inequality holds assuming we choose T as small as possible in (C.4) and by the assumption on n. Combining (C.1), (C.7), (C.8), (C.9), and (C.10) implies the theorem.

Remark C.1. This proof assumes the cost vector c is deterministic; in the setting of Theorem 4.2, the cost vector C is random. In the latter case, we can replace $\mathbb{P}(\cdot)$ by $\mathbb{P}(\cdot|C)$ but otherwise follow the same proof to obtain $\mathbb{P}(\|\hat{v}_{k_*} - v\|_\infty \geq 2\delta|C) \leq \delta$ *a.s.* and then average over C to obtain the same result, assuming the lower bound on n holds almost surely.

C.2 Proof of Theorem 4.3

Define $\underline{Q}, \underline{v}$ as in (4.12). We also define the events

$$\begin{aligned}E_1 &= \cup_{s=1}^S \left\{|\hat{v}_{BD}(s) - v(s)| \geq \varepsilon_{rel} v(s) + \varepsilon_{abs}\right\}, \\ E_{2,s} &= \left\{|\underline{v}(s) - v(s)| \geq \frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2}\right\}, \quad E_2 = \cup_{s=1}^S E_{2,s}, \\ E_{3,s} &= \left\{|\hat{v}_{BD}(s) - \underline{v}(s)| \geq \frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2}\right\}, \quad E_3 = \cup_{s=1}^S E_{3,s}.\end{aligned}$$

As discussed in the proof sketch, we let $\mathcal{G} = \sigma(\{\hat{v}_k, r_k, U_k, \hat{Q}_k, s_{k+1}\}_{k=0}^{k_*})$ denote σ-algebra generated by the random variables in the Algorithm 4.2 subroutine of Algorithm 4.3. Note in particular that $\underline{Q}$ is $\mathcal{G}$-measurable, and thus $\underline{v}$ is $\mathcal{G}$-measurable; consequently, $E_{2,s} \in \mathcal{G}$. Using these definitions, we state two key lemmas.

Lemma C.1. For n_B as in the theorem statement, $\mathbb{P}(E_2) \leq \delta/2$.

Lemma C.2. For n_F as in the theorem statement, and $\forall\ s \in \mathcal{S}$, $\mathbb{P}(E_{3,s}|\mathcal{G})1(E_{2,s}^C) \leq \delta/(2S)$ *a.s.*

Before proving the lemmas, we show that they imply the theorem. Towards this end, first note $E_1 \subset E_2 \cup E_3$ by the triangle inequality, so $E_1 \cap E_2^C \subset E_3 \cap E_2^C$. Consequently,

$$\mathbb{P}(E_1) = \mathbb{P}(E_1 \cap E_2) + \mathbb{P}(E_1 \cap E_2^C) \leq \mathbb{P}(E_2) + \mathbb{P}(E_3 \cap E_2^C).$$

Furthermore, by the union bound and monotonicity, we have

$$\mathbb{P}(E_3 \cap E_2^C) \leq \sum_{s=1}^S \mathbb{P}(E_{3,s} \cap E_2^C) \leq \sum_{s=1}^S \mathbb{P}(E_{3,s} \cap E_{2,s}^C).$$

Now fix $s \in \mathcal{S}$. Then since $E_{2,s}^C \in \mathcal{G}$, we can write

$$\mathbb{P}(E_{3,s} \cap E_{2,s}^C) = \mathbb{E}[\mathbb{P}(E_{3,s}|\mathcal{G})1(E_{2,s}^C)].$$

Combining the previous three inequalities with the two lemmas, we obtain

$$\mathbb{P}(E_1) \leq \mathbb{P}(E_2) + \sum_{s=1}^{S} \mathbb{E}[\mathbb{P}(E_{3,s}|\mathcal{G})1(E_{2,s}^C)] \leq \delta,$$

and by definition of E_1, the theorem follows. We next return to prove the lemmas.

C.2.1 Proof of Lemma C.1

First, we define the constants

$$\bar{T} = \left\lceil \frac{\log(2\|c\|_\infty/\varepsilon_{abs})}{1-\alpha} \right\rceil, \quad \lambda = \frac{\log(1+\varepsilon_{rel}/2)}{\bar{T}}.$$

Next, we prove the following implication that was mentioned in the proof sketch:

$$|\underline{Q}(s,s') - Q(s,s')| \leq \lambda Q(s,s') \;\forall\; s,s' \in \mathcal{S} \Rightarrow |\underline{v}(s) - v(s)| \leq \frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2} \;\forall\; s \in \mathcal{S}. \quad \text{(C.11)}$$

Assume the left side of (C.11) holds and fix $s \in \mathcal{S}$. Then clearly

$$\begin{aligned}
&(1-\alpha)\sum_{t=\bar{T}}^{\infty} \alpha^t \underline{Q}^t(s,\cdot)c \leq (1-\alpha)\sum_{t=\bar{T}}^{\infty} \alpha^t \|c\|_\infty = \alpha^{\bar{T}}\|c\|_\infty \leq e^{-(1-\alpha)\bar{T}}\|c\|_\infty \leq \frac{\varepsilon_{abs}}{2} \\
&\Rightarrow \underline{v}(s) = (1-\alpha)\sum_{t=0}^{\infty} \alpha^t \underline{Q}^t(s,\cdot)c \leq (1-\alpha)\sum_{t=0}^{\bar{T}-1} \alpha^t \underline{Q}^t(s,\cdot)c + \frac{\varepsilon_{abs}}{2}. \qquad \text{(C.12)}
\end{aligned}$$

We next upper bound the term $\underline{Q}^t(s,\cdot)c$ in the t-th summand of (C.12). For $t=0$, this term is simply $c(s) < (1+\lambda)^{\bar{T}}c(s)$. For $t=1$, the left side of (C.11) implies

$$\underline{Q}(s,\cdot)c = \sum_{s'=1}^{S} \underline{Q}(s,s')c(s') \leq (1+\lambda)\sum_{s'=1}^{S} Q(s,s')c(s') = (1+\lambda)Q(s,\cdot)c < (1+\lambda)^{\bar{T}}Q(s,\cdot)c.$$

Finally, for $t \in \{2,\ldots,\bar{T}-1\}$, the left side of (C.11) similarly gives

$$\begin{aligned}
\underline{Q}^t(s,\cdot)c &= \sum_{s'\in\mathcal{S}} \sum_{s_1,\ldots,s_{t-1}\in\mathcal{S}} \underline{Q}(s,s_1)\underline{Q}(s_1,s_2)\cdots\underline{Q}(s_{t-2},s_{t-1})\underline{Q}(s_{t-1},s')c(s') \\
&\leq (1+\lambda)^t \sum_{s'\in\mathcal{S}} \sum_{s_1,\ldots,s_{t-1}\in\mathcal{S}} Q(s,s_1)Q(s_1,s_2)\cdots Q(s_{t-2},s_{t-1})Q(s_{t-1},s')c(s') \\
&= (1+\lambda)^t Q^t(s,\cdot)c < (1+\lambda)^{\bar{T}}Q^t(s,\cdot)c.
\end{aligned}$$

In summary, we have shown $\underline{Q}^t(s,\cdot)c < (1+\lambda)^{\bar{T}} Q^t(s,\cdot)c \;\forall\; t \in \{0,\ldots,\bar{T}-1\}$. Also,

$$(1+\lambda)^{\bar{T}} < e^{\lambda \bar{T}} \leq 1 + \frac{\varepsilon_{rel}}{2}. \tag{C.13}$$

Combining these observations, we can further bound (C.12) as

$$\underline{v}(s) \leq \left(1 + \frac{\varepsilon_{rel}}{2}\right)(1-\alpha)\sum_{t=0}^{\bar{T}-1} \alpha^t Q^t(s,\cdot)c + \frac{\varepsilon_{abs}}{2} \leq \left(1 + \frac{\varepsilon_{rel}}{2}\right) v(s) + \frac{\varepsilon_{abs}}{2}. \tag{C.14}$$

For a lower bound on $\underline{v}(s)$, we similarly have

$$\begin{aligned}
\underline{v}(s) &\geq (1-\alpha)\sum_{t=0}^{\bar{T}-1} \alpha^t \underline{Q}^t(s,\cdot)c \geq (1-\lambda)^{\bar{T}}(1-\alpha)\sum_{t=0}^{\bar{T}-1} \alpha^t Q^t(s,\cdot)c \\
&= (1-\lambda)^{\bar{T}}\left(v(s) - (1-\alpha)\sum_{t=\bar{T}}^{\infty} \alpha^t Q^t(s,\cdot)c\right) \geq (1-\lambda)^{\bar{T}}\left(v(s) - \frac{\varepsilon_{abs}}{2}\right).
\end{aligned}$$

We next loosen this bound further. First, by convexity and (C.13),

$$2 = 2\left(\frac{1+\lambda}{2} + \frac{1-\lambda}{2}\right)^{\bar{T}} \leq (1+\lambda)^{\bar{T}} + (1-\lambda)^{\bar{T}} \leq \left(1 + \frac{\varepsilon_{rel}}{2}\right) + (1-\lambda)^{\bar{T}},$$

and so $(1-\lambda)^{\bar{T}} \geq 1 - \varepsilon_{rel}/2$. Since also $(1-\lambda)^{\bar{T}} \leq 1$, we thus obtain

$$\underline{v}(s) \geq \left(1 - \frac{\varepsilon_{rel}}{2}\right)\left(v(s) - \frac{\varepsilon_{abs}}{2}\right) \geq \left(1 - \frac{\varepsilon_{rel}}{2}\right) v(s) - \frac{\varepsilon_{abs}}{2}. \tag{C.15}$$

In summary, we have shown that if the left side of (C.11) holds, then (C.14) and (C.15) hold as well. Since (C.14) and (C.15) together imply the right side of (C.11), (C.11) is proven. We can now use (C.11) to prove the lemma. First, (C.11) and the union bound imply

$$\begin{aligned}
\mathbb{P}(E_2) &\leq \mathbb{P}\left(\cup_{s,s' \in \mathcal{S}} \{|\underline{Q}(s,s') - Q(s,s')| > \lambda Q(s,s')\}\right) \\
&\leq \sum_{s,s' \in \mathcal{S}} \mathbb{P}(|\underline{Q}(s,s') - Q(s,s')| > \lambda Q(s,s')).
\end{aligned} \tag{C.16}$$

Now for the (s,s')-th summand in (C.16), we first note

$$\begin{aligned}
\mathbb{P}(|\underline{Q}(s,s') - Q(s,s')| > \lambda Q(s,s')) &\leq \mathbb{P}(|\overline{Q}(s,s') - Q(s,s')| > \lambda Q(s,s')) \\
&= \mathbb{P}(|\tilde{Q}(s,s') - Q(s,s')| > \lambda Q(s,s'))
\end{aligned} \tag{C.17}$$

where the inequality holds since $|\underline{Q}(s,s') - Q(s,s')| \leq |\overline{Q}(s,s') - Q(s,s')|$ pointwise by (4.11)-(4.12) and the equality since $\overline{Q}$ and $\tilde{Q}$ have the same distribution. Substituting into (C.16),

$$\mathbb{P}(E_2) \leq \sum_{s,s' \in \mathcal{S}} \mathbb{P}(|\tilde{Q}(s,s') - Q(s,s')| > \lambda Q(s,s')), \tag{C.18}$$

so our goal is to bound each summand in (C.18) by $\delta/(2S^2)$. If $Q(s,s')=0$, this is trivial; if instead $Q(s,s')>0$, the Chernoff bound (C.25) implies

$$\mathbb{P}(|\tilde{Q}(s,s')-Q(s,s')|>\lambda Q(s,s'))\le 2\exp\left(-\frac{n_B\lambda^2\min_{i,j\in\mathcal{S}:Q(i,j)>0}Q(i,j)}{3}\right)\le\frac{\delta}{2S^2},$$

where the final inequality holds by assumption on n_B.

C.2.2 Proof of Lemma C.2

Fix $s\in\mathcal{S}$. Then by definition of $E_{2,s},E_{3,s}$, we aim to show

$$|\underline{v}(s)-v(s)|<\frac{\varepsilon_{rel}}{2}v(s)+\frac{\varepsilon_{abs}}{2}\Rightarrow\mathbb{P}\left(|\hat{v}_{BD}(s)-\underline{v}(s)|\ge\frac{\varepsilon_{rel}}{2}v(s)+\frac{\varepsilon_{abs}}{2}\Big|\mathcal{G}\right)\le\frac{\delta}{2S}\ a.s. \quad \text{(C.19)}$$

Assume the left side of (C.19) holds. Recall by Algorithm 4.3 and the $\underline{Q}$-invariant (4.13),

$$\hat{v}_{BD}(s)=\hat{v}_{k_*}(s)+\frac{1}{n_F}\sum_{i=1}^{n_F}r_{k_*}(Z_{s,i}),\underline{v}(s)=\hat{v}_{k_*}(s)+\underline{\mu_s}r_{k_*}=v_{k_*}(s)+\frac{1}{n_F}\sum_{i=1}^{n_F}\mathbb{E}[r_{k_*}(Z_{s,i})|\mathcal{G}]. \quad \text{(C.20)}$$

Consequently, defining $\bar{Z}_s=\sum_{i=1}^{n_F}r_{k_*}(Z_{s,i})/\varepsilon$, we have

$$\begin{aligned}&\mathbb{P}\left(|\hat{v}_{BD}(s)-\underline{v}(s)|>\frac{\varepsilon_{rel}}{2}v(s)+\frac{\varepsilon_{abs}}{2}\Big|\mathcal{G}\right) \quad \text{(C.21)}\\ &\quad=\mathbb{P}\left(|\bar{Z}_s-\mathbb{E}[\bar{Z}_s|\mathcal{G}]|>\frac{n_F}{\varepsilon}\left(\frac{\varepsilon_{rel}}{2}v(s)+\frac{\varepsilon_{abs}}{2}\right)\Big|\mathcal{G}\right).\end{aligned}$$

Note that conditioned on $\mathcal{G}$, $\bar{Z}_s$ is a sum of independent $[0,1]$-valued random variables, so the Chernoff bounds from Appendix C.5 apply. We consider two cases:

- $\mathbb{E}[\bar{Z}_s|\mathcal{G}]<n_F\varepsilon_{abs}/(12\varepsilon)$: Here we bound the right side of (C.21) as

$$\begin{aligned}&\mathbb{P}\left(|\bar{Z}_s-\mathbb{E}[\bar{Z}_s|\mathcal{G}]|>\frac{n_F}{\varepsilon}\left(\frac{\varepsilon_{rel}}{2}v(s)+\frac{\varepsilon_{abs}}{2}\right)\Big|\mathcal{G}\right)\le\mathbb{P}\left(|\bar{Z}_s-\mathbb{E}[\bar{Z}_s|\mathcal{G}]|>\frac{n_F\varepsilon_{abs}}{2\varepsilon}\Big|\mathcal{G}\right)\\ &=\mathbb{P}\left(\bar{Z}_s-\mathbb{E}[\bar{Z}_s|\mathcal{G}]>\frac{n_F\varepsilon_{abs}}{2\varepsilon}\Big|\mathcal{G}\right)+\mathbb{P}\left(\mathbb{E}[\bar{Z}_s|\mathcal{G}]-\bar{Z}_s>\frac{n_F\varepsilon_{abs}}{2\varepsilon}\Big|\mathcal{G}\right)\\ &\le\mathbb{P}\left(\bar{Z}_s>\frac{n_F\varepsilon_{abs}}{2\varepsilon}\Big|\mathcal{G}\right)+0,\end{aligned}$$

  where the first inequality and the equality are immediate, and the second inequality holds since, by assumption on $\mathbb{E}[\bar{Z}_s|\mathcal{G}]$, $\mathbb{E}[\bar{Z}_s|\mathcal{G}]-\bar{Z}_s\le\mathbb{E}[\bar{Z}_s|\mathcal{G}]<n_F\varepsilon_{abs}/(12\varepsilon)<n_F\varepsilon_{abs}/(2\varepsilon)$, so $\mathbb{E}[\bar{Z}_s|\mathcal{G}]-\bar{Z}_s>n_F\varepsilon_{abs}/(2\varepsilon)$ cannot occur. For the remaining term, since $\mathbb{E}[\bar{Z}_s|\mathcal{G}]<(1/6)\times n_F\varepsilon_{abs}/(2\varepsilon)$ we can use the Chernoff bound (C.26) to obtain

$$\begin{aligned}\mathbb{P}\left(|\bar{Z}_s-\mathbb{E}[\bar{Z}_s|\mathcal{G}]|>\frac{n_F}{\varepsilon}\left(\frac{\varepsilon_{rel}}{2}v(s)+\frac{\varepsilon_{abs}}{2}\right)\Big|\mathcal{G}\right)&\le\mathbb{P}\left(\bar{Z}_s>\frac{n_F\varepsilon_{abs}}{2\varepsilon}\Big|\mathcal{G}\right)\\ &\le 2^{-n_F\varepsilon_{abs}/(2\varepsilon)}\le\frac{\delta}{4S},\end{aligned}$$

where the final inequality holds since, by the theorem statement,

$$n_F \geq \frac{324\varepsilon \log(4S/\delta)}{\varepsilon_{rel}^2 \varepsilon_{abs}} = \frac{162}{\varepsilon_{rel}^2 \log_2 e} \frac{2\varepsilon \log_2(4S/\delta)}{\varepsilon_{abs}} \geq \frac{2\varepsilon \log_2(4S/\delta)}{\varepsilon_{abs}}.$$

- $\mathbb{E}[\bar{Z}_s|\mathcal{G}] \geq n_F \varepsilon_{abs}/(12\varepsilon)$: We first observe

$$\underline{v}(s) < \left(1 + \frac{\varepsilon_{rel}}{2}\right) v(s) + \frac{\varepsilon_{abs}}{2} \quad \Leftrightarrow \quad \frac{\underline{v}(s) - \varepsilon_{abs}/2}{1 + \varepsilon_{rel}/2} < v(s).$$

Consequently, the left side of (C.19) implies

$$\frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2} > \frac{\varepsilon_{rel}}{2} \frac{\underline{v}(s) - \varepsilon_{abs}/2}{1 + \varepsilon_{rel}/2} + \frac{\varepsilon_{abs}}{2} = \frac{\varepsilon_{rel} \underline{v}(s)}{2 + \varepsilon_{rel}} + \frac{\varepsilon_{abs}}{2} \left(1 - \frac{\varepsilon_{rel}/2}{1 + \varepsilon_{rel}/2}\right) > \frac{\varepsilon_{rel} \underline{v}(s)}{3},$$

where we used $\varepsilon_{rel} \in (0, 1)$. Since also $\underline{v}(s) \geq \mathbb{E}[r_{k_*}(Z_{s,i})|\mathcal{G}]$ by (C.20), we thus obtain

$$\frac{n_F}{\varepsilon} \left(\frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2}\right) > \frac{n_F}{\varepsilon} \frac{\varepsilon_{rel} \mathbb{E}[r_{k_*}(Y_{s,i})|\mathcal{G}]}{3} = \frac{\varepsilon_{rel}}{3} \frac{n_F \mathbb{E}[r_{k_*}(Y_{s,i})|\mathcal{G}]}{\varepsilon} = \frac{\varepsilon_{rel}}{3} \mathbb{E}[\bar{Z}_s|\mathcal{G}].$$

Therefore, we can bound the right side of (C.21) as

$$\begin{aligned} \mathbb{P}\left(|\bar{Z}_s - \mathbb{E}[\bar{Z}_s|\mathcal{G}]| > \frac{n_F}{\varepsilon} \left(\frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2}\right) \middle| \mathcal{G}\right) &\leq \mathbb{P}\left(|\bar{Z}_s - \mathbb{E}[\bar{Z}_s|\mathcal{G}]| > \frac{\varepsilon_{rel}}{3} \mathbb{E}[\bar{Z}_s|\mathcal{G}] \middle| \mathcal{G}\right) \\ &\leq 2 \exp\left(-\frac{(\varepsilon_{rel}/3)^2}{3} \mathbb{E}[\bar{Z}_s|\mathcal{G}]\right) \leq 2 \exp\left(-\frac{\varepsilon_{rel}^2}{27} \frac{n_F \varepsilon_{abs}}{12\varepsilon}\right) \leq \frac{\delta}{2S}, \end{aligned}$$

where we used Chernoff bound (C.25), the $\mathbb{E}[\bar{Z}_s|\mathcal{G}]$ assumption, and the n_F assumption.

Remark C.2. The proof of Lemma C.1 extends to random cost vectors C by replacing $\mathbb{P}(\cdot)$ by $\mathbb{P}(\cdot|C)$ and then averaging over C, similar to the proof of Theorem 4.1 (see Remark C.1). Furthermore, recall $r_0 = C$ and thus C is $\mathcal{G}$-measurable by definition of $\mathcal{G}$, so the proof of Lemma C.1 is identical in the case of random cost C. Thus, when C is random, Lemmas C.1 and C.2 hold and can be used to prove the theorem as above.

C.3 Resampling approach

The resampling approach is formally defined in Algorithm C.1. In contrast to `Backward-EPE`, we estimate $Q(s, s_k)$ as follows at each iteration k: for $s \in N_{in}(s_k)$ we draw independent samples $\{X_{s,i}^k\}_{i=1}^n$ from $Q(s, \cdot)$ (Line 5 of Algorithm C.1), and for $s \notin N_{in}(s_k)$ we set $\hat{Q}_k(s, s_k) = 0$ (Line 6). We then compute $\hat{v}_k, r_k$ using the update rule from `Backward-EPE` (Lines 7-8). Finally, as in `Backward-EPE`, we terminate when $\|r_k\|_\infty \leq \varepsilon$.

We derive the martingale property stated in Section 4.4.3. Define the error process $e_k(s)$ as in Section 4.4.3, and define a filtration $\{\mathcal{F}_k\}_{k=0}^{k_*}$ by $\mathcal{F}_k = \sigma(\{\hat{v}_{k'}, r_{k'}, s_{k'+1}\}_{k'=0}^k)$, where by $\sigma(\cdot)$ we mean the generated σ-algebra. Now fix $k \in [k_*], s \in \mathcal{S}$. Then by the update in Lines 7-8 in Algorithm C.1, we have

$$e_k(s) = (\hat{v}_{k-1}(s) + (1 - \alpha) r_{k-1}(s) 1(s = s_k))$$

Algorithm C.1: `Backward-EPE-Resample`

1 $k = 0$, $\hat{v}_k = 0_{S\times 1}$, $r_k = c$
2 **while** $\|r_k\|_\infty > \varepsilon$ **do**
3 $\quad k \leftarrow k+1$, $s_k \sim \arg\max_{s\in\mathcal{S}} r_{k-1}(s)$ uniformly
4 $\quad$ **for** $s \in \mathcal{S}$ **do**
5 $\quad\quad$ **if** $s \in N_{in}(s_k)$ **then** $\{X^k_{s,i}\}_{i=1}^n \sim Q(s,\cdot)$, $\hat{Q}_k(s,s_k) = \frac{1}{n}\sum_{i=1}^n 1(X^k_{s,i} = s_k)$;
6 $\quad\quad$ **else** $\hat{Q}_k(s,s_k) = 0$;
7 $\quad\quad$ **if** $s = s_k$ **then** $\hat{v}_k(s) = \hat{v}_{k-1}(s) + (1-\alpha)r_{k-1}(s)$, $r_k(s) = \alpha\hat{Q}_k(s,s_k)r_{k-1}(s_k)$;
8 $\quad\quad$ **else** $\hat{v}_k(s) = \hat{v}_{k-1}(s)$, $r_k(s) = r_{k-1}(s) + \alpha\hat{Q}_k(s,s_k)r_{k-1}(s_k)$;

$$+\sum_{s'=1}^{s}\mu_s(s')(r_{k-1}(s')1(s'\neq s_k) + \alpha\hat{Q}_k(s',s_k)r_{k-1}(s_k)) - v(s)$$
$$= e_{k-1}(s) + r_{k-1}(s_k)\left(-\mu_s(s_k) + (1-\alpha)1(s=s_k) + \alpha\sum_{s'=1}^{s}\mu_s(s')\hat{Q}_k(s',s_k)\right) \quad \text{(C.22)}$$

Note that all terms in (C.22) except $\hat{Q}_k(s',s_k)$ are $\mathcal{F}_{k-1}$-measurable, and therefore

$$\begin{aligned}
&\mathbb{E}[e_k(s)|\mathcal{F}_{k-1}] \\
&= e_{k-1}(s) + r_{k-1}(s_k)\left(-\mu_s(s_k) + (1-\alpha)1(s=s_k) + \alpha\sum_{s'=1}^{s}\mu_s(s')\mathbb{E}[\hat{Q}_k(s',s_k)|\mathcal{F}_{k-1}]\right) \\
&= e_{k-1}(s) + r_{k-1}(s_k)\left(-\mu_s(s_k) + (1-\alpha)1(s=s_k) + \alpha\sum_{s'=1}^{s}\mu_s(s')Q(s',s_k)\right) = e_{k-1}(s),
\end{aligned}$$

where the first two equalities hold by Lines 5-6 of Algorithm C.1 and the third holds similar to (4.9). Hence, $\{e_k(s)\}_{k=0}^{k_*}$ is a martingale. Also note $e_0(s) = \hat{v}_0(s) + \mu_s r_0 - v(s) = 0 + \mu_s c - v(s) = 0$. We thus conclude $\mathbb{E}e_k(s) = 0$, i.e. the Q-invariant holds in expectation.

C.4 Analysis of existing approach

We recall from Section 4.1.1 that the approach from [19] proceeds as follows. Fix $T \in \mathbb{N}$ and, $\forall\ s \in \mathcal{S}$, sample m length-T trajectories $\{\{W_t^{s,i}\}_{t=0}^{T-1}\}_{i=1}^m$ from s, and estimate $v(s)$ as

$$\hat{v}_E(s) = \frac{1}{m}\sum_{i=1}^{m}(1-\alpha)\sum_{t=0}^{T-1}\alpha^t c(W_t^{s,i}).$$

(We use the subscript E to distinguish the estimate of this existing approach from the estimates of our algorithms.) To analyze this scheme, we follow the analysis of [19, Proposition 5.4]. By the argument leading to (C.4) in Appendix C.1 (but with a different constant),

$$T \geq \frac{\log(2\|c\|_\infty/\varepsilon)}{1-\alpha} \quad \Rightarrow \quad |\hat{v}_E(s) - v(s)| \leq |\hat{v}_E(s) - \mathbb{E}\hat{v}_E(s)| + \varepsilon, \quad \text{(C.23)}$$

so consequently, for T as in (C.23),

$$\mathbb{P}(|\hat{v}_E(s) - v(s)| \geq 2\varepsilon) \leq \mathbb{P}\left(|\hat{v}_E(s) - \mathbb{E}\hat{v}_E(s)| \geq \varepsilon\right).$$

Towards further bounding the right side, we write (as in (C.6))

$$|\hat{v}_E(s) - v(s)| \leq \max_{t \in [T-1]} \left| \frac{1}{m} \sum_{i=1}^{m} (c(W_t^{s,i}) - \mathbb{E}c(W_t^{s,i})) \right| \frac{\alpha}{1-\alpha}.$$

Combining the previous two inequalities, and using the union bound,

$$\mathbb{P}(|\hat{v}_E(s) - v(s)| \geq 2\varepsilon) \leq \sum_{t=1}^{T-1} \mathbb{P}\left(\left| \frac{1}{m} \sum_{i=1}^{m} (c(W_t^{s,i}) - \mathbb{E}c(W_t^{s,i})) \right| \geq \frac{\varepsilon(1-\alpha)}{\alpha} \right).$$

We then apply the Chernoff bound (C.24) to bound the t-th summand by

$$\mathbb{P}\left(\left| \sum_{i=1}^{m} \left(\frac{c(W_t^{s,i})}{\|c\|_\infty} - \frac{\mathbb{E}c(W_t^{s,i}))}{\|c\|_\infty} \right) \right| \geq \frac{m\varepsilon(1-\alpha)}{\|c\|_\infty \alpha} \right) \leq 2\exp\left(-\frac{2m\varepsilon^2(1-\alpha)^2}{\|c\|_\infty^2 \alpha^2} \right).$$

Note this holds uniformly in t; also, we can take a union bound over $s \in \mathcal{S}$ to obtain

$$\mathbb{P}(\|\hat{v}_E - v\|_\infty \geq 2\varepsilon) \leq 2ST\left(-\frac{2m\varepsilon^2(1-\alpha)^2}{\|c\|_\infty^2 \alpha^2} \right) \leq \delta,$$

where the final inequality holds assuming we choose

$$m \geq \frac{\|c\|_\infty^2 \alpha^2}{2\varepsilon^2(1-\alpha)^2} \log\left(\frac{2ST}{\delta} \right).$$

Note here that m is the number of length-T trajectories sampled from each state. Thus, the overall sample complexity is at least STm, which we can lower bound as

$$STm \geq \frac{S\|c\|_\infty^2 \alpha^2 \log(2\|c\|_\infty/\varepsilon)}{2\varepsilon^2(1-\alpha)^3} \log\left(\frac{2S}{\delta} \frac{\log(2\|c\|_\infty/\varepsilon)}{1-\alpha} \right).$$

C.5 Chernoff bounds

The following is a standard result used throughout our analysis.

Theorem C.1. Let $\{R_i\}_{i=1}^m$ be independent $[0,1]$-valued random variables, and define $R = \sum_{i=1}^m R_i$. Then

$$\mathbb{P}(|R - \mathbb{E}R| > \eta) \leq 2\exp(-2\eta^2/m) \ \forall \ \eta > 0, \tag{C.24}$$

$$\mathbb{P}(|R - \mathbb{E}R| > \eta\mathbb{E}R) \leq 2\exp(-\eta^2\mathbb{E}R/3) \ \forall \ \eta \in (0,1), \tag{C.25}$$

$$\mathbb{P}(R > \eta) \leq 2^{-\eta} \ \forall \ \eta > 6\mathbb{E}R. \tag{C.26}$$

Proof. See e.g. [124, Theorem 1.1]. □

APPENDIX D

Proofs for Chapter V

D.1 Existing results

Here we collect some existing results that will be used in our proofs. Most can be found in the textbook [99]. First, we recall some basic properties of total variation distance.

Lemma D.1. Let $\mu, \nu, \eta \in \Delta_{n-1}$. Then the following hold:

- (l_1 equivalence) $\|\mu - \nu\| = \frac{1}{2}\sum_{i=1}^{n} |\mu(i) - \nu(i)| = \frac{1}{2}\|\mu - \nu\|_1$.
- (Triangle inequality) $\|\mu - \nu\| \leq \|\mu - \eta\| + \|\eta - \nu\|$.
- (Convexity) $\|(\gamma\mu + (1-\gamma)\nu) - \eta\| \leq \gamma\|\mu - \eta\| + (1-\gamma)\|\nu - \eta\|$ $\forall\ \gamma \in (0,1)$.
- (Coupling) $\|\mu - \nu\| \leq \mathbb{P}(X \neq Y)$ for any coupling (X, Y) of μ and ν, i.e. for any pair of random variables X and Y with respective marginal distributions μ and ν.

Proof. For l_1 equivalence, see Proposition 4.2 in [99]. The triangle inequality and convexity follow from the corresponding l_1 properties. For coupling, see Proposition 4.7 in [99]. □

We next collect some basic mixing time results. These involve the *relaxation time*

$$t_{\text{rel}}^{(n)} = 1/(1 - \lambda_n^*), \tag{D.1}$$

where $1 - \lambda_n^*$ is the absolute spectral gap of P_n, defined by

$$\lambda_n^* = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } P_n, \lambda \neq 1\}.$$

(Note $P_n \in \mathcal{E}_n \Rightarrow \lambda_n^* < 1$ – see e.g. Lemma 12.1 in [99] – so (D.1) is well-defined in this case.)

Lemma D.2. Let $P_n \in \mathcal{E}_n\ \forall\ n \in \mathbb{N}$, and let $\varepsilon \in (0,1)$ be independent of n.

- For any $n, t \in \mathbb{N}$, $d_n(t) \leq \max_{i,j \in [n]} \|e_i P_n^t - e_j P_n^t\|$.
- If each P_n is lazy, then $\sup_{\delta \in (0,1)} \liminf_{n\to\infty} t_{\text{mix}}^{(n)}(\delta) = \infty$.
- For any $n, k \in \mathbb{N}$, $d_n(k t_{\text{mix}}^{(n)}(\varepsilon)) \leq (2\varepsilon)^k$.[1]
- If P_n is reversible, then $t_{\text{mix}}^{(n)}(\varepsilon) \geq (t_{\text{rel}}^{(n)} - 1)\log(1/(2\varepsilon))$.

[1] Note this motivates the convention $\varepsilon = 1/4$, which yields the convenient inequality $d_n(k t_{\text{mix}}^{(n)}) \leq 2^{-k}$.

- If $\{P_n\}_{n\in\mathbb{N}}$ exhibits pre-cutoff and each P_n is reversible, then $t_{\text{rel}}^{(n)} = o(t_{\text{mix}}^{(n)}(\varepsilon))$.

Proof. The first statement holds by global balance and convexity from Lemma D.1, i.e.

$$d_n(t) = \max_{i\in[n]} \left\| e_i P_n^t - \sum_{j\in[n]} \pi_n(j) e_j P_n^t \right\| \leq \max_{i\in[n]} \sum_{j\in[n]} \pi_n(j) \|e_i P_n^t - e_j P_n^t\| \leq \max_{i,j\in[n]} \|e_i P_n^t - e_j P_n^t\|.$$

For the second statement, let $i_n \in [n]$ be s.t. $\pi_n(i_n) \leq 1/n \ \forall\ n \in \mathbb{N}$ (clearly, such i_n exists). Then by definition of $d_n(t)$, definition of total variation, and laziness, we have $\forall\ n, t \in \mathbb{N}$,

$$d_n(t) \geq \|e_{i_n} P_n^t - \pi_n\| \geq (e_{i_n} P_n^t)(i_n) - \pi_n(i_n) \geq 2^{-t} - 1/n.$$

As a consequence of this inequality, we obtain

$$t_{\text{mix}}^{(n)}(\delta) \geq \log_2\left(\frac{1}{\delta + 1/n}\right) \ \forall\ n \in \mathbb{N}, \delta \in (0,1) \quad \Rightarrow \quad \sup_{\delta\in(0,1)} \liminf_{n\to\infty} t_{\text{mix}}^{(n)}(\delta) = \infty.$$

For the other statements, see Equation 4.34, Theorem 12.4, and Proposition 18.4 in [99]. □

Finally, we state the inequality from [22] discussed in the main text. Define the hitting time of $A \subset [n]$ as $T_n(A) = \inf\{t \in \mathbb{Z}_+ : X_n(t) \in A\}$. Given $\eta_1, \eta_3 \in (0,1)$, let

$$t_{\text{hit}}^{(n)}(1-\eta_3, \eta_1) = \min\left\{ t : \max_{x\in[n], A\subset[n]:\pi_n(A)\geq 1-\eta_3} \mathbb{P}_x(T_n(A) > t) \leq \eta_1 \right\}, \tag{D.2}$$

where $\mathbb{P}_x$ denotes probability conditioned on the chain starting from $X_n(0) = x$. We now state the aforementioned inequality, which relates (D.2) to mixing and relaxation times.

Lemma D.3. Let $P_n \in \mathcal{E}_n$ be lazy and reversible. Then for any $\eta_1, \eta_2, \eta_3 \in (0,1)$,

$$t_{\text{mix}}^{(n)}((\eta_1+\eta_2)\wedge 1) \leq t_{\text{hit}}^{(n)}(1-\eta_3, \eta_1) + \left\lceil \frac{t_{\text{rel}}^{(n)}}{2} \max\left\{ \log\left(\frac{2(1-\eta_1)^2}{\eta_1\eta_2\eta_3} \right), 0 \right\} \right\rceil.$$

Proof. See Corollary 3.1 in [22]. □

D.2 Proof of Lemma 5.1

For the upper bound, let $\{\tilde{P}_n\}_{n\in\mathbb{N}}$ satisfy $\tilde{P}_n \in B(P_n, \alpha_n)\ \forall\ n \in \mathbb{N}$, and let $\delta \in (0,1)$ be arbitrary. It suffices to show that for some $N \in \mathbb{N}$, $\|\pi_n - \tilde{\pi}_n\| < \delta\ \forall\ n \geq N$. First, $\forall\ n, t \in \mathbb{N}$,

$$\|\pi_n - \tilde{\pi}_n\| = \|\pi_n - \tilde{\pi}_n \tilde{P}_n^t\| \leq \|\pi_n - \tilde{\pi}_n P_n^t\| + \|\tilde{\pi}_n P_n^t - \tilde{\pi}_n \tilde{P}_n^t\| \leq d_n(t) + \max_{x\in[n]} \|e_x P_n^t - e_x \tilde{P}_n^t\|, \tag{D.3}$$

where we have used global balance and Lemma D.1. For the second term in (D.3), we claim

$$\max_{x\in[n]} \|e_x P_n^t - e_x \tilde{P}_n^t\| \leq \alpha_n t. \tag{D.4}$$

We prove (D.4) by induction. For $t = 1$, (D.4) holds by assumption. For general t, note

$$P_n^t - \tilde{P}_n^t = P_n(P_n^{t-1} - \tilde{P}_n^{t-1}) + (P_n - \tilde{P}_n)\tilde{P}_n^{t-1},$$

where we added and subtracted $P_n\tilde{P}_n^{t-1}$. Hence, by Lemma D.1, we have $\forall\ x \in [n]$,

$$\|e_x P_n^t - e_x \tilde{P}_n^t\| \leq \|e_x P_n(P_n^{t-1} - \tilde{P}_n^{t-1})\| + \|e_x(P_n - \tilde{P}_n)\tilde{P}_n^{t-1}\|. \tag{D.5}$$

For the first term in (D.5), we have by Lemma D.1 and the inductive hypothesis,

$$\|e_x P_n(P_n^{t-1} - \tilde{P}_n^{t-1})\| \leq \max_{y\in[n]} \|e_y P_n^{t-1} - e_y \tilde{P}_n^{t-1}\| \leq \alpha_n(t-1). \tag{D.6}$$

For the second term, we use the following: for a vector x and a row stochastic matrix A,

$$\|xA\|_1 \leq \sum_i \sum_j |x(j)|A(j,i) = \sum_j |x(j)| \sum_i A(j,i) = \sum_j |x(j)| = \|x\|_1.$$

Using this inequality and Lemma D.1, we can bound the second term in (D.5) as

$$\|e_x(P_n - \tilde{P}_n)\tilde{P}_n^{t-1}\| = \frac{1}{2}\|e_x(P_n - \tilde{P}_n)\tilde{P}_n^{t-1}\|_1 \leq \frac{1}{2}\|e_x(P_n - \tilde{P}_n)\|_1 = \|e_x(P_n - \tilde{P}_n)\| \leq \alpha_n, \tag{D.7}$$

where the final inequality holds by assumption. Combining (D.6) and (D.7) establishes (D.4). Substituting into (D.3), we have therefore shown

$$\|\pi_n - \tilde{\pi}_n\| \leq d_n(t) + \alpha_n t\ \forall\ n, t \in \mathbb{N}.$$

Now set $k = \lceil \log(2/\delta)/\log(1/(2\varepsilon)) \rceil$ and $t = kt_{\text{mix}}^{(n)}(\varepsilon)$. Note $k \in \mathbb{N}$ since $\delta \in (0,1)$ and $\varepsilon \in (0, 1/2)$. Hence, we can use Lemma D.2 to obtain

$$d_n(t) = d_n(kt_{\text{mix}}^{(n)}(\varepsilon)) \leq (2\varepsilon)^k \leq (2\varepsilon)^{\log(2/\delta)/\log(1/(2\varepsilon))} = \frac{\delta}{2}.$$

Furthermore, since k is independent of n and $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to 0$, we can find N s.t.

$$\alpha_n t = k\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) < \frac{\delta}{2}\ \forall\ n \geq N.$$

Hence, combining the previous three inequalities, we obtain $\|\pi_n - \tilde{\pi}_n\| < \delta\ \forall\ n \geq N$.

For the lower bound, we begin by stating and proving a weaker version of the result.

Lemma D.4. Let $P_n \in \mathcal{E}_n, \alpha_n \in (0,1)\ \forall\ n \in \mathbb{N}$, and let $\delta \in (0, 1/2)$ be independent of n. Assume $\{P_n\}_{n\in\mathbb{N}}$ exhibits pre-cutoff, each P_n is lazy and reversible, and $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\delta) = \infty$. Then $\exists\ \{\tilde{P}_n\}_{n\in\mathbb{N}}$ s.t. $\tilde{P}_n \in B(P_n, \alpha_n)\ \forall\ n \in \mathbb{N}$ and $\liminf_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| \geq 1 - 3\delta$.

Proof. First note that $1 - \delta, 1 - 2\delta \in (0,1)$ by assumption on δ, so $t_n := t_{\text{hit}}^{(n)}(1-\delta, 1-2\delta)$ is well-defined. Hence, by definition, $\exists\ x_n \in [n], A_n \subset [n]$ satisfying

$$\pi_n(A_n) \geq 1 - \delta, \quad \mathbb{P}_{x_n}(T_n(A_n) > t_n - 1) > 1 - 2\delta. \tag{D.8}$$

Now set $\tilde{P}_n = P_{\alpha_n, e_{x_n}}$. Then by the PPR power iteration property (1.1),

$$\tilde{\pi}_n(A_n) = \alpha_n \sum_{t=0}^{t_n-1} (1-\alpha_n)^t \mathbb{P}_{x_n}(X_n(t) \in A_n) + \alpha_n \sum_{t=t_n}^{\infty} (1-\alpha_n)^t \mathbb{P}_{x_n}(X_n(t) \in A_n). \tag{D.9}$$

We consider the two summands in (D.9) in turn. For the first summand, we note

$$\mathbb{P}_{x_n}(X_n(t) \in A_n) \leq \mathbb{P}_{x_n}(T_n(A_n) \leq t) \leq \mathbb{P}_{x_n}(T_n(A_n) \leq t_n - 1) < 2\delta,$$

where the second inequality holds for $t < t_n$ and the third by (D.8). It follows that

$$\alpha_n \sum_{t=0}^{t_n-1} (1-\alpha_n)^t \mathbb{P}_{x_n}(X_n(t) \in A_n) < 2\delta.$$

For the second summand in (D.9), we simply upper bound the probabilities by 1 to obtain

$$\alpha_n \sum_{t=t_n}^{\infty} (1-\alpha_n)^t \mathbb{P}_{x_n}(X_n(t) \in A_n) \leq (1-\alpha_n)^{t_n} \leq \exp(-\alpha_n t_n).$$

Taken together, we have shown $\tilde{\pi}_n(A_n) < 2\delta + \exp(-\alpha_n t_n)$. Combined with (D.8),

$$\|\pi_n - \tilde{\pi}_n\| \geq \pi_n(A_n) - \tilde{\pi}_n(A_n) > 1 - 3\delta - \exp(-\alpha_n t_n). \tag{D.10}$$

Next, applying Lemma D.3 with $\eta_1 = 1 - 2\delta$ and $\eta_2 = \eta_3 = \delta$, we obtain

$$t_{\text{mix}}^{(n)}(1-\delta) \leq t_n + \left\lceil \frac{t_{\text{rel}}^{(n)}}{2} \log\left(\frac{8}{1-2\delta}\right) \right\rceil,$$

which, after rearranging, yields

$$\frac{\alpha_n t_n}{\alpha_n t_{\text{mix}}^{(n)}(\delta)} \geq \frac{t_{\text{mix}}^{(n)}(1-\delta)}{t_{\text{mix}}^{(n)}(\delta)} - \frac{\left\lceil t_{\text{rel}}^{(n)} \log(8/(1-2\delta))/2 \right\rceil}{t_{\text{mix}}^{(n)}(\delta)}. \tag{D.11}$$

Now since pre-cutoff holds, $t_{\text{mix}}^{(n)}(1-\delta)/t_{\text{mix}}^{(n)}(\delta)$ is lower bounded by a positive constant as $n \to \infty$ (by definition of pre-cutoff) and $t_{\text{rel}}^{(n)} = o(t_{\text{mix}}^{(n)}(\delta))$ (by Lemma D.2). Hence,

$$\liminf_{n\to\infty} \frac{\alpha_n t_n}{\alpha_n t_{\text{mix}}^{(n)}(\delta)} > 0.$$

Since $\alpha_n t_{\text{mix}}^{(n)}(\delta) \to \infty$, this implies $\alpha_n t_n \to \infty$, so take $n \to \infty$ in (D.10). □

We now prove the $c = \infty$ case of the lemma. First, for $k \in \mathbb{N}$, let $\delta_k = 2^{-(k+1)}/3$; clearly, $\delta_k \in (0, 1/2)\ \forall\ k \in \mathbb{N}$. We claim $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\delta_k) = \infty\ \forall\ k \in \mathbb{N}$, which we prove as follows:

- If $\delta_k \leq \varepsilon$, then $t_{\text{mix}}^{(n)}(\delta_k) \geq t_{\text{mix}}^{(n)}(\varepsilon)$ by (5.4), so $\alpha_n t_{\text{mix}}^{(n)}(\delta_k) \to \infty$ by assumption.

- If $\delta_k > \varepsilon$ and $\varepsilon < 1/2$, then $\delta_k < 1/2 < 1 - \varepsilon$, so $t_{\text{mix}}^{(n)}(\delta_k) \geq t_{\text{mix}}^{(n)}(1-\varepsilon)$ by (5.4), and

$$\alpha_n t_{\text{mix}}^{(n)}(\delta_k) = \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \frac{t_{\text{mix}}^{(n)}(\delta_k)}{t_{\text{mix}}^{(n)}(\varepsilon)} \geq \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \frac{t_{\text{mix}}^{(n)}(1-\varepsilon)}{t_{\text{mix}}^{(n)}(\varepsilon)} \xrightarrow[n\to\infty]{} \infty,$$

  where the limit holds since $\alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$ by assumption and since $t_{\text{mix}}^{(n)}(1-\varepsilon)/t_{\text{mix}}^{(n)}(\varepsilon)$ is lower bounded by a positive constant as $n \to \infty$ by pre-cutoff.
- The final case, $\delta_k > \varepsilon$ and $\varepsilon \geq 1/2$, cannot occur, since $\delta_k < 1/2 \ \forall \ k \in \mathbb{N}$.

We have verified the conditions of Lemma D.4, so for each $k \in \mathbb{N}$ we can find $\{\tilde{P}_n^{(k)}\}_{n\in\mathbb{N}}$ s.t. $\tilde{P}_n^{(k)} \in B(P_n, \alpha_n) \ \forall \ n \in \mathbb{N}$, and, denoting the stationary distribution of $\tilde{P}_n^{(k)}$ by $\tilde{\pi}_n^{(k)}$,

$$\liminf_{n\to\infty} \|\pi_n - \tilde{\pi}_n^{(k)}\| \geq 1 - 3\delta_k = 1 - 2^{-(k+1)}. \tag{D.12}$$

Note that, as a consequence of (D.12), $\forall \ k \in \mathbb{N} \ \exists \ N_k \in \mathbb{N}$ s.t.

$$\|\pi_n - \tilde{\pi}_n^{(k)}\| > 1 - 2^{-k} \ \forall \ n \geq N_k. \tag{D.13}$$

We assume temporarily that $\lim_{k\to\infty} N_k = \infty$. Our goal is to use $\{\tilde{P}_n^{(k)}\}_{n,k\in\mathbb{N}}$ to construct $\{\tilde{P}_n\}_{n\in\mathbb{N}}$ satisfying the lemma statement. The construction proceeds as follows:

- If $n < \min_{k\in\mathbb{N}} N_k$, set $\tilde{P}_n = \tilde{P}_n^{(1)}$. (The choice $k = 1$ is arbitrary.)
- If $n \geq \min_{k\in\mathbb{N}} N_k$, let $k_n = \max\{k \in \mathbb{N} : n \geq N_k\}$ and set $\tilde{P}_n = \tilde{P}_n^{(k_n)}$. (Note $n \geq \min_{k\in\mathbb{N}} N_k$ guarantees $\{k \in \mathbb{N} : n \geq N_k\} \neq \emptyset$, while $N_k \to \infty$ guarantees $|\{k \in \mathbb{N} : n \geq N_k\}| < \infty$, so k_n is well-defined.)

Note that, since $\tilde{P}_n^{(k)} \in B(P_n, \alpha_n) \ \forall \ n, k \in \mathbb{N}$ by Lemma D.4, this construction guarantees $\tilde{P}_n \in B(P_n, \alpha_n) \ \forall \ n \in \mathbb{N}$ as well. Additionally, for $n \geq \min_{k\in\mathbb{N}} N_k$, we have $n \geq N_{k_n}$ by definition. Hence, because $\tilde{\pi}_n = \tilde{\pi}_n^{(k_n)}$ for all such n, we can use (D.13) to obtain

$$\|\pi_n - \tilde{\pi}_n\| = \|\pi_n - \tilde{\pi}_n^{(k_n)}\| > 1 - 2^{-k_n} \ \forall \ n \geq \min_{k\in\mathbb{N}} N_k \quad \Rightarrow \quad \liminf_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| \geq 1 - \limsup_{n\to\infty} 2^{-k_n}. \tag{D.14}$$

Thus, to complete the proof, it suffices to show $k_n \to \infty$ as $n \to \infty$. For this, let $M > 0$ and define $N^{(m)} = \max\{N_1, \ldots, N_{\lceil M \rceil}\}$. Then $k_n \geq \lceil M \rceil \geq M \ \forall \ n \geq N^{(m)}$, so since $M > 0$ was arbitrary, $k_n \to \infty$ as $n \to \infty$ follows. Thus, by (D.14), we obtain

$$\liminf_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| \geq 1 \geq \limsup_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| \quad \Rightarrow \quad \lim_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| = 1.$$

We now return to the case $\lim_{k\to\infty} N_k < \infty$. Here the construction is much simpler: we set $\tilde{P}_n = \tilde{P}_n^{(n)} \ \forall \ n \in \mathbb{N}$. Then for all n sufficiently large, $n \geq N_n$, so for such n,

$$\|\pi_n - \tilde{\pi}_n\| = \|\pi_n - \tilde{\pi}_n^{(n)}\| > 1 - 2^{-n},$$

from which it is clear that $\lim_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| = 1$.

D.3 Proof of Lemma 5.2

For the upper bound, let $\{\sigma_n\}_{n\in\mathbb{N}}$ be s.t. $\sigma_n \in \Delta_{n-1}\ \forall\ n \in \mathbb{N}$. Then $\forall\ n \in \mathbb{N}$,

$$\|\pi_n - \pi_{\alpha_n,\sigma_n}\| \leq \alpha_n \sum_{t=0}^{\infty}(1-\alpha_n)^t \|\pi_n - \sigma_n P_n^t\| \leq \alpha_n \sum_{t=0}^{\infty}(1-\alpha_n)^t d_n(t), \tag{D.15}$$

where we used Lemma D.2 and the PPR power iteration. Now since $d_n(t) \leq 1$, we can write

$$\begin{aligned}\alpha_n \sum_{t=0}^{\infty}(1-\alpha_n)^t d_n(t) &= \alpha_n \sum_{t=0}^{t_{\text{mix}}^{(n)}(\varepsilon)-1}(1-\alpha_n)^t d_n(t) + \alpha_n \sum_{t=t_{\text{mix}}^{(n)}(\varepsilon)}^{\infty}(1-\alpha_n)^t d_n(t) \\ &\leq 1 - (1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)} + \alpha_n \sum_{t=t_{\text{mix}}^{(n)}(\varepsilon)}^{\infty}(1-\alpha_n)^t d_n(t).\end{aligned} \tag{D.16}$$

We now consider the two cases of the bound in turn. First, assume $\varepsilon \in [1/2, 1)$. Then

$$\alpha_n \sum_{t=t_{\text{mix}}^{(n)}(\varepsilon)}^{\infty}(1-\alpha_n)^t d_n(t) \leq \varepsilon(1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)},$$

where we have used $d_n(t) \leq \varepsilon$ whenever $t \geq t_{\text{mix}}^{(n)}(\varepsilon)$. Thus, by (D.15) and (D.16), we obtain

$$\|\pi_n - \pi_{\alpha_n,\sigma_n}\| \leq 1 - (1-\varepsilon)(1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)} \xrightarrow[n\to\infty]{} 1 - (1-\varepsilon)e^{-c}. \tag{D.17}$$

Note this argument also holds for $\varepsilon \in (0, 1/2)$. Hence, for $\varepsilon \in (0, 1/2)$, it suffices to show

$$\limsup_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| \leq \frac{1-e^{-c}}{1-2\varepsilon e^{-c}}, \tag{D.18}$$

after which we can take a minimum over the bounds in (D.17) and (D.18) to complete the proof. To prove (D.18), we first bound the remaining summation in (D.16) as

$$\begin{aligned}\alpha_n \sum_{t=t_{\text{mix}}^{(n)}(\varepsilon)}^{\infty}(1-\alpha_n)^t d_n(t) &= \alpha_n \sum_{j=1}^{\infty} \sum_{t=jt_{\text{mix}}^{(n)}(\varepsilon)}^{(j+1)t_{\text{mix}}^{(n)}(\varepsilon)-1}(1-\alpha_n)^t d_n(t) \\ &\leq \alpha_n \sum_{j=1}^{\infty} d_n(jt_{\text{mix}}^{(n)}(\varepsilon)) \sum_{t=jt_{\text{mix}}^{(n)}(\varepsilon)}^{(j+1)t_{\text{mix}}^{(n)}(\varepsilon)-1}(1-\alpha_n)^t \\ &= \left(1-(1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)}\right) \sum_{j=1}^{\infty} d_n(jt_{\text{mix}}^{(n)}(\varepsilon))(1-\alpha_n)^{jt_{\text{mix}}^{(n)}(\varepsilon)},\end{aligned} \tag{D.19}$$

where the first equality is immediate, the inequality holds by monotonicity of d_n, and for the second equality we computed a geometric series. Now by definition, $d_n(t_{\text{mix}}^{(n)}(\varepsilon)) \leq \varepsilon < 2\varepsilon$. Furthermore, by Lemma D.2, $d_n(jt_{\text{mix}}^{(n)}(\varepsilon)) \leq (2\varepsilon)^j \ \forall \ j > 1$. We can therefore write

$$\sum_{j=1}^{\infty} d_n(jt_{\text{mix}}^{(n)}(\varepsilon))(1-\alpha_n)^{jt_{\text{mix}}^{(n)}(\varepsilon)} < \sum_{j=1}^{\infty} \left(2\varepsilon(1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)}\right)^j = \frac{2\varepsilon(1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)}}{1-2\varepsilon(1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)}}. \quad \text{(D.20)}$$

Hence, combining (D.15), (D.16), (D.19), and (D.20), we have ultimately shown

$$\begin{aligned}\|\pi_n - \tilde{\pi}_n\| &< \left(1-(1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)}\right)\left(1+\frac{2\varepsilon(1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)}}{1-2\varepsilon(1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)}}\right) \\ &= \frac{1-(1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)}}{1-2\varepsilon(1-\alpha_n)^{t_{\text{mix}}^{(n)}(\varepsilon)}} \xrightarrow[n\to\infty]{} \frac{1-e^{-c}}{1-2\varepsilon e^{-c}}.\end{aligned}$$

We turn to the lower bound. Similar to the $c = \infty$ case of Lemma 5.1, we begin with a weaker result. This result is almost identical to Lemma D.4; its proof is similar and leverages the stronger assumption of cutoff to obtain a useful bound when $\alpha_n t_{\text{mix}}^{(n)}(\delta) \to (0,\infty)$.

Lemma D.5. Let $P_n \in \mathcal{E}_n, \alpha_n \in (0,1) \ \forall \ n \in \mathbb{N}$, and let $\delta \in (0,1/2)$ be independent of n. Assume $\{P_n\}_{n\in\mathbb{N}}$ exhibits cutoff, each P_n is lazy and reversible, and $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\delta) = c \in (0,\infty)$. Then $\exists \ \{\tilde{P}_n\}_{n\in\mathbb{N}}$ s.t. $\tilde{P}_n \in B(P_n, \alpha_n) \ \forall \ n \in \mathbb{N}$ and $\liminf_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| \geq 1 - 3\delta - e^{-c}$.

Proof. By the argument preceding (D.10) in the Lemma D.4 proof, we obtain $\{\tilde{P}_n\}_{n\in\mathbb{N}}$ s.t.

$$\|\pi_n - \tilde{\pi}_n\| > 1 - 3\delta - \exp(-\alpha_n t_{\text{hit}}^{(n)}(1-\delta, 1-2\delta)).$$

Furthermore, by the same argument leading to (D.11) in the proof of Lemma D.4, we have

$$\frac{\alpha_n t_{\text{hit}}^{(n)}(1-\delta,1-2\delta)}{\alpha_n t_{\text{mix}}^{(n)}(\delta)} \geq \frac{t_{\text{mix}}^{(n)}(1-\delta)}{t_{\text{mix}}^{(n)}(\delta)} - \frac{\left\lceil t_{\text{rel}}^{(n)} \log(8/(1-2\delta))/2 \right\rceil}{t_{\text{mix}}^{(n)}(\delta)}.$$

Now when cutoff holds, $t_{\text{mix}}^{(n)}(1-\delta)/t_{\text{mix}}^{(n)}(\delta) \to 1$ (by definition) and $t_{\text{rel}}^{(n)}/t_{\text{mix}}^{(n)}(\delta) \to 0$ (by Lemma D.2) as $n \to \infty$. Hence, by assumption $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\delta) = c$, we conclude

$$\liminf_{n\to\infty} \alpha_n t_{\text{hit}}^{(n)}(1-\delta,1-2\delta) \geq c.$$

To summarize, we have shown

$$\liminf_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| \geq 1 - 3\delta - \exp(-\liminf_{n\to\infty} \alpha_n t_{\text{hit}}^{(n)}(1-\delta,1-2\delta)) \geq 1 - 3\delta - e^{-c}. \qquad \square$$

We use Lemma D.5 to prove the lower bound in Lemma 5.2 in the manner we used Lemma D.4 to prove the $c = \infty$ case of Lemma 5.1. First, for $k \in \mathbb{N}$, let $\delta_k = 2^{-(k+1)}/3 \in (0,1/2)$; we

claim $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\delta_k) = c$. To prove this, first note (provided the limits exist in $(0,\infty)$)

$$\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\delta_k) = \lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \lim_{n\to\infty} t_{\text{mix}}^{(n)}(\delta_k)/t_{\text{mix}}^{(n)}(\varepsilon) = c \lim_{n\to\infty} t_{\text{mix}}^{(n)}(\delta_k) t_{\text{mix}}^{(n)}(\varepsilon),$$

so it suffices to show $\lim_{n\to\infty} t_{\text{mix}}^{(n)}(\delta_k)/t_{\text{mix}}^{(n)}(\varepsilon) = 1$. This can be proven as follows:

- If $\varepsilon \le \delta_k \le 1-\varepsilon$, we have $t_{\text{mix}}^{(n)}(1-\varepsilon) \le t_{\text{mix}}^{(n)}(\delta_k) \le t_{\text{mix}}^{(n)}(\varepsilon)$ by (5.4), so by cutoff,

$$1 = \lim_{n\to\infty} \frac{t_{\text{mix}}^{(n)}(1-\varepsilon)}{t_{\text{mix}}^{(n)}(\varepsilon)} \le \lim_{n\to\infty} \frac{t_{\text{mix}}^{(n)}(\delta_k)}{t_{\text{mix}}^{(n)}(\varepsilon)} \le \lim_{n\to\infty} \frac{t_{\text{mix}}^{(n)}(\varepsilon)}{t_{\text{mix}}^{(n)}(\varepsilon)} = 1.$$

- If $\delta_k \le \varepsilon \le 1-\varepsilon$, we have $t_{\text{mix}}^{(n)}(1-\delta_k) \le t_{\text{mix}}^{(n)}(\varepsilon) \le t_{\text{mix}}^{(n)}(\delta_k)$ by (5.4), so by cutoff,

$$1 = \lim_{n\to\infty} \frac{t_{\text{mix}}^{(n)}(1-\delta_k)}{t_{\text{mix}}^{(n)}(\delta_k)} \le \lim_{n\to\infty} \frac{t_{\text{mix}}^{(n)}(\varepsilon)}{t_{\text{mix}}^{(n)}(\delta_k)} \le \lim_{n\to\infty} \frac{t_{\text{mix}}^{(n)}(\varepsilon)}{t_{\text{mix}}^{(n)}(\varepsilon)} = 1.$$

- If $1-\varepsilon \le \delta_k \le \varepsilon$ or $\delta_k \le 1-\varepsilon \le \varepsilon$, the result holds by reversing the roles of ε and $1-\varepsilon$.
- Finally, $\varepsilon \le 1-\varepsilon \le \delta_k$ and $1-\varepsilon \le \varepsilon \le \delta_k$ cannot occur since $\delta_k < 1/2$.

We have shown $\delta_k \in (0,1/2)$ and $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\delta_k) = c \ \forall\ k \in \mathbb{N}$. Hence, for each k, we can use Lemma D.5 to find $\{\tilde{P}_n^{(k)}\}_{n\in\mathbb{N}}$ s.t. $\tilde{P}_n^{(k)} \in B(P_n, \alpha_n) \ \forall\ n \in \mathbb{N}$, and

$$\liminf_{n\to\infty} \|\pi_n - \tilde{\pi}_n^{(k)}\| \ge 1 - e^{-c} - 2^{-(k+1)}.$$

From here, proof can be completed in a similar manner as the $c = \infty$ case of Lemma 5.1, by replacing 1 with $1-e^{-c}$ in the analysis following (D.12).

D.4 Proof of Theorem 5.1

The lower bounds (5.13) and (5.14) follow from the lower bounds in Lemmas 5.1 and 5.2. Hence, we only need to prove the upper bounds (5.11) and (5.12). Towards this end, first assume $\varepsilon = 1/4$; we will then extend the proof to the case $\varepsilon \ne 1/4$. In the case $\varepsilon = 1/4$ (in fact, any $\varepsilon < 1/2$), (5.11) follows immediately from the upper bound in Lemma 5.1. To prove (5.12), assume for the sake of contradiction $\exists\ \{\sigma_n\}_{n\in\mathbb{N}}$ with $\sigma_n \in \Delta_{n-1} \ \forall\ n \in \mathbb{N}$ and

$$\limsup_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| > 1 - e^{-c}.$$

If this inequality holds, then the interval

$$\Big(0, \min\Big\{1/4, e^c\Big(\limsup_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| - (1-e^{-c})\Big)\Big\}\Big)$$

is nonempty, so we can choose δ in this interval. Since $\delta < 1/4$ by construction, (5.4) implies

$$\alpha_n t_{\text{mix}}^{(n)} \le \alpha_n t_{\text{mix}}^{(n)}(\delta) = \alpha_n t_{\text{mix}}^{(n)}(1-\delta) \frac{t_{\text{mix}}^{(n)}(\delta)}{t_{\text{mix}}^{(n)}(1-\delta)} \le \alpha_n t_{\text{mix}}^{(n)} \frac{t_{\text{mix}}^{(n)}(\delta)}{t_{\text{mix}}^{(n)}(1-\delta)}.$$

Hence, using the definition of c and the cutoff assumption,

$$c = \lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)} \le \lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\delta) \le \lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)} \times \lim_{n\to\infty} \frac{t_{\text{mix}}^{(n)}(\delta)}{t_{\text{mix}}^{(n)}(1-\delta)} = c \times 1 = c, \qquad \text{(D.21)}$$

so that $\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\delta) = c$. Assuming for the moment that $t_{\text{mix}}^{(n)}(\delta) \to \infty$, we can then use (5.10) and the choice of δ to obtain

$$\limsup_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| \le 1 - e^{-c} + \delta e^{-c} < \limsup_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\|,$$

which is a contradiction. Now to see why $t_{\text{mix}}^{(n)}(\delta) \to \infty$ holds, first note that $\forall\ \delta' \in (0,\delta)$,

$$t_{\text{mix}}^{(n)}(\delta) = t_{\text{mix}}^{(n)}(\delta') \frac{t_{\text{mix}}^{(n)}(\delta)}{t_{\text{mix}}^{(n)}(\delta')} \ge t_{\text{mix}}^{(n)}(\delta') \frac{t_{\text{mix}}^{(n)}(1-\delta')}{t_{\text{mix}}^{(n)}(\delta')},$$

where the inequality holds by (5.4). Hence, by cutoff, we obtain $\forall\ \delta' \in (0,\delta)$,

$$\liminf_{n\to\infty} t_{\text{mix}}^{(n)}(\delta) \ge \liminf_{n\to\infty} t_{\text{mix}}^{(n)}(\delta').$$

On the other hand, the previous inequality immediately holds $\forall\ \delta' \in [\delta, 1)$ by (5.4). Therefore,

$$\liminf_{n\to\infty} t_{\text{mix}}^{(n)}(\delta) \ge \sup_{\delta'\in(0,1)} \liminf_{n\to\infty} t_{\text{mix}}^{(n)}(\delta') = \infty,$$

where the equality holds by Lemma D.2.

Finally, we extend the upper bounds to $\varepsilon \neq 1/4$, for which it suffices to show

$$\lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) = c \quad \Rightarrow \quad \lim_{n\to\infty} \alpha_n t_{\text{mix}}^{(n)} = c, \qquad \text{(D.22)}$$

after which we can invoke the result from the case $\varepsilon = 1/4$ to complete the proof. (D.22) is an almost direct consequence of cutoff. To prove it, we first use (5.4) to obtain

$$\begin{aligned}
\varepsilon \in (0,1/4) \Rightarrow \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \ge \alpha_n t_{\text{mix}}^{(n)} = \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \frac{t_{\text{mix}}^{(n)}}{t_{\text{mix}}^{(n)}(\varepsilon)} \ge \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \frac{t_{\text{mix}}^{(n)}(1-\varepsilon)}{t_{\text{mix}}^{(n)}(\varepsilon)}, \\
\varepsilon \in (1/4,3/4] \Rightarrow \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \le \alpha_n t_{\text{mix}}^{(n)} = \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \frac{t_{\text{mix}}^{(n)}}{t_{\text{mix}}^{(n)}(\varepsilon)} \le \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \frac{t_{\text{mix}}^{(n)}}{t_{\text{mix}}^{(n)}(3/4)}, \\
\varepsilon \in (3/4,1) \Rightarrow \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \le \alpha_n t_{\text{mix}}^{(n)} = \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \frac{t_{\text{mix}}^{(n)}}{t_{\text{mix}}^{(n)}(\varepsilon)} \le \alpha_n t_{\text{mix}}^{(n)}(\varepsilon) \frac{t_{\text{mix}}^{(n)}(1-\varepsilon)}{t_{\text{mix}}^{(n)}(\varepsilon)}.
\end{aligned}$$

Now letting $n \to \infty$ and using cutoff in the three cases, (D.22) follows as in (D.21).

D.5 Proof of Theorem 5.2

For convenience, we restate the definition of "coincides with" from the main text: a sequence $\{\alpha_{n,\varepsilon}\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)} \subset (0,1)$ coincides with $\{t_{\text{mix}}^{(n)}(\varepsilon)\}_{n\in\mathbb{N},\varepsilon\in(0,1)}$ if

$$\sup_{\varepsilon\in(0,1/2)} \liminf_{n\to\infty} \alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(\varepsilon) = \infty, \frac{\alpha_{n,\varepsilon}}{\alpha_{n,\delta}} \in \left[\frac{t_{\text{mix}}^{(n)}(1-\delta)}{t_{\text{mix}}^{(n)}(1-\varepsilon)}, 1\right] \ \forall\ \varepsilon,\delta\in(0,1/2) \text{ s.t. } \varepsilon\geq\delta, \forall\ n\in\mathbb{N}. \tag{D.23}$$

Such sequences always exist for lazy chains. In particular, if $c \in (0,1)$ is independent of n and ε, and if $\alpha_{n,\varepsilon} = c\ \forall\ n\in\mathbb{N}, \varepsilon\in(0,1/2)$, then $\{\alpha_{n,\varepsilon}\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)}$ satisfies (D.23). To see why, note that the first condition in (D.23) follows immediately from Lemma D.2; for the second condition, the upper bound clearly holds, and the interval is nonempty by (5.4).

We now prove the crucial property that was discussed in Section 5.4.

Lemma D.6. If pre-cutoff holds and $\{\alpha_{n,\varepsilon}\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)} \subset (0,1)$ coincides with the mixing times $\{t_{\text{mix}}^{(n)}(\varepsilon)\}_{n\in\mathbb{N},\varepsilon\in(0,1)}$, then $\forall\ \varepsilon\in(0,1/2)$, $\lim_{n\to\infty} \alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(\varepsilon) = \infty$.

Proof. Let $\varepsilon\in(0,1/2)$; we aim to show $\alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$. Fix $n\in\mathbb{N}$. Then $\forall\ \delta\in(0,\varepsilon]$,

$$\alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(\varepsilon) \geq \alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(1-\varepsilon) \geq \alpha_{n,\delta} t_{\text{mix}}^{(n)}(1-\delta) = \alpha_{n,\delta} t_{\text{mix}}^{(n)}(\delta) \frac{t_{\text{mix}}^{(n)}(1-\delta)}{t_{\text{mix}}^{(n)}(\delta)}, \tag{D.24}$$

where the first inequality holds by (5.4) (since $\varepsilon < 1/2$), and the second holds by the lower bound of the interval in (D.23). On the other hand, $\forall\ \delta\in[\varepsilon,1/2)$,

$$\alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(\varepsilon) \geq \alpha_{n,\delta} t_{\text{mix}}^{(n)}(\delta) \geq \alpha_{n,\delta} t_{\text{mix}}^{(n)}(\delta) \frac{t_{\text{mix}}^{(n)}(1-\delta)}{t_{\text{mix}}^{(n)}(\delta)}, \tag{D.25}$$

where the first inequality holds by the upper bound of the interval in (D.23) and by (5.4), and the second by (5.4) (since $\delta < 1/2$). Now $n\in\mathbb{N}$ was arbitrary, so (D.24) and (D.25) imply

$$\liminf_{n\to\infty} \alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(\varepsilon) \geq \liminf_{n\to\infty} \alpha_{n,\delta} t_{\text{mix}}^{(n)}(\delta) \frac{t_{\text{mix}}^{(n)}(1-\delta)}{t_{\text{mix}}^{(n)}(\delta)} \ \forall\ \delta\in(0,1/2).$$

Also, by definition of pre-cutoff, $\exists\ K>0$ independent of n,δ such that $\forall\ \delta\in(0,1/2)$,

$$\begin{aligned}\liminf_{n\to\infty} \alpha_{n,\delta} t_{\text{mix}}^{(n)}(\delta) \frac{t_{\text{mix}}^{(n)}(1-\delta)}{t_{\text{mix}}^{(n)}(\delta)} &\geq \liminf_{n\to\infty} \frac{t_{\text{mix}}^{(n)}(1-\delta)}{t_{\text{mix}}^{(n)}(\delta)} \liminf_{n\to\infty} \alpha_{n,\delta} t_{\text{mix}}^{(n)}(\delta) \\ &\geq K \liminf_{n\to\infty} \alpha_{n,\delta} t_{\text{mix}}^{(n)}(\delta).\end{aligned}$$

Combining the previous two bounds, and since these bounds hold $\forall\ \delta\in(0,1/2)$,

$$\liminf_{n\to\infty} \alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(\varepsilon) \geq K \sup_{\delta\in(0,1/2)} \liminf_{n\to\infty} \alpha_{n,\delta} t_{\text{mix}}^{(n)}(\delta) = \infty,$$

where the equality holds by (D.23). □

We turn to the proof of the theorem. First, we show pre-cutoff implies Condition 5.1. For this, let $\{\alpha_{n,\varepsilon}\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)} \subset (0,1)$ coincide with $\{t_{\text{mix}}^{(n)}(\varepsilon)\}_{n\in\mathbb{N},\varepsilon\in(0,1)}$, and fix $\varepsilon \in (0,1/2)$. Lemma D.6 ensures $\alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(\varepsilon) \to \infty$; hence, by Lemma 5.1, $\exists\ \{\sigma_{n,\varepsilon}\}_{n\in\mathbb{N}}$ s.t.

$$\sigma_{n,\varepsilon} \in \Delta_{n-1}\ \forall\ n \in \mathbb{N}, \quad \lim_{n\to\infty} \|\pi_n - \pi_{\alpha_{n,\varepsilon},\sigma_{n,\varepsilon}}\| = 1.$$

Next, assume (5.15) holds and set $\alpha_{n,\varepsilon} = 1/(2t_{\text{mix}}^{(n)}(1-\varepsilon))\ \forall\ n \in \mathbb{N}, \varepsilon \in (0,1/2)$. Then

$$\frac{\alpha_{n,\varepsilon}}{\alpha_{n,\delta}} = \frac{t_{\text{mix}}^{(n)}(1-\delta)}{t_{\text{mix}}^{(n)}(1-\varepsilon)}\ \forall\ \varepsilon,\delta \in (0,1/2).$$

Furthermore, since (5.15) holds by assumption,

$$\sup_{\varepsilon\in(0,1/2)} \liminf_{n\to\infty} \alpha_{n,\varepsilon} t_{\text{mix}}^{(n)}(\varepsilon) = \frac{1}{2} \sup_{\varepsilon\in(0,1/2)} \liminf_{n\to\infty} \frac{t_{\text{mix}}^{(n)}(\varepsilon)}{t_{\text{mix}}^{(n)}(1-\varepsilon)} = \infty.$$

The previous two lines show that $\{\alpha_{n,\varepsilon}\}_{n\in\mathbb{N},\varepsilon\in(0,1/2)}$ coincides with $\{t_{\text{mix}}^{(n)}(\varepsilon)\}_{n\in\mathbb{N},\varepsilon\in(0,1)}$. Fixing $\varepsilon \in (0,1/2)$ and $\{\sigma_{n,\varepsilon}\}_{n\in\mathbb{N}}$ s.t. $\sigma_{n,\varepsilon} \in \Delta_{n-1}\ \forall\ n \in \mathbb{N}$, we can then use (D.15) to obtain

$$\begin{aligned}
\|\pi_n - \pi_{\alpha_{n,\varepsilon},\sigma_{n,\varepsilon}}\| &\le \alpha_{n,\varepsilon} \sum_{t=0}^{t_{\text{mix}}^{(n)}(1-\varepsilon)-1} (1-\alpha_{n,\varepsilon})^t d_n(t) + \alpha_{n,\varepsilon} \sum_{t=t_{\text{mix}}^{(n)}(1-\varepsilon)}^{\infty} (1-\alpha_{n,\varepsilon})^t d_n(t) \\
&\le \alpha_{n,\varepsilon} \sum_{t=0}^{t_{\text{mix}}^{(n)}(1-\varepsilon)-1} (1-\alpha_{n,\varepsilon})^t + \alpha_{n,\varepsilon} \sum_{t=t_{\text{mix}}^{(n)}(1-\varepsilon)}^{\infty} (1-\alpha_{n,\varepsilon})^t (1-\varepsilon) \\
&= 1 - \varepsilon(1-\alpha_{n,\varepsilon})^{t_{\text{mix}}^{(n)}(1-\varepsilon)} = 1 - \varepsilon\left(1 - \frac{1/2}{t_{\text{mix}}^{(n)}(1-\varepsilon)}\right)^{t_{\text{mix}}^{(n)}(1-\varepsilon)} \le 1 - \frac{\varepsilon}{2},
\end{aligned}$$

where the final inequality is Bernoulli's. Since $\varepsilon, \{\sigma_{n,\varepsilon}\}_{n\in\mathbb{N}}$ were arbitrary, Condition 5.1 fails.

D.6 Proof of Proposition 5.1

D.6.1 Winning streak reversal

For the winning streak reversal, most of the arguments are recounted from Section 4.6 of [99]. First, for $i \in [n-1]$, note the chain started from i reaches stationarity in i steps, i.e.

$$e_i P_n^i = e_i P_n^{i-1} P_n = e_1 P_n = \pi_n.$$

It remains to analyze the chain starting from n. First, we claim that for $j \in [n-1]$,

$$e_n P_n^j = \sum_{i=1}^{j} 2^{i-j-1} e_{n-i} + 2^{-j} e_n. \tag{D.26}$$

This claim can be proven inductively: for $j = 1$, the left side of (D.26) is $(e_{n-1} + e_n)/2$ by (5.19), while the right side of (D.26) is clearly $(e_{n-1} + e_n)/2$; assuming true for j, we have

$$\begin{aligned} e_n P_n^{j+1} &= e_n P_n P_n^j = \frac{1}{2}(e_{n-1} + e_n) P_n^j = \frac{1}{2} e_{n-1-j} + \frac{1}{2}\left(\sum_{i=1}^{j} 2^{i-j-1} e_{n-i} + 2^{-j} e_n\right) \\ &= 2^{-1} e_{n-(j+1)} + \sum_{i=1}^{j} 2^{i-(j+1)-1} e_{n-i} + 2^{-(j+1)} e_n = \sum_{i=1}^{j+1} 2^{i-(j+1)-1} e_{n-i} + 2^{-(j+1)} e_n, \end{aligned}$$

which establishes (D.26). Now taking $j = n - 1$ in (D.26), we obtain

$$e_n P_n^{n-1} = \sum_{i=1}^{n-1} 2^{i-(n-1)-1} e_{n-i} + 2^{-(n-1)} e_n = 2^{-1} e_1 + \cdots + 2^{-(n-1)} e_{n-1} + 2^{-(n-1)} e_n = \pi_n.$$

To summarize, we have shown $e_i P_n^i = \pi_n \ \forall \ i \in [n-1]$ and $e_i P_n^{n-1} = \pi_n$, which implies

$$d_n(n-1) = \max_{i \in [n]} \| e_i P_n^{n-1} - \pi_n \| = 0 \quad \Rightarrow \quad t_{\text{mix}}^{(n)}(1-\varepsilon), t_{\text{mix}}^{(n)}(\varepsilon) \leq n - 1.$$

For a lower bound on the ε-mixing time, note that, by (D.26), $P_n^{n-2}(n, 1) = 0$, where $P_n^{n-2}(n, 1)$ is the $(n, 1)$-th element of P_n^{n-2}. Hence, we immediately obtain

$$d_n(n-2) \geq \| e_n P_n^{n-2} - \pi_n \| \geq \pi_n(1) - P_n^{n-2}(n, 1) = \frac{1}{2} > \varepsilon \quad \Rightarrow \quad t_{\text{mix}}^{(n)}(\varepsilon) > n - 2,$$

so, combining with the above, we conclude $t_{\text{mix}}^{(n)}(\varepsilon) = n - 1$. Finally, to lower bound the $(1-\varepsilon)$-mixing time, first note that for any $t \in \{0, \ldots, n-2\}$, we have $e_{n-1} P_n^t = e_{n-1-t}$, so

$$d_n(t) \geq \| e_{n-1} P_n^t - \pi_n \| = \| e_{n-1-t} - \pi_n \| \geq 1 - \pi_n(n - 1 - t) = 1 - 2^{-n+1+t}.$$

Hence, for $t < n - 1 - \log_2(1/\varepsilon)$, we obtain

$$d_n(t) \geq 1 - 2^{-n+1+t} > 1 - 2^{-\log_2(1/\varepsilon)} = 1 - \varepsilon \quad \Rightarrow \quad t_{\text{mix}}^{(n)}(1-\varepsilon) \geq n - 1 - \log_2(1/\varepsilon).$$

D.6.2 Complete graph bijection

For the complete graph bijection, we denote by $N(i)$ the neighbors of $i \in [n]$ in the underlying graph, i.e.

$$N(i) = \begin{cases} \{1, \ldots, i-1, i+1, \ldots, n/2, i+n/2\}, & n \text{ even}, i \leq n/2 \\ \{i - n/2, 1 + n/2, \ldots, i-1, i+1, \ldots, n\}, & n \text{ even}, i > n/2 \\ \{1, \ldots, i-1, i+1, \ldots, (n-1)/2, i + (n-1)/2, n\}, & n \text{ odd}, i \leq (n-1)/2 \\ \{i - (n-1)/2, 1 + (n-1)/2, \ldots, i-1, i+1, \ldots, n\}, & n \text{ odd}, (n-1)/2 < i < n \\ \{1, \ldots, n-1\}, & n \text{ odd}, i = n \end{cases}.$$

As an example, for the $n = 6$ graph in Figure 5.2b, we have

$$N(1) = \{2, 3, 4\}, \quad N(2) = \{1, 3, 5\}, \quad N(3) = \{1, 2, 6\},$$
$$N(4) = \{1, 5, 6\}, \quad N(5) = \{2, 4, 6\}, \quad N(6) = \{3, 4, 5\},$$

while for the $n = 7$ graph in the same figure, we have

$$N(1) = \{2, 3, 4, 7\}, \quad N(2) = \{1, 3, 5, 7\}, \quad N(3) = \{1, 2, 6, 7\},$$
$$N(4) = \{1, 5, 6, 7\}, \quad N(5) = \{2, 4, 6, 7\}, \quad N(6) = \{3, 4, 5, 7\},$$
$$N(7) = \{1, 2, 3, 4, 5, 6\}.$$

We now show $t_{\text{mix}}^{(n)}(1-\varepsilon) = 1$ for n large. For n even, we have by Lemma D.1 and (5.20)-(5.21),

$$\begin{aligned}
&2\|e_1 P_n - \pi_n\| \\
&\quad = |P_n(1,1) - \pi_n(1)| + \sum_{j \in N(1)} |P_n(1,j) - \pi_n(j)| + \sum_{j \in [n] \setminus (\{1\} \cup N(1))} |P_n(1,j) - \pi_n(j)| \\
&\quad = \left|\frac{1}{2} - \frac{1}{n}\right| + \frac{n}{2}\left|\frac{1}{n} - \frac{1}{n}\right| + \left(\frac{n}{2} - 1\right)\left|0 - \frac{1}{n}\right| \xrightarrow[n\to\infty]{} 1,
\end{aligned}$$

so, by symmetry, $\max_{i \in [n]} \|e_i P_n - \pi_n\| \to 1/2$ along even n. If n is odd, we similarly have

$$\begin{aligned}
2\|e_1 P_n - \pi_n\| &= |P_n(1,1) - \pi_n(1)| + |P_n(1,n) - \pi_n(n)| \\
&\quad + \sum_{j \in N(1) \setminus \{n\}} |P_n(1,j) - \pi_n(j)| + \sum_{j \in [n] \setminus (\{1\} \cup N(1))} |P_n(1,j) - \pi_n(j)| \\
&= \left|\frac{1}{2} - \frac{n+1}{(n+3)(n-1)}\right| + \left|\frac{1}{n+1} - \frac{2}{n+3}\right| \\
&\quad + \frac{n-1}{2}\left|\frac{1}{n+1} - \frac{n+1}{(n+3)(n-1)}\right| + \frac{n-3}{2}\left|0 - \frac{n+1}{(n+3)(n-1)}\right| \xrightarrow[n\to\infty]{} 1,
\end{aligned}$$

so, by symmetry, $\max_{i \in [n-1]} \|e_i P_n - \pi_n\| \to 1/2$ along odd n. We also note

$$\begin{aligned}
2\|e_n P_n - \pi_n\| &= \sum_{j=1}^{n-1} |P_n(n,j) - \pi_n(j)| + |P_n(n,n) - \pi_n| \\
&= (n-1)\left|\frac{1}{2(n-1)} - \frac{n+1}{(n+3)(n-1)}\right| + \left|\frac{1}{2} - \frac{2}{n+3}\right| \xrightarrow[n\to\infty]{} \frac{1}{2},
\end{aligned}$$

so $\max_{i \in [n]} \|e_i P_n - \pi_n\| \to 1/2$ along odd n. Combined with the analysis for n even,

$$\limsup_{n\to\infty} d_n(1) = \limsup_{n\to\infty} \max_{i \in [n]} \|e_i P_n - \pi_n\| \le \frac{1}{2} < 1 - \varepsilon, \tag{D.27}$$

so $t_{\text{mix}}^{(n)}(1-\varepsilon) \leq 1$ for large n. Finally, by the discussion in Section 5.2, we also know $t_{\text{mix}}^{(n)}(1-\varepsilon) \neq 0$ for large n, so we conclude $t_{\text{mix}}^{(n)}(1-\varepsilon) = 1$ for such n.

We next show $t_{\text{mix}}^{(n)}(\varepsilon) = \Theta(n)$. We begin with the easier proof, $t_{\text{mix}}^{(n)}(\varepsilon) = \Omega(n)$. For n even, the intuition is that the stationary distribution places equal weight on both cliques, whereas the distribution of $X_n(t)$ is biased towards $[n/2]$ if $X_n(0) = 1$. Hence, we write

$$d_n(t) \geq \|e_1 P_n^t - \pi_n\| \geq P_n^t(1, [n/2]) - \pi_n([n/2]) = P_n^t(1, [n/2]) - \frac{1}{2}, \tag{D.28}$$

where $P_n^t(i,j)$ is the (i,j)-th element of P_n^t for $i, j \in [n]$ and $P_n^t(i, A) = \sum_{j \in A} P_n^t(i,j)$ for $A \subset [n]$. It remains to lower bound $P_n^t(1, [n/2])$. For this, we claim

$$P_n^t(i, [n/2]) \geq \left(1 - \frac{1}{n}\right)^t \ \forall\ t \in \mathbb{Z}_+, i \in [n/2]. \tag{D.29}$$

We prove (D.29) by induction. For $t = 0$, (D.29) is immediate. Assuming (D.29) holds for t,

$$\begin{aligned} P_n^{t+1}(i, [n/2]) &= \sum_{k \in [n]} P_n(i,k) P_n^t(k, [n/2]) \geq \sum_{k \in [n/2]} P_n(i,k) P_n^t(k, [n/2]) \\ &\geq \left(1 - \frac{1}{n}\right)^t P_n(i, [n/2]) = \left(1 - \frac{1}{n}\right)^{t+1}, \end{aligned}$$

where the first inequality holds by nonnegativity, the second inequality is the inductive hypothesis, and the last equality holds by (5.20). This proves (D.29). Substituting into (D.28),

$$d_n(t) \geq \left(1 - \frac{1}{n}\right)^t - \frac{1}{2} \geq \left(1 - \frac{t}{n}\right) - \frac{1}{2} = \frac{1}{2} - \frac{t}{n},$$

where we have also used Bernoulli's inequality. The following is then immediate:

$$t < n\left(\frac{1}{2} - \varepsilon\right) \quad \Rightarrow \quad d_n(t) > \varepsilon \quad \Rightarrow \quad t_{\text{mix}}^{(n)}(\varepsilon) > n\left(\frac{1}{2} - \varepsilon\right). \tag{D.30}$$

We next assume n is odd. Here the argument is nearly identical: since by (5.20),

$$P_n\left(i, \left[\frac{n-1}{2}\right]\right) = 1 - \frac{2}{n+1} \ \forall\ i \in \left[\frac{n-1}{2}\right],$$

we can use an inductive argument as above to obtain

$$P_n^t\left(1, \left[\frac{n-1}{2}\right]\right) \geq \left(1 - \frac{2}{n+1}\right)^t \ \forall\ t \in \mathbb{Z}_+.$$

On the other hand, by (5.20) we have

$$\pi_n\left(\left[\frac{n-1}{2}\right]\right) = \frac{n-1}{2} \frac{n+1}{(n-1)(n+3)} \leq \frac{1}{2}.$$

Hence, combining the previous two lines, and using Bernoulli's inequality, we obtain

$$d_n(t) \geq \|e_1 P_n^t - \pi_n\| \geq \frac{1}{2} - \frac{2t}{n+1}.$$

The following implications are then immediate:

$$t < \frac{n+1}{2}\left(\frac{1}{2} - \varepsilon\right) \quad \Rightarrow \quad d_n(t) > \varepsilon \quad \Rightarrow \quad t_{\text{mix}}^{(n)}(\varepsilon) > \frac{n+1}{2}\left(\frac{1}{2} - \varepsilon\right). \tag{D.31}$$

Combining (D.30) and (D.31), we conclude $t_{\text{mix}}^{(n)}(\varepsilon) = \Omega(n)$.

For the remainder of the proof, we aim to show $t_{\text{mix}}^{(n)} = O(n)$, for which we use couplings.[2] More specifically, by Lemmas D.1 and D.2, we aim to bound

$$d_n(t) \leq \max_{i,j \in [n]} \|e_i P_n^t - e_j P_n^t\| \leq \max_{i,j \in [n]} \mathbb{P}_{ij}(X_n(t) \neq Y_n(t)), \tag{D.32}$$

where $\{X_n(t)\}_{t \in \mathbb{Z}_+}$ and $\{Y_n(t)\}_{t \in \mathbb{Z}_+}$, respectively, are Markov chains with transition matrix P_n starting from $X_n(0) = i$ and $Y_n(0) = j$, respectively (as denoted by the subscript in $\mathbb{P}_{ij}$). For n even, we will refer to the sets $\{1, \ldots, n/2\}$ and $\{1 + n/2, \ldots, n\}$ as cliques (since these sets form complete subgraphs in the underlying graph); similarly, for n odd, we will call the sets $\{1, \ldots, (n-1)/2\}$ and $\{1 + (n-1)/2, \ldots, n-1\}$ cliques.

We begin with the case where n is even. Our approach is to first bring the two chains to the same clique, after which they remain in the same clique forever. Once the chains are in the same clique, we bring them to the same state, after which they remain in the same state forever. More specifically, given $X_n(t), Y_n(t)$, we assign $X_n(t+1), Y_n(t+1)$ as follows:

(A) If $X_n(t) \neq Y_n(t)$, proceed to (B). Otherwise, let $X_n(t+1) \sim e_{X_n(t)} P_n$ and set $Y_n(t+1) = X_n(t+1)$ (i.e. run the chains together).

(B) If $X_n(t), Y_n(t)$ are in the same clique, proceed to (C). Otherwise, flip an independent fair coin. If heads, sample $X_n(t+1)$ from $N(X_n(t))$ uniformly (i.e. move this chain) and set $Y_n(t+1) = Y_n(t)$ (i.e. keep this chain lazy). If tails, set $X_n(t+1) = X_n(t)$ (i.e. keep lazy) and sample $Y_n(t+1)$ from $N(Y_n(t))$ uniformly (i.e. move).[3]

(C) Flip an independent fair coin. If heads, set $X_n(t+1) = X_n(t), Y_n(t+1) = Y_n(t)$ (i.e. keep both chains lazy). If tails, roll a three-sided die that lands 1, 2, and 3 with probability $\frac{2}{n}$, $\frac{2}{n}$, and $1 - \frac{4}{n}$, respectively, and proceed as follows:

- If 1, define $X_n(t+1), Y_n(t+1)$ as follows (i.e. move to the other clique):

$$(X_n(t+1), Y_n(t+1)) = \begin{cases} (X_n(t) + n/2, Y_n(t) + n/2), & X_n(t) \leq n/2 \\ (X_n(t) - n/2, Y_n(t) - n/2), & X_n(t) > n/2 \end{cases}.$$

- If 2, set $X_n(t+1) = Y_n(t), Y_n(t+1) = X_n(t)$ (i.e. swap the chains).[4]

[2] Note this bound is order optimal in the sense that is matches the $\Omega(n)$ lower bound. Hence, while some intermediate bounds may seem needlessly loose, this is not a major concern.

[3] By moving only one chain, we ensure the chains do not switch cliques, i.e. we prevent e.g. the case $X_n(t) \in \{1, \ldots, n/2\}, Y_n(t) \in \{1 + n/2, \ldots, n\}$ and $X_n(t+1) \in \{1 + n/2, \ldots, n\}, Y_n(t+1) \in \{1, \ldots, n/2\}$.

[4] When the die is 2 or 3, both chains move within the clique. By swapping the chains when the die is 2, we

- If 3, sample $X_n(t+1)$ uniformly from $N(X_n(t)) \setminus \{Y_n(t)\}$, set $Y_n(t+1) = X_n(t+1)$.

To analyze this, first suppose $X_n(0) = i, Y_n(0) = j$ for some $i \neq j$ in the same clique. Then $X_n(t) \neq Y_n(t)$ implies that at each step $\tau \in \{0, \dots, t-1\}$, one of the following occur:

- The coin in (C) lands heads, so both chains are lazy. This occurs with probability $1/2$.
- The coin in (C) lands tails and the die in (C) lands 1 or 2, so that both chains move, but to different states. This occurs with probability $(1/2) \times (4/n) = 2/n$.

By independence of these coin flips and die rolls, it follows that

$$\mathbb{P}_{ij}(X_n(t) \neq Y_n(t)) \leq \left(\frac{1}{2} + \frac{2}{n}\right)^t. \tag{D.33}$$

Next, suppose $X_n(0) = i, Y_n(0) = j$ for $i \neq j$ in different cliques. Fix $t \in \mathbb{N}, \tau \in \{1, \dots, t\}$, and let E_τ denote the event that $X_n(\tau), Y_n(\tau)$ are in the same clique. Then

$$\begin{aligned}\mathbb{P}_{ij}(X_n(t) \neq Y_n(t)) &= \mathbb{P}(X_n(t) \neq Y_n(t) | E_\tau)\mathbb{P}(E_\tau | X_n(0) = i, Y_n(0) = j) \\ &\quad + \mathbb{P}(X_n(t) \neq Y_n(t) | E_\tau^C)\mathbb{P}(E_\tau^C | X_n(0) = i, Y_n(0) = j) \\ &\leq \mathbb{P}(X_n(t) \neq Y_n(t) | E_\tau) + \mathbb{P}(E_\tau^C | X_n(0) = i, Y_n(0) = j), \end{aligned} \tag{D.34}$$

where we used the Markov property. For the first summand in (D.34), we use time invariance and the fact that (D.33) holds for any $i \neq j$ in the same clique to obtain

$$\mathbb{P}(X_n(t) \neq Y_n(t) | E_\tau) = \mathbb{P}(X_n(t-\tau) \neq Y_n(t-\tau) | E_0) \leq \left(\frac{1}{2} + \frac{2}{n}\right)^{t-\tau}$$

For the second summand in (D.34), note that E_τ^C implies $X_n(\tau'), Y_n(\tau')$ are not in the same clique $\forall\, \tau' \leq \tau$ (since once they reach the same clique, they remain in the same clique forever). This in turn implies that at each such τ', the chain that moves in (B) at step τ' moves within its current clique, which occurs with probability $1 - 2/n$. Thus, by independence,

$$\mathbb{P}(E_\tau^C | X_n(0) = i, Y_n(0) = j) \leq \left(1 - \frac{2}{n}\right)^\tau \leq \exp\left(-\frac{2\tau}{n}\right).$$

To summarize, we have shown that if $X_n(0) = i, Y_n(0) = j$ for $i \neq j$ not in the same clique,

$$\mathbb{P}_{ij}(X_n(t) \neq Y_n(t)) \leq \left(\frac{1}{2} + \frac{2}{n}\right)^{t-\tau} + \exp\left(-\frac{2\tau}{n}\right). \tag{D.35}$$

Combining (D.33) and (D.35), we thus obtain for any $t \in \mathbb{N}, \tau \in \{1, \dots, t\}$,

$$\max_{i,j \in [n]} \mathbb{P}_{ij}(X_n(t) \neq Y_n(t)) \leq \max\left\{\left(\frac{1}{2} + \frac{2}{n}\right)^t, \left(\frac{1}{2} + \frac{2}{n}\right)^{t-\tau} + \exp\left(-\frac{2\tau}{n}\right)\right\}. \tag{D.36}$$

can sample uniformly from the clique, excluding the states $X_n(t), Y_n(t)$, when the die is 3.

Now it is straightforward to verify that if (for example)

$$n \geq 6, \quad \tau \geq \frac{n}{2}\log\left(\frac{2}{\varepsilon}\right), \quad t \geq \tau + \frac{\log(2/\varepsilon)}{\log(6/5)} \geq \frac{n}{2}\log\left(\frac{2}{\varepsilon}\right) + \frac{\log(2/\varepsilon)}{\log(6/5)},$$

then (D.36) is bounded by ε. Hence, by (D.32), we obtain for some a_ε independent of n,

$$t_{\text{mix}}^{(n)}(\varepsilon) \leq \frac{n}{2}\log\left(\frac{2}{\varepsilon}\right) + \frac{\log(2/\varepsilon)}{\log(6/5)} \leq a_\varepsilon n \ \forall\ n \in \{6, 8, \ldots\}. \tag{D.37}$$

We next consider n odd. Here we could use a similar approach, but the auxiliary state n complicates this. Hence, we instead leverage this auxiliary state as follows: we wait until both chains leave state n (if necessary); we then ensure that the next visits to n occur simultaneously (after which point the chains run together indefinitely). More specifically, given $X_n(t), Y_n(t)$, we assign $X_n(t+1), Y_n(t+1)$ as follows:

(D) If $X_n(t) \neq Y_n(t)$, proceed to (E); else, let $X_n(t+1) \sim e_{X_n(t)}P_n, Y_n(t+1) = X_n(t+1)$.

(E) If $X_n(t) \neq n$ and $Y_n(t) \neq n$, proceed to (F). Otherwise, flip an independent fair coin. If heads, sample $X_n(t+1)$ from $N(X_n(t))$ uniformly and set $Y_n(t+1) = Y_n(t)$. If tails, set $X_n(t+1) = X_n(t)$ and sample $Y_n(t+1)$ from $N(Y_n(t))$ uniformly.

(F) Roll a die that lands 1, 2, and 3 with probability $\frac{1}{2}$, $\frac{1}{2} - \frac{1}{n+1}$, and $\frac{1}{n+1}$, respectively.
- If 1, set $X_n(t+1) = X_n(t), Y_n(t+1) = Y_n(t)$.
- If 2, independently and uniformly sample $X_n(t+1)$ and $Y_n(t+1)$ from $N(X_n(t)) \setminus \{n\}$ and $N(Y_n(t)) \setminus \{n\}$, respectively.
- If 3, set $X_n(t+1) = Y_n(t+1) = n$.

To analyze this coupling, first suppose $X_n(0) = i, Y_n(0) = j$ for some $i, j \in [n] \setminus \{n\}$ s.t. $i \neq j$. Then $X_n(t) \neq Y_n(t)$ implies the following, for each $\tau \leq t$:
- $X_n(\tau) \neq Y_n(\tau)$. (This can be proven by contradiction. Namely, if $X_n(\tau) = Y_n(\tau)$, then $X_n(t) \neq Y_n(t)$ is violated, since the chains run together forever after meeting by (D).)
- $X_n(\tau) \neq n, Y_n(\tau) \neq n$. (This can be proven inductively. For $\tau = 0$, it holds by assumption. For $\tau > 0$, we have $X_n(\tau - 1) \neq Y_n(\tau - 1)$ by the previous item and $X_n(\tau - 1) \neq n, Y_n(\tau - 1) \neq n$ by the inductive hypothesis. Hence, $X_n(\tau), Y_n(\tau)$ are assigned via (F). This implies $X_n(\tau) \neq n, Y_n(\tau) \neq n$; else, $X_n(\tau) = Y_n(\tau) = n$ by (F).)

By the argument of the second item, we can also conclude that, if $X_n(t) \neq Y_n(t)$, then $X_n(\tau), Y_n(\tau)$ were assigned via (F) for each $\tau \leq t$. Thus, at all such τ, the die in (F) must have landed 1 or 2 (else, $X_n(t) \neq Y_n(t)$ is violated); this occurs with probability $1 - \frac{1}{n+1}$. Hence, by independence,

$$\mathbb{P}_{ij}(X_n(t) \neq Y_n(t)) \leq \left(1 - \frac{1}{n+1}\right)^t \leq \exp\left(-\frac{t}{n+1}\right) \ \forall\ i, j \in [n] \setminus \{n\} \ s.t.\ i \neq j. \tag{D.38}$$

We next consider the case $X_n(0) = n$ or $Y_n(0) = n$; without loss of generality, assume $X_n(0) = n, Y_n(0) = j \neq n$. Let $\tau \leq t$ and define $E_\tau = \{X_n(\tau) \neq n, Y_n(\tau) \neq n\}$. Then

$$\begin{aligned}\mathbb{P}_{nj}(X_n(t) \neq Y_n(t)) = \mathbb{P}(X_n(t) \neq Y_n(t), E_\tau | X_n(0) = n, Y_n(0) = j) \\ + \mathbb{P}(X_n(t) \neq Y_n(t), E_\tau^C | X_n(0) = n, Y_n(0) = j)\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{P}(X_n(t) \neq Y_n(t)|E_\tau)\mathbb{P}(E_\tau|X_n(0)=n, Y_n(0)=j) \\
&\quad + \mathbb{P}(X_n(t) \neq Y_n(t), E_\tau^C|X_n(0)=n, Y_n(0)=j) \\
&\leq \mathbb{P}(X_n(t) \neq Y_n(t)|E_\tau) \\
&\quad + \mathbb{P}(X_n(t) \neq Y_n(t), E_\tau^C|X_n(0)=n, Y_n(0)=j), \qquad \text{(D.39)}
\end{aligned}
$$

where the equalities use the Markov property, and the inequality is immediate. Now for the first summand in (D.39), we can use time invariance and (D.38) to obtain

$$
\mathbb{P}(X_n(t) \neq Y_n(t)|E_\tau) = \mathbb{P}(X_n(t-\tau) \neq Y_n(t-\tau)|E_0) \leq \exp\left(-\frac{t-\tau}{n+1}\right). \qquad \text{(D.40)}
$$

For the second summand in (D.39), we again use $X_n(t) \neq Y_n(t) \Rightarrow X_n(\tau) \neq Y_n(\tau)$ to obtain

$$
\mathbb{P}(X_n(t) \neq Y_n(t), E_\tau^C|X_n(0)=n, Y_n(0)=j) \leq \mathbb{P}(X_n(\tau) \neq Y_n(\tau), E_\tau^C|X_n(0)=n, Y_n(0)=j) \qquad \text{(D.41)}
$$

We next claim (and will return to prove) that

$$
\{X_n(\tau) \neq Y_n(\tau), E_\tau^C\}|\{X_n(0)=n, Y_n(0)=j\} \Rightarrow X_n(\tau') = n \ \forall\ \tau' \leq \tau, \qquad \text{(D.42)}
$$

i.e. conditioned on the event $\{X_n(0)=n, Y_n(0)=j\}$, the event $\{X_n(\tau) \neq Y_n(\tau), E_\tau^C\}$ can only occur if the X_n-chain is lazy at every step up to τ. In other words, we require the τ independent coin tosses at the first τ iterations of (E) to all land tails. Hence, we conclude

$$
\mathbb{P}(X_n(\tau) \neq Y_n(\tau), E_\tau^C|X_n(0)=n, Y_n(0)=j) \leq 2^{-\tau}. \qquad \text{(D.43)}
$$

Combining (D.38), (D.39), (D.40), (D.41), and (D.43), we have shown that for n odd,

$$
\max_{i,j\in[n]} \mathbb{P}_{ij}(X_n(t) \neq Y_n(t)) \leq \max\left\{\exp\left(-\frac{t}{n+1}\right), 2^{-\tau} + \exp\left(-\frac{t-\tau}{n+1}\right)\right\}. \qquad \text{(D.44)}
$$

Therefore, if we choose (for example)

$$
\tau \geq \log_2\left(\frac{2}{\varepsilon}\right), \quad t \geq \tau + (n+1)\log\left(\frac{2}{\varepsilon}\right) \geq (n+1)\log\left(\frac{2}{\varepsilon}\right) + \log_2\left(\frac{2}{\varepsilon}\right),
$$

we conclude (D.44) is further bounded by ε. We thus obtain for some b_ε independent of n,

$$
t_{\text{mix}}^{(n)}(\varepsilon) \leq (n+1)\log\left(\frac{2}{\varepsilon}\right) + \log_2\left(\frac{2}{\varepsilon}\right) \leq b_\varepsilon n \ \forall\ n \in \{1,3,\ldots\}. \qquad \text{(D.45)}
$$

Finally, we can combine (D.37) and (D.45) to obtain for some $a_\varepsilon, b_\varepsilon$ independent of n,

$$
t_{\text{mix}}^{(n)}(\varepsilon) \leq \max\{a_\varepsilon, b_\varepsilon\} n \ \forall\ n \geq 6 \quad \Rightarrow \quad t_{\text{mix}}^{(n)}(\varepsilon) = O(n).
$$

We have completed the proof of $t_{\text{mix}}^{(n)}(\varepsilon) = O(n)$, assuming (D.42) holds. We now return to prove (D.42). Assume (for the sake of contradiction) that $X_n(\tau^*) = n, X_n(\tau^*+1) \neq n$ for

some $\tau^* < \tau$. (i.e. the X_n-chain was non-lazy at some $\tau^* < \tau$). Then, by (E), the Y_n-chain was lazy at time τ^*, i.e. $Y_n(\tau^*) = Y_n(\tau^* + 1)$. Now consider two cases:

1. $\tau^* = \tau - 1$: By assumption, $X_n(\tau) = X_n(\tau^* + 1) \neq n$. Also, we must have $Y_n(\tau) \neq n$: if instead $Y_n(\tau) = n$, then $n = Y_n(\tau) = Y_n(\tau^* + 1) = Y_n(\tau^*)$ (since $\tau^* = \tau - 1$ and the Y_n-chain was lazy at τ^*), which implies $X_n(\tau^*) = Y_n(\tau^*) = n$, which contradicts $X_n(\tau) \neq Y_n(\tau)$ in (D.42). Hence, $X_n(\tau) \neq n, Y_n(\tau) \neq n$, contradicting E_τ^C in (D.42).
2. $\tau^* < \tau - 1$: Similarly,$X_n(\tau^*+1) \neq n, Y_n(\tau^*+1) \neq n$ and $X_n(\tau^*+1) \neq Y_n(\tau^*+1)$. This implies $X_n(\tau^*+2), Y_n(\tau^*+2)$ were assigned via (F). In (F), the chains only move to n if they move to n together, after which point they remain together forever. Thus, neither chain can move to n at time $\tau^* + 2$, else $X_n(\tau) \neq Y_n(\tau)$ in (D.42) is contradicted. Repeating this argument for $\tau^* + 3, \ldots, \tau$ then contradicts E_τ^C in (D.42).

Since both cases yield contradictions, (D.42) is proven.

D.7 Proof of Proposition 5.2

D.7.1 Winning streak reversal

For the WSR, let $\{\alpha_n\}_{n\in\mathbb{N}}$, $\{\sigma_n\}_{n\in\mathbb{N}}$, c_1, c_2, and c_3 be as in the statement of the proposition. For $n \in \mathbb{N}$, set $m_n = \lfloor n^{c_1(1+c_2)/2} \rfloor$. Then by $\alpha_n = \Theta(n^{-c_1})$, $c_1 > 0$, and $c_2 > 1$,

$$\alpha_n m_n = \Theta\left(n^{-c_1} n^{c_1(1+c_2)/2}\right) = \Theta\left(n^{c_1(c_2-1)/2}\right) \quad \Rightarrow \quad \lim_{n\to\infty} \alpha_n m_n = \infty. \tag{D.46}$$

Again using $\alpha_n = \Theta(n^{-c_1})$, $c_1 > 0$, and $c_2 > 1$, we also observe

$$\left\lfloor c_3 \alpha_n^{-c_2} \right\rfloor - m_n = \Theta\left(n^{c_1 c_2} - n^{c_1(1+c_2)/2}\right) \quad \Rightarrow \quad \lim_{n\to\infty} \left(\left\lfloor c_3 \alpha_n^{-c_2} \right\rfloor - m_n\right) = \infty.$$

Consequently, we can find a sequence of positive integers $\{m_n'\}_{n\in\mathbb{N}}$ such that

$$\left\lfloor c_3 \alpha_n^{-c_2} \right\rfloor - m_n + 2 > m_n' \ \forall\ n \in \mathbb{N} \text{ sufficiently large}, \quad \lim_{n\to\infty} m_n' = \infty. \tag{D.47}$$

Now letting $e_{[m_n']} = \sum_{i\in[m_n']} e_i = \sum_{i=1}^{m_n'} e_i$, we can use Lemma 1.1 to obtain

$$\begin{aligned}
\pi_{\alpha_n,\sigma_n}([m_n']) &= \alpha_n \sum_{t=0}^{\infty} (1-\alpha_n)^t \sigma_n P_n^t e_{[m_n']}^\mathsf{T} = \alpha_n \sum_{t=0}^{\infty} (1-\alpha_n)^t \sum_{i=1}^{n} \sigma_n(i) e_i P_n^t e_{[m_n']}^\mathsf{T} \\
&= \alpha_n \sum_{t=0}^{m_n-1} (1-\alpha_n)^t \sum_{i=1}^{\lfloor c_3 \alpha_n^{-c_2} \rfloor} \sigma_n(i) e_i P_n^t e_{[m_n']}^\mathsf{T} && \text{(D.48)} \\
&\quad + \alpha_n \sum_{t=m_n}^{\infty} (1-\alpha_n)^t \sum_{i=1}^{n} \sigma_n(i) e_i P_n^t e_{[m_n']}^\mathsf{T} && \text{(D.49)} \\
&\quad + \alpha_n \sum_{t=0}^{m_n-1} (1-\alpha_n)^t \sum_{i=\lfloor c_3 \alpha_n^{-c_2} \rfloor + 1}^{n} \sigma_n(i) e_i P_n^t e_{[m_n']}^\mathsf{T}. && \text{(D.50)}
\end{aligned}$$

To bound the summands in (D.48)-(D.49), we use $e_i P_n^t e_{[m_n']}^\mathsf{T} \leq 1 \ \forall \ i,t$ to obtain

$$\alpha_n \sum_{t=0}^{m_n-1} (1-\alpha_n)^t \sum_{i=1}^{\lfloor c_3 \alpha_n^{-c_2} \rfloor} \sigma_n(i) e_i P_n^t e_{[m_n']}^\mathsf{T} \leq \alpha_n \sum_{t=0}^{m_n-1} (1-\alpha_n)^t \sum_{i=1}^{\lfloor c_3 \alpha_n^{-c_2} \rfloor} \sigma_n(i) \leq \sum_{i=1}^{\lfloor c_3 \alpha_n^{-c_2} \rfloor} \sigma_n(i),$$

$$\alpha_n \sum_{t=m_n}^{\infty} (1-\alpha_n)^t \sum_{i=1}^{n} \sigma_n(i) e_i P_n^t e_{[m_n']}^\mathsf{T} \leq \alpha_n \sum_{t=m_n}^{\infty} (1-\alpha_n)^t = (1-\alpha_n)^{m_n} \leq \exp(-\alpha_n m_n).$$

We next consider (D.50). First note that, whenever $i - t > m_n' > 0$, we have by (5.19),

$$e_i P_n^t e_{[m_n']} = e_{i-t} e_{[m_n']} = 0.$$

Also, every i,t pair in the summation in (D.50) satisfies, for n sufficiently large by (D.47),

$$i - t \geq \lfloor c_3 \alpha_n^{-c_2} \rfloor + 1 - (m_n - 1) = \lfloor c_3 \alpha_n^{-c_2} \rfloor - m_n + 2 > m_n',$$

which implies (D.50) is zero for all n large. We have therefore shown

$$\limsup_{n\to\infty} \pi_{\alpha_n,\sigma_n}([m_n']) \leq \limsup_{n\to\infty} \left(\sum_{i=1}^{\lfloor c_3 \alpha_n^{-c_2} \rfloor} \sigma_n(i) + \exp(-\alpha_n m_n) \right) = 0,$$

where the equality holds by assumption and (D.46). Since also $\pi_{\alpha_n,\sigma_n}([m_n']) \geq 0 \ \forall \ n \in \mathbb{N}$, we conclude $\lim_{n\to\infty} \pi_{\alpha_n,\sigma_n}([m_n']) = 0$. On the other hand,

$$\pi_n([m_n']) = \sum_{i=1}^{m_n'} \pi_n(i) = \sum_{i=1}^{m_n'} 2^{-i} = 1 - 2^{-m_n'} \xrightarrow[n\to\infty]{} 1,$$

where the limit holds since $m_n' \to \infty$ by (D.47). Combining arguments, we have shown

$$\liminf_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| \geq \liminf_{n\to\infty} (\pi_n([m_n']) - \pi_{\alpha_n,\sigma_n}([m_n'])) = 1,$$

and so, since $\|\pi_n - \pi_{\alpha_n,\sigma_n}\| \leq 1 \ \forall \ n \in \mathbb{N}$, we conclude $\lim_{n\to\infty} \|\pi_n - \pi_{\alpha_n,\sigma_n}\| = 1$.

D.7.2 Complete graph bijection

For the CGB, let $\{\alpha_n\}_{n\in\mathbb{N}}, \{\tilde{P}_n\}_{n\in\mathbb{N}}$ be given. Then

$$\|\pi_n - \tilde{\pi}_n\| \leq \max_{i\in[n]} \|\pi_n - e_i P_n\| + \max_{i\in[n]} \|e_i P_n - e_i \tilde{P}_n\| \leq d_n(1) + \alpha_n \ \forall \ n \in \mathbb{N},$$

where we have used Lemma D.1, global balance, and the fact that $\tilde{P}_n \in B(P_n, \alpha_n)$. Thus, using (D.27) from Appendix D.6 and the assumption $\limsup \alpha_n = \bar{\alpha}$, we obtain

$$\limsup_{n\to\infty} \|\pi_n - \tilde{\pi}_n\| \leq \limsup_{n\to\infty} d_n(1) + \limsup_{n\to\infty} \alpha_n = \frac{1}{2} + \bar{\alpha}.$$

APPENDIX E

Proofs and Experimental Details for Chapter VI

E.1 Proof of Theorems 6.1 and 6.2 (outline)

The proofs of Theorems 6.1 and 6.2 proceed in two steps. First, we show that the graph construction can be locally approximated by a certain branching process. Second, we analyze the beliefs of agents in the graph by instead analyzing the beliefs of agents in the tree resulting from the branching process. We note that studying tree agent beliefs rather than graph agent beliefs is advantageous because the tree has a comparatively simple structure.

The first step is identical for both theorems, while the second step requires a different analysis for each theorem. In Appendix E.1.1, we outline the first step, and in Appendices E.1.2 and E.1.3, respectively, we outline the second step for Theorems 6.1 and 6.2, respectively. To highlight the key ideas of our analysis, we defer many details to Appendix E.2; in particular, proofs pertaining to Appendices E.1.1, E.1.2, and E.1.3 , respectively, can be found in Appendices E.2.1, E.2.2, and E.2.3, respectively. Finally, we note that throughout the analysis we use $\mathbb{P}_n$ and $\mathbb{E}_n$, respectively, to denote probability and expectation, respectively, conditioned on the degree sequence $\{d_{out}(i), d_{in}^A(i), d_{in}^B(i)\}_{i\in[n]}$.

E.1.1 Branching process approximation

We first show the belief of any agent in the graph depends (asymptotically) only on the structure of the agent's neighborhood and on certain signals realized within this neighborhood. This will facilitate the definition of the branching process with which we will approximate the graph construction. Importantly, the agent's belief will *not* depend on the priors α_0, β_0 (asymptotically). This is necessary as we have not specified these priors (beyond assuming they are bounded by some $\bar{\alpha}, \bar{\beta}$ independent of n, as discussed in Section 6.2.1).

To begin, we require some notation. Let P denote the graph's column-normalized adjacency matrix, i.e. $P(i,j) = |\{i' \to j' \in E : i' = i, j' = j\}|/d_{in}(j)$, and set $Q = (1-\eta)I + \eta P$, where I is the identity matrix of appropriate dimension. (Recall from Section 6.2.2 that E is in general a multi-set; hence, the numerator in $P(i,j)$ may exceed 1.) Next, for $t \in \mathbb{N}$, let s_t denote the collection of signals $\{s_t(i)\}_{i\in A\cup B}$ in vector form. Finally, for $i \in A$ define

$$\vartheta_{T_n}(i) = \frac{1}{T_n}\sum_{t=0}^{T_n-1} s_{T_n-t} Q^t e_i^{\mathsf{T}}. \tag{E.1}$$

We note that (E.1) can be rewritten as

$$\vartheta_{T_n}(i) = \frac{1}{T_n} \sum_{t=0}^{T_n-1} \sum_{j \in A} s_{T_n-t}(j) e_j Q^t e_i^\mathsf{T}, \tag{E.2}$$

where we have used the fact that $s_t(j) = 0 \; \forall \; t \in \mathbb{N}, j \in B$. From this expression, it is clear that $\vartheta_{T_n}(i)$ only depends on the structure of the T_n-step neighborhood into i (since only this sub-graph affects the $e_j Q^t e_i^\mathsf{T}$ terms) and on certain signals within this neighborhood, as mentioned above. We can then establish the following.

Lemma E.1. Given (A4), $\forall \; \varepsilon > 0 \; \exists \; N$ s.t. $\forall \; n \geq N$, $|\theta_{T_n}(i) - \vartheta_{T_n}(i)| < \varepsilon$ *a.s.* $\forall \; i \in A$.

Proof. See Appendix E.2.1.1. □

Before defining the aforementioned branching process, we formally define the graph construction described in Section 6.2.2. For this, we will use the following additional notation.

- We let $A_l, l \in \mathbb{N}_0$ denote the set of agents at distance l from the initial agent i^*, i.e. $i \in A_l$ means a path from i to i^* of length l exists, but no shorter path exists. Similarly, we let $B_l, l \in \mathbb{N}_0$ denote the set of bots at distance l from i^*.
- We let $\{(i,j) : j \in [d_{out}(i)]\}$ denote the set of outstubs belonging to $i \in A$; we let O_A denote the set of all such outstubs.
- For each $(i,j) \in O_A$, we define a label $g((i,j)) \in \{1,2,3\}$ as follows:

$$g((i,j)) = \begin{cases} 1, & i \text{ does not yet belong to graph} \\ 2, & i \text{ belongs to graph but } (i,j) \text{ has not been paired} \\ 3, & i \text{ belongs to graph and } (i,j) \text{ has been paired} \end{cases}. \tag{E.3}$$

  We will explain the utility of these labels shortly.

With this notation in place, we present the formal graph construction as Algorithm E.1. We offer some further comments to help explain the algorithm:

- The algorithm takes as input the degree sequence $\{d_{out}(i), d_{in}^A(i), d_{in}^B(i)\}_{i \in A}$, which is used in Line 1 to define O_A. Also in Line 1, we label all outstubs as 1 (since no agents have been added to the graph), and we initialize the set of bots to the empty set.
- In Line 2, we sample the agent i^* from which the graph construction begins. Since i^* then belongs to the graph, we change the labels of its outstubs to 2.
- The remainder of the algorithm proceeds in a breadth-first-search fashion, iterating over l and agents i at distance l from i^*. For each such agent, we do the following:
  - For each of the $d_{in}^A(i)$ instubs of i intended for pairing with agent outstubs, we sample an agent outstub uniformly (Line 7), resampling until an unpaired outstub (i.e. one with label 1 or 2) has been found (Line 9). Upon finding such an outstub, denoted (i', j'), we pair it with i's instub to form an edge from i' to i (Line 10). Note that $g((i', j')) = 1$ implies i' was added to the graph when edge $i' \to i$ was formed; hence, because $i \in A_l$, i' is at distance $l+1$ from i^* and must be added to A_{l+1} (Line 11). Finally, we update the labels of the outstubs of i' via (E.3) (Lines 11-12). (Line 8 will be used in the branching process approximation to come.)

Algorithm E.1: `Graph-Construction`

1 Set $O_A = \{(i,j) : i \in A, j \in [d_{out}(i)]\}$, $g((i,j)) = 1 \; \forall \; (i,j) \in O_A$, $B = \emptyset$
2 Sample i^* uniformly from A; set $g((i^*,j)) = 2 \; \forall \; j \in [d_{out}(i^*)]$; set $A_0 = \{i^*\}$
3 **for** $l = 0$ **to** ∞ **do**
4 Set $A_{l+1} = B_{l+1} = \emptyset$
5 **for** $i \in A_l$ **do**
6 **for** $j = 1$ **to** $d^A_{in}(i)$ **do**
7 Sample (i',j') from O_A uniformly
8 **if** $g((i',j')) \neq 1$ **and** $\tau_n = \infty$ **then** set $\tau_n = l$
9 **while** $g((i',j')) = 3$ **do** sample (i',j') from O_A uniformly
10 Add directed edge from i' to i
11 **if** $g((i',j')) = 1$ **then** set $A_{l+1} = A_{l+1} \cup \{i'\}$, $g((i',j')) = 3$, $g((i',j'')) = 2 \; \forall \; j'' \in [d_{out}(i')] \setminus \{j'\}$
12 **else if** $g((i',j')) = 2$ **then** set $g((i',j')) = 3$
13 **for** $j = 1$ **to** $d^B_{in}(i)$ **do**
14 Add bot $b = n + |B| + 1$ with self-loop and unpaired outstub, set $B = B \cup \{b\}, B_{l+1} = B_{l+1} \cup \{b\}$
15 Add directed edge from b to i
16 **if** $g((i',j')) = 3 \; \forall \; (i',j') \in O_A$ **then return**

- For each of the $d^B_{in}(i)$ instubs of i intended for pairing with bot outstubs, we add a new bot with a self-loop and an unpaired outstub to the set of bots, updating B_{l+1} accordingly (Line 14), and then add an edge from the new bot to i (Line 15). Note here that $B = \emptyset$ at the start of the construction; it follows that the k-th bot added to the graph is $n + k + 1$, so $B = n + [\sum_{i \in A} d^B_{in}(i)]$ is the set of bots at the end of the construction.
- Finally, if all outstubs have been paired, the construction terminates (Line 16).

We return to discuss Line 8 of Algorithm E.1. Here τ_n denotes the first iteration an outstub with label 2 or 3 is sampled for pairing with an instub. Put differently, $\tau_n > l$ means that for the first l iterations of the construction, only outstubs with label 1 have been sampled. This has two consequences. First, no edges have been added between two nodes both at distance $\leq l$ from i^*, i.e. the l-step incoming neighborhood of i^* is a tree (except for bot self-loops). Second, no resampling of outstubs has occurred (Line 9); this implies that the outstub (i',j') paired in Line 10 is chosen uniformly from O_A, so the degrees $(d_{out}(i'), d^A_{in}(i'), d^B_{in}(i'))$ of i' are distributed according to the out-degree distribution f_n defined in (6.7).

These observations motivate a tree construction that we define next. In particular, we will construct a tree (except for bot self-loops) with edges pointing towards the root. Agents will be added to the tree with degrees sampled from f_n, except for the root node, whose degrees are sampled from f^*_n (6.7), corresponding to the degrees of i^* in the graph.

The tree construction requires further notation. First, we let $\hat{A}_l$ ($\hat{B}_l$, respectively) denote agents (bots, respectively) at distance l from the tree's root. We also set $\hat{A} = \cup_{l=0}^{\infty} \hat{A}_l, \hat{B} = \cup_{l=0}^{\infty} \hat{B}_l$. (Here and moving forward, we use $\hat{\cdot}$ to distinguish tree-related objects from similarly-

Algorithm E.2: `Tree-Construction`

1 Define f_n, f_n^* via (6.7), set $\hat{A}_0 = \{\phi\}$, sample $(d_{out}(\phi), d_{in}^A(\phi), d_{in}^B(\phi))$ from f_n^*
2 Set $X_0^1 = X_0^2 = \phi$
3 **for** $l = 0$ **to** ∞ **do**
4 Set $\hat{A}_{l+1} = \hat{B}_{l+1} = \emptyset$
5 **for** $\imath \in \hat{A}_l$ **do**
6 **for** $k \in \{1, 2\}$ **do**
7 **if** $X_l^k = \imath$ **then**
8 Sample j^* from $[d_{in}^A(\imath) + d_{in}^B(\imath)]$ uniformly, set $X_{l+1}^k = (\imath, j^*)$
9 **if** $j^* > d_{in}^A(\imath)$ **then** set $X_{l'}^k = (\imath, j^*)$ $\forall\ l' \in \{l+2, l+3, \cdots\}$
10 **for** $j = 1$ **to** $d_{in}^A(\imath)$ **do**
11 Sample $(d_{out}((\imath, j)), d_{in}^A((\imath, j)), d_{in}^B((\imath, j)))$ from f_n
12 Add directed edge from $(\imath, j)$ to $\imath$, set $\hat{A}_{l+1} = \hat{A}_{l+1} \cup \{(\imath, j)\}$
13 **for** $j = 1$ **to** $d_{in}^B(\imath)$ **do**
14 Add bot $b = (\imath, d_{in}^A(\imath) + j)$ with self-loop and unpaired outstub, set $\hat{B}_{l+1} = \hat{B}_{l+1} \cup \{b\}$
15 Add directed edge from b to i

defined graph-related ones.) At times, we will use branching process terminology and e.g. refer to $\hat{A}_l$ as the l-th *generation* of agents. We let ϕ denote the root node, so that $\hat{A}_0 = \{\phi\}$. We will denote generic node in $\hat{A}_l \cup \hat{B}_l$ as $\imath \in \mathbb{N}^l$; here $\imath = (i_1, \ldots, i_l)$ encodes the ancestry of $\imath$, i.e. $(i_1, \ldots, i_l)$ is the child of $(i_1, \ldots, i_{l-1})$, the grandchild of $(i_1, \ldots, i_{l-2})$, etc. Finally, for such $\imath$ and for $j \in \mathbb{N}$, $(\imath, j) = (i_1, \ldots, i_l, j)$ is the concatenation operation and $\imath|j = (i_1, \ldots, i_j)$ denotes $\imath$'s ancestor in generation j, with $\imath|0 = \phi$ by convention (note $\imath|l = \imath$).

We define the tree construction in Algorithm E.2 and offer several comments:

- Lines 2 and 6-9 define a random walk used in Appendix E.1.2; they do not affect the tree structure and we defer further explanation to Appendix E.1.2.
- As mentioned above, the root node ϕ has degrees sampled from f_n^* (Line 1), while all other agents have degrees sampled from f_n (Line 11).
- In Line 12, a directed edge is added from $(\imath, j)$ to $\imath$; the other $d_{out}((\imath, j)) - 1$ outstubs of $(\imath, j)$ are left unpaired so that the tree structure is preserved (except for bot self-loops).
- At the conclusion of the l-th iteration, $\imath \in \hat{A}_l$ has incoming neighbor set (offspring, in the branching process terminology) $\{(\imath, j) : j \in [d_{in}^A(\imath) + d_{in}^B(\imath)]\}$. More specifically, the subset $(\imath, 1), \ldots, (\imath, d_{in}^A(\imath))$ of $\imath$'s incoming neighbors are agents (Line 12), while the subset $(\imath, d_{in}^A(\imath) + 1), \ldots, (\imath, d_{in}^A(\imath) + d_{in}^B(\imath))$ of $\imath$'s incoming neighbors are bots (Line 14).
- Unlike the graph construction, the tree construction continues indefinitely, yielding an infinite tree (except for bot self-loops) with edges pointing towards the root node ϕ.

Having defined the tree construction, we also define $\hat{\vartheta}_{T_n}(\phi)$ as in (E.1) but using the tree from Algorithm E.2 instead of the graph from Algorithm E.1. Specifically, we let

$$\hat{\vartheta}_{T_n}(\phi) = \frac{1}{T_n} \sum_{t=0}^{T_n - 1} \hat{s}_{T_n - t} \hat{Q}^t e_\phi^\mathsf{T} = \frac{1}{T_n} \sum_{t=0}^{T_n - 1} \sum_{\imath \in \hat{A}} \hat{s}_{T_n - t}(\imath) e_\imath \hat{Q}^t e_\phi^\mathsf{T}, \tag{E.4}$$

where $\hat{s}_t(\imath) \sim \text{Bernoulli}(\theta)\ \forall\ t \in \mathbb{N}, \imath \in \hat{A}$; $\hat{s}_t(\imath) = 0\ \forall\ t \in \mathbb{N}, \imath \in \hat{B}$; $\hat{Q} = (1-\eta)I + \eta\hat{P}$; and $\hat{P}$ is the column-normalized adjacency matrix of the tree from Algorithm E.2. We note

$$0 \leq \hat{\vartheta}_{T_n}(\phi) \leq \frac{1}{T_n}\sum_{t=0}^{T_n-1} \mathbf{1}\hat{Q}^t e_\phi^\intercal = 1, \tag{E.5}$$

where the first inequality holds since (E.4) is a sum of nonnegative terms, the second follows since $\sum_{\imath\in\hat{A}} \hat{s}_{T_n-t}(\imath)e_\imath \leq \mathbf{1}$ component-wise (where $\mathbf{1}$ is the all ones vectors) and since $\hat{Q}^t e_\phi^\intercal$ is element-wise nonnegative, and the equality holds by column stochasticity of $\hat{Q}$.

We can now state Lemma E.2, which relates the belief of a uniformly random agent in the graph with the belief of the root node in the tree. For the first statement in the lemma, we argue that, conditioned on $\tau_n > T_n$, the T_n-step neighborhood of i^* in the graph and the T_n-step neighborhood of ϕ in the tree are constructed via the same procedure; since the signals are defined in the same manner as well, this implies $\vartheta_{T_n}(i^*)$ and $\hat{\vartheta}_{T_n}(\phi)$ have the same distribution. The second statement of the lemma says that the condition $\tau_n > T_n$ occurs with high probability; it is essentially implied by [48, Lemma 5.4]. We note that the assumptions (A1) and (A2) are required for this second statement to hold, and are standard assumptions needed to locally approximate a sparse random graph construction with a branching process. Finally, we recall $\zeta < 1/2$ by (A2), which is why the limit shown in Lemma E.2 holds.

Lemma E.2. Assume (A1) and (A2) hold, and let $\overset{\mathscr{D}}{=}$ denote equality in distribution. Then

$$\vartheta_{T_n}(i^*)|\{\tau_n > T_n\} \overset{\mathscr{D}}{=} \hat{\vartheta}_{T_n}(\phi), \quad \mathbb{P}(\tau_n \leq T_n|\Omega_{n,1}) = O\left(n^{\zeta-1/2}\right) \xrightarrow[n\to\infty]{} 0.$$

Proof. See Appendix E.2.1.2. □

We can now state and prove Lemma E.3, which is the main result for Step 1 of the proofs of the theorems. This result will allow us to analyze convergence of $\theta_{T_n}(i^*)$ (the graph agent belief) by instead analyzing convergence of $\hat{\vartheta}_{T_n}(\phi)$ (the tree agent belief).

Lemma E.3. Assume (A1), (A2), and (A4) hold. Then $\forall\ x \in \mathbb{R}$ and all $n \in \mathbb{N}$ large,

$$\mathbb{P}(|\theta_{T_n}(i^*) - x| > \varepsilon) \leq \mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - x| > \varepsilon/2) + \mathbb{P}(\Omega_{n,1}^C) + O\left(n^{\zeta-1/2}\right).$$

Proof. First, given $\varepsilon > 0$, we have for sufficiently large n,

$$\mathbb{P}(|\theta_{T_n}(i^*) - x| > \varepsilon) \leq \mathbb{P}(|\theta_{T_n}(i^*) - \vartheta_{T_n}(i^*)| + |\vartheta_{T_n}(i^*) - x| > \varepsilon) \leq \mathbb{P}(|\vartheta_{T_n}(i^*) - x| > \varepsilon/2),$$

where the first inequality uses the triangle inequality and in the second we used Lemma E.1 to bound $|\theta_{T_n}(i^*) - \vartheta_{T_n}(i^*)|$ by $\varepsilon/2$ *a.s.* Furthermore, by the law of total probability, we have

$$\mathbb{P}(|\vartheta_{T_n}(i^*) - x| > \varepsilon/2) \leq \mathbb{P}(|\vartheta_{T_n}(i^*) - x| > \varepsilon/2|\tau_n > T_n) + \mathbb{P}(\tau_n \leq T_n|\Omega_{n,1}) + \mathbb{P}(\Omega_{n,1}^C).$$

Combining the previous two inequalities and using Lemma E.2, we obtain

$$\mathbb{P}(|\theta_{T_n}(i^*) - x| > \varepsilon) \leq \mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - x| > \varepsilon/2) + O\left(n^{\zeta-1/2}\right) + \mathbb{P}(\Omega_{n,1}^C),$$

which is what we set out to prove. □

Before proceeding, we state a lemma that will be used in Step 2 of the proofs for both theorems. This lemma uses the fact that each agent in the tree has a unique path to the root. As a result, we can obtain an alternate expression for the terms $e_{\imath}\hat{Q}^t e_{\phi}^{\mathsf{T}}$ in (E.4).

Lemma E.4. For each $n \in \mathbb{N}$,

$$\hat{\vartheta}_{T_n}(\phi) = \frac{1}{T_n}\sum_{t=0}^{T_n-1}\sum_{l=0}^{t}\binom{t}{l}\eta^l(1-\eta)^{t-l}\sum_{\imath\in\hat{A}_l}\hat{s}_{T_n-t}(\imath)\prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}\ a.s., \tag{E.6}$$

where by convention $\prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1} = 1$ when $l = 0$.

Proof. See Appendix E.2.1.3. □

E.1.2 Step 2 for proof of Theorem 6.1

We next establish convergence of $\hat{\vartheta}_{T_n}(\phi)$, from which convergence of $\theta_{T_n}(i^*)$ will follow via Lemma E.3. We will use Chebyshev's inequality, so we begin with two lemmas describing the limiting behavior of the mean and variance of $\hat{\vartheta}_{T_n}(\phi)$. Here and moving forward, for random variables X and Y we use $\mathrm{Var}_n(X) = \mathbb{E}_n[X^2] - (\mathbb{E}_n[X])^2$ and $\mathrm{Cov}_n(X,Y) = \mathbb{E}_n[XY] - \mathbb{E}_n[X]\mathbb{E}_n[Y]$ to denote variance and covariance conditional on the degree sequence.

Lemma E.5. Given (A3) and (A4), we have the following:

$$\lim_{n\to\infty} T_n(1-p_n) = 0 \Rightarrow \lim_{n\to\infty} |\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] - \theta| 1(\Omega_{n,2}) = 0\ a.s.$$
$$\lim_{n\to\infty} T_n(1-p_n) = c \in (0,\infty) \Rightarrow \lim_{n\to\infty} \left|\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] - \theta\frac{1-e^{-c\eta}}{c\eta}\right| 1(\Omega_{n,2}) = 0\ a.s.$$
$$\lim_{n\to\infty} T_n(1-p_n) = \infty \Rightarrow \lim_{n\to\infty} |\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)]| 1(\Omega_{n,2}) = 0\ a.s.$$

Proof. See Appendix E.2.2.1. □

Lemma E.6. Given (A3) and (A4), $\lim_{n\to\infty} \mathrm{Var}_n(\hat{\vartheta}_{T_n}(\phi)) 1(\Omega_{n,2}) = 0\ a.s.$

Proof. See Appendix E.2.2.2. □

Before proceeding, we briefly describe our approach to proving these lemmas. First, we note that in analyzing the moments of $\hat{\vartheta}_{T_n}(\phi)$, the i.i.d. Bernoulli random variables $\hat{s}_{T_n-t}(\imath)$ in (E.6) are easily dealt with; the difficulty arises from the $\prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}$ terms. Luckily, there is a simple interpretation of that guides our analysis and that proceeds as follows. First, define a random walk $\{X_l^1\}_{l\in\mathbb{N}_0}$ with $X_0^1 = \phi$ and X_l^1 chosen uniformly from the incoming neighbors of X_{l-1}^1, for each $l \in \mathbb{N}$. Then, as shown in (E.25) in Appendix E.2.2.1,

$$\mathbb{E}\sum_{\imath\in\hat{A}_l}\prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1} = \mathbb{P}(X_l^1 \in \hat{A}_l).$$

In short, computing the mean of $\hat{\vartheta}_{T_n}(\phi)$ amounts to computing hitting probabilities of the form $\mathbb{P}(X_l^1 \in \hat{A}_l)$. Similarly, to analyze the second moment of $\hat{\vartheta}_{T_n}(\phi)$, we compute hitting probabilities of the form $\mathbb{P}(X_l^1 \in \hat{A}_l, X_l^2 \in \hat{A}_l)$, where X_l^2 is defined in the same manner as X_l^1 and is conditionally independent of X_l^1 given the tree structure. We note that, in principal, the k-th moment of $\hat{\vartheta}_{T_n}(\phi)$ can be computed by analyzing k walks. However, the calculations become exceedingly complex as k grows, and because we only require two moments, we do not study any case $k > 2$.

This interpretation explains Lines 2 and 6-9 of Algorithm E.2: in Line 2, we begin two walks at the root node ϕ; each time Lines 6-9 are reached, we advance the walks one step. Importantly, we simultaneously sample the walks and construct the tree, i.e. the l-th step of the walk is taken at Line 8, *before* the degrees of the corresponding node are realized in Line 11; this is crucial to our computation of the aforementioned hitting probabilities. Finally, we note that in Line 9 of Algorithm E.2, the condition $j^* > d_{in}^A(\imath)$ implies the walk reaches the set of bots $\hat{B}$; since bots have self-loops but no other incoming edges, they act as absorbing states on the walk. This is why the future trajectory of the walk can be defined in Line 9.

In Lemmas E.7 and E.8, we compute the hitting probabilities needed for the proofs of Lemmas E.5 and E.6. We note that, in addition to the random variables $\tilde{p}_n, \tilde{p}_n^*, \tilde{q}_n$ defined in (6.8) in Section 6.3.1, Lemma E.8 requires the definition of several similar random variables; we define these in (E.7) (and recall the definitions of $\tilde{p}_n, \tilde{p}_n^*, \tilde{q}_n$). We discuss these in more detail shortly.

$$\begin{aligned}
\tilde{p}_n &= \sum_{j\in\mathbb{N},k\in\mathbb{N}_0} \frac{j}{j+k} \sum_{i\in\mathbb{N}} f_n(i,j,k) & \tilde{p}_n^* &= \sum_{j\in\mathbb{N},k\in\mathbb{N}_0} \frac{j}{j+k} \sum_{i\in\mathbb{N}} f_n^*(i,j,k) \\
\tilde{q}_n &= \sum_{j\in\mathbb{N},k\in\mathbb{N}_0} \frac{j}{j+k}\frac{1}{j+k} \sum_{i\in\mathbb{N}} f_n(i,j,k) & \tilde{q}_n^* &= \sum_{j\in\mathbb{N},k\in\mathbb{N}_0} \frac{j}{j+k}\frac{1}{j+k} \sum_{i\in\mathbb{N}} f_n^*(i,j,k) \\
\tilde{r}_n &= \sum_{j\in\mathbb{N},k\in\mathbb{N}_0} \frac{j}{j+k}\frac{j-1}{j+k} \sum_{i\in\mathbb{N}} f_n(i,j,k) & \tilde{r}_n^* &= \sum_{j\in\mathbb{N},k\in\mathbb{N}_0} \frac{j}{j+k}\frac{j-1}{j+k} \sum_{i\in\mathbb{N}} f_n^*(i,j,k)
\end{aligned} \tag{E.7}$$

Lemma E.7. We have

$$\mathbb{P}_n(X_l^1 \in \hat{A}) = \begin{cases} \tilde{p}_n^* \tilde{p}_n^{l-1}, & l \in \mathbb{N} \\ 1, & l = 0 \end{cases}.$$

Proof. See Appendix E.2.2.4. □

Lemma E.8. For $l' > l$, we have

$$\mathbb{P}_n(X_l^1 \in \hat{A}, X_{l'}^2 \in \hat{A}) = \begin{cases} \mathbb{P}_n(X_l^1 \in \hat{A}, X_l^2 \in \hat{A})\tilde{p}_n^{l'-l}, & l \in \mathbb{N} \\ \tilde{p}_n^* \tilde{p}_n^{l'-1}, & l = 0 \end{cases}.$$

Furthermore,

$$\mathbb{P}_n(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}) = \begin{cases} \tilde{r}_n^* \tilde{p}_n^{2(l-1)} + \sum_{t=2}^{l} \tilde{q}_n^* \tilde{q}_n^{t-2} \tilde{r}_n \tilde{p}_n^{2(l-t)} + \tilde{q}_n^* \tilde{q}_n^{l-1}, & l \in \{2,3,\ldots\} \\ \tilde{r}_n^* + \tilde{q}_n^*, & l = 1 \\ 1, & l = 0 \end{cases}. \tag{E.8}$$

Proof. See Appendix E.2.2.5. □

Before proceeding, we comment on the form of (E.8), which helps explain the definitions in (E.7). Namely, in (E.8), $\tilde{r}_n^* \tilde{p}_n^{2(l-1)}$ is the probability of the two random walks visiting different agents on the first step of the walk ($\tilde{r}_n^*$ term), then separately remaining in the agent set for the next $l-1$ steps of the walk ($\tilde{p}_n^{2(l-1)}$ term); similarly, $\tilde{q}_n^* \tilde{q}_n^{t-2} \tilde{r}_n \tilde{p}_n^{2(l-t)}$ is the probability of the walks visiting the same agents for $t-1$ steps ($\tilde{q}_n^* \tilde{q}_n^{t-2}$ term), then visiting a different agent on the t-th step ($\tilde{r}_n$ term), then separately remaining in the agent set for $l-t$ steps ($\tilde{p}_n^{2(l-t)}$ term); finally, $\tilde{q}_n^* \tilde{q}_n^{l-1}$ is the probability of the walks remaining together and in the agent set for l steps. Each of these arguments follows from (E.7): $\tilde{p}_n$ gives the probability of a single walk proceeding to an agent ($j/(j+k)$ term), $\tilde{q}_n$ gives the probability of two walks proceeding to the same agent ($j/(j+k)$ term for the first walk, $1/(j+k)$ term for the second walk), and $\tilde{r}_n$ gives the probability of two walks proceeding to different agents ($j/(j+k)$ term for the first walk, $(j-1)/(j+k)$ term for the second walk). Similar arguments apply to $\tilde{p}_n^*, \tilde{q}_n^*, \tilde{r}_n^*$, except these pertain to the first steps of the walks.

Equipped with Lemmas E.5 and E.6, we can prove Theorem 6.1. First, suppose $T_n(1-p_n) \to 0$. Given $\varepsilon > 0$, we can use Lemma E.3 to obtain (provided the limits exist)

$$\begin{aligned} \lim_{n\to\infty} \mathbb{P}(|\theta_{T_n}(i^*) - \theta| > \varepsilon) &\leq \lim_{n\to\infty} \left(\mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \theta| > \varepsilon/2) + \mathbb{P}(\Omega_{n,1}^C) + O\left(n^{\zeta - 1/2}\right) \right) \\ &= \lim_{n\to\infty} \mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \theta| > \varepsilon/2), \end{aligned} \tag{E.9}$$

where we have used $\mathbb{P}(\Omega_{n,1}^C) \to 0$ by (A1) and $\zeta < 1/2$ by (A2). Next, using total probability,

$$\mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \theta| > \varepsilon/2) \leq \mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \theta| > \varepsilon/2, \Omega_{n,2}) + \mathbb{P}(\Omega_{n,2}^C). \tag{E.10}$$

We can further expand the first summand in (E.10) as

$$\begin{aligned} &\mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \theta| > \varepsilon/2, \Omega_{n,2}) \leq \mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}_n \hat{\vartheta}_{T_n}(\phi)| + |\mathbb{E}_n \hat{\vartheta}_{T_n}(\phi) - \theta| > \varepsilon/2, \Omega_{n,2}) \\ &\leq \mathbb{P}\left(|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}_n \hat{\vartheta}_{T_n}(\phi)| > \frac{\varepsilon}{4}, \Omega_{n,2}\right) + \mathbb{P}\left(|\mathbb{E}_n \hat{\vartheta}_{T_n}(\phi) - \theta| > \frac{\varepsilon}{4}, \Omega_{n,2}\right), \end{aligned} \tag{E.11}$$

where we have simply used the triangle inequality and the union bound. Now for the first summand in (E.11), we have by Chebyshev's inequality,

$$\begin{aligned} \mathbb{P}\left(|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}_n \hat{\vartheta}_{T_n}(\phi)| > \frac{\varepsilon}{4}, \Omega_{n,2}\right) &= \mathbb{E}\left[\mathbb{P}_n\left(|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}_n \hat{\vartheta}_{T_n}(\phi)| > \frac{\varepsilon}{4}\right) 1(\Omega_{n,2})\right] \\ &\leq \frac{16}{\varepsilon^2} \mathbb{E}\left[\mathrm{Var}_n(\hat{\vartheta}_{T_n}(\phi)) 1(\Omega_{n,2})\right] \xrightarrow[n\to\infty]{} 0, \end{aligned} \tag{E.12}$$

where the limit holds by Lemma E.6. For second summand in (E.11), we write

$$\begin{aligned} \mathbb{P}\left(|\mathbb{E}_n \hat{\vartheta}_{T_n}(\phi) - \theta| > \frac{\varepsilon}{4}, \Omega_{n,2}\right) &= \mathbb{E}\left[1\left(|\mathbb{E}_n \hat{\vartheta}_{T_n}(\phi) - \theta| > \frac{\varepsilon}{4}\right) 1(\Omega_{n,2})\right] \\ &\leq \frac{4}{\varepsilon} \mathbb{E}[|\mathbb{E}_n \hat{\vartheta}_{T_n}(\phi) - \theta| 1(\Omega_{n,2})] \xrightarrow[n\to\infty]{} 0, \end{aligned} \tag{E.13}$$

where the first two lines use total expectation and the inequality $1(x > y) \leq x/y$ for $x, y > 0$ (which is easily verified), and the limit holds by Lemma E.5. Finally, combining (E.9), (E.10), (E.11), (E.12), and (E.13), and recalling that $\mathbb{P}(\Omega_{n,2}^C) \to 0$ by (A3), we obtain

$$0 \leq \lim_{n\to\infty} \mathbb{P}(|\theta_{T_n}(i^*) - \theta| > \varepsilon) \leq \lim_{n\to\infty} \mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \theta| > \varepsilon/2) = 0.$$

Since $\varepsilon > 0$ was arbitrary, we conclude that $\theta_{T_n}(i^*)$ converges to θ in probability, completing the proof in the case $T_n(1-p_n) \to 0$. For the cases $T_n(1-p_n) \to c \in (0,\infty)$ and $T_n(1-p_n) \to \infty$, respectively, we can replace θ with $\theta(1-e^{-c\eta})/(c\eta)$ and 0, respectively (the corresponding cases from Lemma E.5), but otherwise follow the same approach.

E.1.3 Step 2 for proof of Theorem 6.2

Similar to the second step in the proof of Theorem 6.1, we begin by analyzing the limiting behavior of $\hat{\vartheta}_{T_n}(\phi)$. However, we will use a different approach than that used in Theorem 6.1. This approach is made possible by the stronger assumptions of Theorem 6.2, and it will yield a fast rate of convergence that will allow us to prove the theorem.

To explain our approach, we first recall that Lemma E.4 states

$$\hat{\vartheta}_{T_n}(\phi) = \frac{1}{T_n} \sum_{t=0}^{T_n-1} \sum_{l=0}^{t} \binom{t}{l} \eta^l (1-\eta)^{t-l} \sum_{\imath \in \hat{A}_l} \hat{s}_{T_n-t}(\imath) \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}.$$

Hence, letting $\mathcal{T}$ denote the collection of random variables defining the tree structure,

$$\begin{aligned} \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] &= \frac{1}{T_n} \sum_{t=0}^{T_n-1} \sum_{l=0}^{t} \binom{t}{l} \eta^l (1-\eta)^{t-l} \sum_{\imath \in \hat{A}_l} \mathbb{E}[\hat{s}_{T_n-t}(\imath)|\mathcal{T}] \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1} \qquad \text{(E.14)} \\ &= \frac{\theta}{T_n} \sum_{t=0}^{T_n-1} \sum_{l=0}^{t} \binom{t}{l} \eta^l (1-\eta)^{t-l} \sum_{\imath \in \hat{A}_l} \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}, \end{aligned}$$

where we have simply used the fact that the signals are i.i.d. Bernoulli(θ) random variables. Our basic approach will now proceed in two steps. First, in Lemma E.9 we condition on the tree structure, so that $\hat{\vartheta}_{T_n}(\phi)$ is simply a weighted sum of i.i.d. Bernoulli(θ) random variables; the lemma shows that this weighted sum is close to its conditional mean $\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]$ with high probability. Second, in Lemma E.10, we show that the conditional mean $\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]$ converges to zero in probability. Before proceeding, we also note that an argument similar to (E.5) implies the following, which will be used throughout the section:

$$0 \leq \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] \leq \theta \ a.s. \qquad \text{(E.15)}$$

We now state Lemma E.9. As mentioned, the proof involves analyzing a weighted sum of i.i.d. random variables; hence, our analysis is similar to Hoeffding's.

Lemma E.9. Assume $\exists\ \mu > 0$ and $N' \in \mathbb{N}$ independent of n s.t. the following hold:

- (A4), with $T_n \geq \mu \log n\ \forall\ n \geq N'$.

Then $\forall\ \varepsilon > 0$,

$$\mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]| > \varepsilon) = O\left(n^{-2\varepsilon^2\mu}\right).$$

Proof. See Appendix E.2.3.1. □

Lemma E.10 states that conditional mean $\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]$ converges to zero in probability. Note that the only source of randomness in $\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]$ is the tree structure. Since the tree is generated recursively, $\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]$ has a martingale-like structure; this allows us to use an approach similar to the Azuma-Hoeffding inequality.

Lemma E.10. Assume $\exists\ \kappa, \mu > 0$ and $N' \in \mathbb{N}$ independent of n s.t. the following hold:

- (A3), with $P(\Omega_{n,2}) = O(n^{-\kappa})$ and $p < 1$.
- (A4), with $T_n \geq \mu \log n\ \forall\ n \geq N'$.

Then $\forall\ \varepsilon > 0$,

$$\mathbb{P}(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon) = O\left(n^{-\min\{\mu(\varepsilon\eta(1-p)/\theta)^2,\kappa\}}\right).$$

Proof. See Appendix E.2.3.2. □

We prove Theorem 6.2. First, since $\theta_{T_n}(i^*), \hat{\vartheta}_{T_n}(\phi) \geq 0$, taking $x = 0$ in Lemma E.3 gives

$$\begin{aligned}\mathbb{P}(\theta_{T_n}(i^*) > \varepsilon) &\leq \mathbb{P}(\hat{\vartheta}_{T_n}(\phi) > \varepsilon/2) + \mathbb{P}(\Omega_{n,1}^C) + O\left(n^{\zeta-1/2}\right) \\ &= \mathbb{P}(\hat{\vartheta}_{T_n}(\phi) > \varepsilon/2) + O\left(n^{-\kappa}\right) + O\left(n^{\zeta-1/2}\right), \end{aligned} \tag{E.16}$$

where the equality is by the theorem assumptions. For the first summand in (E.16), we write

$$\begin{aligned}\mathbb{P}(\hat{\vartheta}_{T_n}(\phi) > \varepsilon/2) &\leq \mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]| + \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon/2) \\ &\leq \mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]| > \varepsilon/4) + \mathbb{P}(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon/4) \\ &= O\left(n^{-\varepsilon^2\mu/8} + n^{-\min\{\mu(\varepsilon\eta(1-p)/\theta)^2/16,\kappa\}}\right) = O\left(n^{-\min\{\mu(\varepsilon\eta(1-p)/\theta)^2/16,\kappa\}}\right),\end{aligned}$$

where the first inequality is immediate, the second inequality uses the union bound, the second equality uses Lemmas E.9 and E.10, and the final equality holds since $\eta, p \in (0,1)$ implies $\varepsilon^2\mu/8 > \mu(\varepsilon\eta(1-p)/\theta)^2/16$. Substituting into (E.16),

$$\mathbb{P}(\theta_{T_n}(i^*) > \varepsilon) = O\left(n^{-\min\{(1/2)-\zeta,\mu(\varepsilon\eta(1-p)/\theta)^2/16,\kappa\}}\right). \tag{E.17}$$

We can then write

$$\mathbb{E}\left|\{i \in [n] : \theta_{T_n}(i) > \varepsilon\}\right| = n\mathbb{P}(\theta_{T_n}(i^*) > \varepsilon) = O\left(n^{1-\min\{(1/2)-\zeta,\mu(\varepsilon\eta(1-p)/\theta)^2/16,\kappa\}}\right),$$

where we have used (E.17). Hence, by Markov's inequality,

$$\begin{aligned}\mathbb{P}\left(\left|\{i \in [n] : \theta_{T_n}(i) > \varepsilon\}\right| > Kn^k\right) &\leq K^{-1}n^{-k}\mathbb{E}\left|\{i \in [n] : \theta_{T_n}(i) > \varepsilon\}\right| \\ &= O\left(n^{-k+(1-\min\{(1/2)-\zeta,\mu(\varepsilon\eta(1-p)/\theta)^2/16,\kappa\})}\right) \xrightarrow[n\to\infty]{} 0,\end{aligned}$$

where the limit holds by the assumption on k in the statement of the theorem.

E.1.4 Other remarks

E.1.4.1 A sufficient condition for extending Theorem 6.2

Here we show that the condition (6.12) from Section 6.3.3 is sufficient to extend Theorem 6.2 to other cases of p_n. Recall this condition is

$$\exists\ \gamma' > 0\ s.t.\ \mathbb{P}(|\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] - L(p_n)| > \varepsilon) = O\left(n^{-\gamma'}\right), \tag{E.18}$$

where $L(p_n)$ is the limit from Theorem 6.1 based on the relative asymptotics of T_n and p_n, i.e.

$$L(p_n) = \begin{cases} \theta, & T_n(1-p_n) \xrightarrow[n\to\infty]{} 0 \\ \theta(1-e^{-c\eta})/(c\eta), & T_n(1-p_n) \xrightarrow[n\to\infty]{} c \in (0,\infty) \\ 0, & T_n(1-p_n) \xrightarrow[n\to\infty]{} \infty \end{cases}. \tag{E.19}$$

Suppose (E.18) holds in the case $T_n(1-p_n) \to 0$, so that $L(p_n) = \theta$. In this case, we have

$$\begin{aligned} \mathbb{P}(|\theta_{T_n}(i^*) - \theta| > \varepsilon) &\leq \mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \theta| > \varepsilon/2) + O\left(n^{-\min\{\kappa,(1/2)-\zeta\}}\right) \\ &\leq \mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]| > \varepsilon/4) + \mathbb{P}(|\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] - \theta| > \varepsilon/4) + O\left(n^{-\min\{\kappa,(1/2)-\zeta\}}\right) \\ &\leq O\left(n^{-\varepsilon^2\mu/8}\right) + O\left(n^{-\gamma'}\right) + O\left(n^{-\min\{\kappa,(1/2)-\zeta\}}\right) = O\left(n^{-\min\{\varepsilon^2\mu/8,\gamma',\kappa,(1/2)-\zeta\}}\right), \end{aligned}$$

where the first inequality is Lemma E.3 (which holds for all cases of p_n) with $\mathbb{P}(\Omega_{n,1}) = O(n^{-\kappa})$ and the third uses Lemma E.9 (which holds for all cases of p_n) and the sufficient condition (E.18). Hence, by the argument following (E.17), we obtain for any $\varepsilon > 0$, $K > 0$, and $k' > 1 - \min\{\varepsilon^2\mu/8, \gamma', \kappa, (1/2) - \zeta\}$,

$$\mathbb{P}\left(|\{i \in [n] : |\theta_{T_n}(i) - \theta| > \varepsilon\}| > Kn^{k'}\right) \xrightarrow[n\to\infty]{} 0,$$

i.e. Theorem 6.2 holds with k replaced by k'. The same argument shows that Theorem 6.2 holds (with a change of k) if $T_n(1-p_n) \to c \in (0,\infty)$ or $T_n(1-p_n) \to \infty$ with $p_n \to 1$.

E.1.4.2 Comparing Step 2 for proofs of Theorems 6.1 and 6.2

As shown in Appendices E.1.2 and E.1.3, Step 2 for the proofs of both theorems involves bounding $\mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - L(p_n)| > \varepsilon/2)$ for the appropriate $L(p_n)$. One may wonder why we have conducted a different analysis for the two theorems. The reason is that, as shown in Appendix E.2.3.3, the analysis for Step 2 of Theorem 6.2 yields a bound that does not decay with n in the case $T_n(1-p_n) \to c \in [0,\infty)$. Hence, we have derived a bound for Theorem 6.1 that encompasses all cases of $\lim_{n\to\infty} T_n(1-p_n)$. On the other hand, the bound from Theorem 6.1 only states $\mathbb{P}(|\hat{\vartheta}_{T_n}(\phi) - L(p_n)| > \varepsilon/2) \to 0$ but does not provide a rate of convergence so cannot be used to prove Theorem 6.2. We also note Appendix E.2.3.3 shows that, while the bound for Step 2 of Theorem 6.2 *does* decay in n for the case $T_n(1-p_n) \to \infty$ with $p_n \to 1$, it does not decay quickly enough to establish (6.12).

E.2 Proof of Theorems 6.1 and 6.2 (details)

E.2.1 Branching process approximation

E.2.1.1 Proof of Lemma E.1

For $t \in \mathbb{N}_0$, let α_t, β_t denote the parameters $\{\alpha_t(i)\}_{i\in A\cup B}, \{\beta_t(i)\}_{i\in A\cup B}$ in vector form, and let $\mathbf{1}$ denote the all ones vector. We claim

$$\alpha_t = (1-\eta)\sum_{\tau=1}^{t} s_\tau Q^{t-\tau} + \alpha_0 Q^t, \quad \beta_t = (1-\eta)\sum_{\tau=1}^{t}(\mathbf{1}-s_\tau)Q^{t-\tau} + \beta_0 Q^t \quad \forall\, t \in \mathbb{N} \tag{E.20}$$

We prove (E.20) for α_t; the proof for β_t follows the same approach. First, we use the parameter update equations (6.6), and the definitions of P and Q from Appendix E.1.1 (P being the column-normalized adjacency matrix and $Q = (1-\eta)I + \eta P$) to write

$$\alpha_t = (1-\eta)(\alpha_t + s_t) + \eta\alpha_{t-1}P = (1-\eta)s_t + \alpha_{t-1}Q. \tag{E.21}$$

Now for $t = 1$, (E.20) is equivalent to (E.21). Assuming (E.20) holds for $t-1$, we have

$$\begin{aligned}\alpha_t &= (1-\eta)s_t + \alpha_{t-1}Q = (1-\eta)s_t + \left((1-\eta)\sum_{\tau=1}^{t-1} s_\tau Q^{(t-1)-\tau} + \alpha_0 Q^{t-1}\right)Q \\ &= (1-\eta)s_t + (1-\eta)\sum_{\tau=1}^{t-1} s_\tau Q^{t-\tau} + \alpha_0 Q^t = (1-\eta)\sum_{\tau=1}^{t} s_\tau Q^{t-\tau} + \alpha_0 Q^t.\end{aligned}$$

Next, recalling e_i is the vector with 1 in the i-th position and 0 elsewhere,

$$\begin{aligned}\theta_{T_n}(i) &= \frac{\alpha_{T_n}(i)}{\alpha_{T_n}(i)+\beta_{T_n}(i)} = \frac{(1-\eta)\sum_{\tau=1}^{T_n} s_\tau Q^{T_n-\tau}e_i^\mathsf{T} + \alpha_0 Q^{T_n}e_i^\mathsf{T}}{(1-\eta)\sum_{\tau=1}^{T_n}\mathbf{1}Q^{T_n-\tau}e_i^\mathsf{T} + (\alpha_0+\beta_0)Q^{T_n}e_i^\mathsf{T}} \\ &= \frac{(1-\eta)\sum_{\tau=1}^{T_n} s_\tau Q^{T_n-\tau}e_i^\mathsf{T} + \alpha_0 Q^{T_n}e_i^\mathsf{T}}{(1-\eta)T_n + (\alpha_0+\beta_0)Q^{T_n}e_i^\mathsf{T}} = \frac{\frac{1}{T_n}\sum_{\tau=1}^{T_n} s_\tau Q^{T_n-\tau}e_i^\mathsf{T} + \frac{1}{(1-\eta)T_n}\alpha_0 Q^{T_n}e_i^\mathsf{T}}{1 + \frac{1}{(1-\eta)T_n}(\alpha_0+\beta_0)Q^{T_n}e_i^\mathsf{T}},\end{aligned}$$

where the equalities hold by definition, by (E.20), since the columns of Q sum to 1 by definition, and by multiplying numerator and denominator by $\frac{1}{(1-\eta)T_n}$, respectively. Next, recall from Section 6.2.1 that $\alpha_0(j) \in [0, \bar{\alpha}]\ \forall\ j \in A\cup B$ for some $\bar{\alpha} > 0$. Hence, α_0 is element-wise upper bounded by $\bar{\alpha}\mathbf{1}$, so $\alpha_0 Q^{T_n}e_i^\mathsf{T} \le \bar{\alpha}\mathbf{1}Q^{T_n}e_i^\mathsf{T} = \bar{\alpha}$, where we have used column stochasticity of Q. Additionally, $\alpha_0 Q^{T_n}e_i^\mathsf{T} \ge 0$ (since the three terms in the product are elementwise nonnegative). By a similar argument, $0 \le \beta_0 Q^{T_n}e_i^\mathsf{T} \le \bar{\beta}$. Taken together, we can use the previous equation to obtain

$$\frac{\frac{1}{T_n}\sum_{\tau=1}^{T_n} s_\tau Q^{T_n-\tau}e_i^\mathsf{T}}{1+\frac{\bar{\alpha}+\bar{\beta}}{(1-\eta)T_n}} \le \theta_{T_n}(i) \le \frac{1}{T_n}\sum_{\tau=1}^{T_n} s_\tau Q^{T_n-\tau}e_i^\mathsf{T} + \frac{\bar{\alpha}}{(1-\eta)T_n}.$$

Finally, recall from Section 6.2.1 that $\bar{\alpha}$ and $\bar{\beta}$ are independent of n. Hence, because $T_n \to \infty$ as $n \to \infty$ (by (A4) in the statement of the lemma), $\bar{\alpha}/T_n, \bar{\beta}/T_n \to 0$ as $n \to \infty$. It follows

that, for given $\varepsilon > 0$ and n sufficiently large, $|\theta_{T_n}(i) - \frac{1}{T_n}\sum_{\tau=1}^{T_n} s_\tau Q^{T_n-\tau} e_i^\mathsf{T}| < \varepsilon$. Finally, by changing the index of summation, $\frac{1}{T_n}\sum_{\tau=1}^{T_n} s_\tau Q^{T_n-\tau} e_i^\mathsf{T} = \vartheta_{T_n}(i)$, completing the proof.

E.2.1.2 Proof of Lemma E.2

We begin by arguing $\vartheta_{T_n}(i^*)|\{\tau_n > T_n\} \overset{\mathscr{D}}{=} \hat{\vartheta}_{T_n}(\phi)$. For this, first consider the sub-graph containing only edges between two agents formed during the first T_n iterations of Algorithm E.1. Conditioned on $\tau_n > T_n$, this sub-graph is constructed as follows:

- The initial agent i^* is sampled uniformly from A (Line 2), which implies its degrees $(d_{out}(i^*), d_{in}^A(i^*), d_{in}^B(i^*))$ are distributed as f_n^*. (In fact, this holds even if $\tau_n \leq T_n$.)
- Each time an edge is added to the sub-graph (Line 10), the paired outstub (i', j') is sampled uniformly from O_A (else, $\tau_n > T_n$ is contradicted by Line 8-9), so the degrees $(d_{out}(i'), d_{in}^A(i'), d_{in}^B(i'))$ of the corresponding agent i' are distributed as f_n.
- The initial agent i^* has no paired outstubs, while all other agents in the sub-graph have one paired outstub (else, an outstub with label 2 was paired within the first T_n iterations, contradicting $\tau_n > T_n$ by Line 8); in particular, the sub-graph has $|\cup_{l=0}^{T_n} A_l|$ nodes and $|\cup_{l=0}^{T_n} A_l| - 1$ edges. Also, every agent in the sub-graph has a path to i^* by the breadth-first-search construction, so, neglecting edge polarities, we obtain a connected graph with $|\cup_{l=0}^{T_n} A_l|$ nodes and $|\cup_{l=0}^{T_n} A_l| - 1$ edges, i.e. a tree. Finally, since all edges point towards i^* (see Line 10), the sub-graph is a directed tree pointed towards i^*.

In summary, the sub-graph is a directed tree pointing towards an agent with degrees distributed as f_n^*, in which all other nodes have degrees distributed as f_n. This is precisely the procedure used to construct the sub-graph of agents during the first T_n iterations of Algorithm E.2. Additionally, Algorithms E.1 and E.2 add bots in the same manner (Lines 14-15 in Algorithm E.1, Lines 14-15 in Algorithm E.2). Taken together, we conclude that, conditioned on $\tau_n > T_n$, the T_n-step neighborhood into i^* is constructed in the same manner in Algorithm E.1 as the T_n-step neighborhood into ϕ is constructed in Algorithm E.2. Furthermore, by (E.2) and (E.4), it is clear that $\vartheta_{T_n}(i)$ and $\hat{\vartheta}_{T_n}(\phi)$, respectively, depend only on these respective neighborhoods, and on the signals $s_{T_n-t}(i)$ and $\hat{s}_{T_n-t}(\imath)$, respectively. Since the signals $s_{T_n-t}(i)$ and $\hat{s}_{T_n-t}(\imath)$ are also defined in the same manner $(s_{T_n-t}(i), \hat{s}_{T_n-t}(\imath) \sim \text{Bernoulli}(\theta)$ for $i \in A, \imath \in \hat{A}$; $s_{T_n-t}(i) = \hat{s}_{T_n-t}(\imath) = 0$ for $i \in B, \imath \in \hat{B})$, we ultimately conclude that $\vartheta_{T_n}(i^*)$ and $\hat{\vartheta}_{T_n}(\phi)$ have the same distribution when $\tau_n > T_n$ holds.

We next argue $\{\tau_n > T_n\}$ occurs with high probability when $\Omega_{n,1}$ holds. For this, we note that Algorithm E.1 is identical to the graph construction described in [48, Section 5.2] except the construction in [48] does not include the pairing of agent instubs with bots in Lines 14-15 of Algorithm E.1. However, these lines do not affect τ_n. Moreover, when (A1) holds, the assumptions of [48, Lemma 5.4] are satisfied. This lemma states that, if $t_n < (\log n)/(2\log(\nu_3/\nu_1))$ and $\nu_3 > \nu_1$ (with ν_1, ν_3 defined as in (A1)), then $P(\tau_n \leq t_n|\Omega_{n,1}) = O((\nu_3/\nu_1)^{t_n}/\sqrt{n})$. In particular, by (A2) we have $T_n \leq \zeta \log(n)/\log(\nu_3/\nu_1)$ for n sufficiently large, with $\zeta \in (0, 1/2)$ independent of n; substituting gives

$$\mathbb{P}(\tau_n \leq T_n|\Omega_{n,1}) = O\left(\frac{(\nu_3/\nu_1)^{\zeta \log(n)/\log(\nu_3/\nu_1)}}{\sqrt{n}}\right) = O\left(n^{\zeta - 1/2}\right).$$

E.2.1.3 Proof of Lemma E.4

We first claim that for $l \in \mathbb{N}_0$ and $\imath \in \hat{A}_l$,

$$e_\imath \hat{P}^{l'} e_\phi = \begin{cases} \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}, & l' = l \\ 0, & l' \in \mathbb{N}_0 \setminus \{l\} \end{cases}. \tag{E.22}$$

(Recall $\hat{P}$ is the column-normalized adjacency matrix.) We prove (E.22) separately for $l = 0$ and $l \in \mathbb{N}$. When $l = 0$, the only case is $\imath = \phi$ (since $\hat{A}_0 = \{\phi\}$); if $l' = 0$, the left side is clearly 1 and the right side is 1 by convention; if $l' \in \mathbb{N}$, the left side is 0 since $e_\phi \hat{P}^{l'} = 0$ (ϕ has no outgoing neighbors in the tree). Next, we aim to prove (E.22) for $\imath \in \hat{A}_l$ and $l \in \mathbb{N}$. For such $\imath$, there is a unique path from $\imath$ to ϕ with length l that visits the nodes $\imath|l = \imath, \imath|l-1, \ldots, \imath|0 = \phi$. By definition of $\hat{P}$, it follows that

$$e_\imath \hat{P}^l e_\phi = \hat{P}(\imath|l, \imath|l-1)\hat{P}(\imath|l-1, \imath|l-2)\cdots\hat{P}(\imath|1, \imath|0) = \frac{1}{d_{in}(\imath|l-1)}\frac{1}{d_{in}(\imath|l-2)}\cdots\frac{1}{d_{in}(\phi)}.$$

On the other hand, if $l' \neq l$, no path of length l' from $\imath$ to ϕ exists, so $e_\imath \hat{P}^{l'} e_\phi = 0$.

Recalling that $\hat{Q} = (1-\eta)I + \eta\hat{P}$, we next claim that $\forall\ t \in \mathbb{N}_0$,

$$\hat{Q}^t = \sum_{l=0}^{t} \binom{t}{l} \eta^l (1-\eta)^{t-l} \hat{P}^l. \tag{E.23}$$

We prove (E.23) inductively: both sides equal I when $t = 0$; assuming (E.23) is true for t,

$$\begin{aligned}
\hat{Q}^{t+1} &= ((1-\eta)I + \eta\hat{P}) \sum_{l=0}^{t} \binom{t}{l} \eta^l (1-\eta)^{t-l} \hat{P}^l \\
&= \sum_{l=0}^{t} \binom{t}{l} \eta^l (1-\eta)^{t+1-l} \hat{P}^l + \sum_{l=1}^{t+1} \binom{t}{l-1} \eta^l (1-\eta)^{t+1-l} \hat{P}^l \\
&= (1-\eta)^{t+1} I + \sum_{l=1}^{t} \left(\binom{t}{l} + \binom{t}{l-1} \right) \eta^l (1-\eta)^{t+1-l} \hat{P}^l + \eta^{t+1} \hat{P}^{t+1} \\
&= (1-\eta)^{t+1} I + \sum_{l=1}^{t} \binom{t+1}{l} \eta^l (1-\eta)^{t+1-l} \hat{P}^l + \eta^{t+1} \hat{P}^{t+1},
\end{aligned}$$

where in the first line we have used the definition of $\hat{Q}$ and the inductive hypothesis, the second line simply uses the distributive property, the third rearranges summations, and the fourth uses Pascal's rule ($[t+1]$ has $\binom{t+1}{l}$ subsets of cardinality l; $\binom{t}{l-1}$ that contain 1 and $\binom{t}{l}$ that do not contain 1). This completes the proof of (E.23).

Having established (E.23) and (E.22), we can combine them to obtain $\forall\ t \in \mathbb{N}_0, \imath \in \hat{A}_l$,

$$e_\imath \hat{Q}^t e_\phi = \begin{cases} \binom{t}{l} \eta^l (1-\eta)^{t-l} \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}, & l \leq t \\ 0, & l > t \end{cases}.$$

Finally, substituting the previous equation into (E.4), and recalling $\hat{A} = \cup_{l=0}^{\infty} \hat{A}_l$, we obtain

$$\hat{\vartheta}_{T_n}(\phi) = \frac{1}{T_n} \sum_{t=0}^{T_n-1} \sum_{l=0}^{t} \sum_{\imath \in \hat{A}_l} \hat{s}_{T_n-t}(\imath) \binom{t}{l} \eta^l (1-\eta)^{t-l} \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1},$$

which completes the proof.

E.2.2 Step 2 for proof of Theorem 6.1

E.2.2.1 Proof of Lemma E.5

First, letting $\mathcal{D}$ denote the degree sequence and $\mathcal{T}$ denote the set of random variables defining the tree structure, we can use Lemma E.4 to write

$$\begin{aligned}
\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] &= \frac{1}{T_n} \sum_{t=0}^{T_n-1} \sum_{l=0}^{t} \binom{t}{l} \eta^l (1-\eta)^{t-l} \mathbb{E}_n \left[\sum_{\imath \in \hat{A}_l} \mathbb{E}[\hat{s}_{T_n-t}(\imath)|\mathcal{D}, \mathcal{T}] \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1} \right] \\
&= \frac{\theta}{T_n} \sum_{t=0}^{T_n-1} \sum_{l=0}^{t} \binom{t}{l} \eta^l (1-\eta)^{t-l} \mathbb{P}_n(X_l^1 \in \hat{A}), \qquad \text{(E.24)}
\end{aligned}$$

where the first equality uses the fact that $\hat{A}_l$ and $d(\imath|j)^{-1}$ are fixed given the tree structure, and the second uses the fact that $\hat{s}_{T_n-t}(\imath) \sim \text{Bernoulli}(\theta)$ and the definition of X_l^1, i.e.

$$\mathbb{P}_n(X_l^1 \in \hat{A}) = \mathbb{E}_n[\mathbb{P}(X_l^1 \in \hat{A}|\mathcal{D}, \mathcal{T})] = \mathbb{E}_n \left[\sum_{\imath \in \hat{A}_l} \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1} \right]. \qquad \text{(E.25)}$$

Here we have also used the fact that $\{X_l^1\}_{l \in \mathbb{N}}$ is a walk starting at the root of a directed tree; hence, for $\imath \in \hat{A}_l$, $\mathbb{P}(X_l^1 = \imath|\mathcal{D}, \mathcal{T})$ is the probability of the path from ϕ to $\imath$, which is $\prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}$, and $X_l^1 \in \hat{A} \Leftrightarrow X_l^1 = \imath$ for some $\imath \in \hat{A}_l$. Next, using (E.24) and Lemma E.7,

$$\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] = \frac{\theta}{T_n} \sum_{t=0}^{T_n-1} \left(\sum_{l=1}^{t} \binom{t}{l} \eta^l (1-\eta)^{t-l} \tilde{p}_n^* \tilde{p}_n^{l-1} + (1-\eta)^t \right), \qquad \text{(E.26)}$$

where by convention the summation over l is zero when $t = 0$. Adding and subtracting $(1-\eta)^t \tilde{p}_n^* / \tilde{p}_n$, the previous equation can be rewritten as

$$\begin{aligned}
\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] &= \frac{\theta}{T_n} \frac{\tilde{p}_n^*}{\tilde{p}_n} \sum_{t=0}^{T_n-1} \sum_{l=0}^{t} \binom{t}{l} (\eta \tilde{p}_n)^l (1-\eta)^{t-l} + \frac{\theta}{T_n} \left(1 - \frac{\tilde{p}_n^*}{\tilde{p}_n} \right) \sum_{t=0}^{T_n-1} (1-\eta)^t \qquad \text{(E.27)} \\
&= \frac{\theta}{T_n} \frac{\tilde{p}_n^*}{\tilde{p}_n} \sum_{t=0}^{T_n-1} (1 - \eta(1-\tilde{p}_n))^t + \frac{\theta}{T_n} \left(1 - \frac{\tilde{p}_n^*}{\tilde{p}_n} \right) \frac{1 - (1-\eta)^{T_n}}{\eta} \\
&= \frac{\theta}{T_n} \frac{\tilde{p}_n^*}{\tilde{p}_n} \frac{1 - (1 - \eta(1-\tilde{p}_n))^{T_n}}{\eta(1-\tilde{p}_n)} + \frac{\theta}{T_n} \left(1 - \frac{\tilde{p}_n^*}{\tilde{p}_n} \right) \frac{1 - (1-\eta)^{T_n}}{\eta},
\end{aligned}$$

where we have simply used the binomial theorem and computed two geometric series.

Next, we assume temporarily that $p_n \to 1$ as $n \to \infty$. By (A3), we have for $\omega \in \Omega_{n,2}$

$$\tilde{p}_n(\omega) \in (p_n - \delta_n, p_n + \delta_n).$$

Hence, by $p_n \to 1$, and $\delta_n \to 0$ by (A3), we have for $\gamma_1 > 0$, n sufficiently large, and such ω

$$1 - \gamma_1 < \frac{\tilde{p}_n^*(\omega)}{\tilde{p}_n(\omega)} < 1 + \gamma_1,$$

where we have also used the fact that $1 \geq \tilde{p}_n^* \geq \tilde{p}_n$ on $\Omega_{n,2}$ by (A3). Also, by (A4), it is clear that $(1 - (1 - \eta))^{T_n}/T_n \to 0$, so for given $\gamma_2 > 0$ and n sufficiently large,

$$0 < \frac{\theta}{T_n} \frac{1 - (1 - \eta)^{T_n}}{\eta} < \gamma_2.$$

Combining the previous four equations implies that for n sufficiently large and $\omega \in \Omega_{n,2}$,

$$\begin{aligned} \mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)](\omega) &< (1 + \gamma_1)\frac{\theta}{T_n} \frac{1 - (1 - \eta(1 - p_n - \delta_n))^{T_n}}{\eta(1 - p_n - \delta_n)} + \gamma_1\gamma_2, \\ \mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)](\omega) &> (1 - \gamma_1)\frac{\theta}{T_n} \frac{1 - (1 - \eta(1 - p_n + \delta_n))^{T_n}}{\eta(1 - p_n + \delta_n)} - \gamma_1\gamma_2. \end{aligned} \tag{E.28}$$

We complete the proof for the case $T_n(1-p_n) \to 0$; the proof for other cases is similar. First, we use Lemma E.11 from Appendix E.2.4 to obtain for any $\gamma_3 > 0$ and for n large enough

$$\begin{aligned} 1 - \gamma_3 &< \frac{1 - (1 - \eta(1 - p_n - \delta_n))^{T_n}}{T_n\eta(1 - p_n - \delta_n)} < 1 + \gamma_3, \\ 1 - \gamma_3 &< \frac{1 - (1 - \eta(1 - p_n + \delta_n))^{T_n}}{T_n\eta(1 - p_n + \delta_n)} < 1 + \gamma_3. \end{aligned}$$

Combining the previous two equations gives for n large and $\omega \in \Omega_{n,2}$

$$\begin{aligned} \mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)](\omega) &< \theta(1 + \gamma_1)(1 + \gamma_3) + \gamma_1\gamma_2 = \theta + \theta(\gamma_1 + \gamma_3 + \gamma_1\gamma_3) + \gamma_1\gamma_2, \\ \mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)](\omega) &> \theta(1 - \gamma_1)(1 - \gamma_3) - \gamma_1\gamma_2 = \theta - \theta(\gamma_1 + \gamma_3 - \gamma_1\gamma_3) - \gamma_1\gamma_2. \end{aligned}$$

Hence, for given $\gamma > 0$, we can find $\gamma_1, \gamma_2, \gamma_3$ small and n large such that, for $\omega \in \Omega_{n,2}$, $|\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)](\omega) - \theta| < \gamma$. This clearly also implies $|\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)](\omega) - \theta|1(\Omega_{n,2})(\omega) < \gamma$ for such ω. On the other hand, for $\omega \notin \Omega_{n,2}$, it is trivial that $|\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)](\omega) - \theta|1(\Omega_{n,2})(\omega) = 0 < \gamma$. This completes the proof for the case $T_n(1 - p_n) \to 0$.

We return to the case $p_n \to p \in [0, 1)$. Here it follows from (A4) that $T_n(1-p_n) \to [0, \infty)$ cannot occur, i.e. we need only consider the case $T_n(1 - p_n) \to \infty$. First, note that since $p_n \to p < 1$ and $\delta_n \to 0$, we have $p_n + \delta_n < 1 - \gamma_1$ for some $\gamma_1 > 0$ and n sufficiently large.

For such n, and for $\omega \in \Omega_{n,2}$, we then obtain $\tilde{p}_n(\omega) < 1 - \gamma_1$; substituting into (E.26) gives

$$\begin{aligned}\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)](\omega) &< \frac{\theta}{T_n}\sum_{t=0}^{T_n-1}\left(\sum_{l=1}^{t}\binom{t}{l}\eta^l(1-\eta)^{t-l}(1-\gamma_1)^{l-1} + (1-\eta)^t\right) \\ &< \frac{\theta}{T_n}\frac{1}{1-\gamma_1}\sum_{t=0}^{T_n-1}\sum_{l=0}^{t}\binom{t}{l}\eta^l(1-\eta)^{t-l}(1-\gamma_1)^l \\ &= \frac{\theta}{T_n}\frac{1}{1-\gamma_1}\frac{1-(1-\eta\gamma_1)^{T_n}}{\eta\gamma_1} < \frac{\theta}{T_n}\frac{1}{1-\gamma_1}\frac{1}{\eta\gamma_1}\end{aligned} \tag{E.29}$$

where in the first inequality we used $\tilde{p}_n(\omega) < 1 - \gamma_1$ and $\tilde{p}_n^*(\omega) \leq 1$, in the second we used $1 - \gamma_1 \in (0, 1)$ (so that $(1-\eta)^t < (1-\eta)^t/(1-\gamma_1)$), for the equality we used the binomial theorem and computed a geometric series, and the final inequality is immediate. Since θ, η, γ_1 are independent of n, while $T_n \to \infty$ as $n \to \infty$ by (A4), it is clear from this final expression that, for given $\gamma > 0$, n sufficiently large, and $\omega \in \Omega_{n,2}$, $0 \leq \mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)](\omega) < \gamma$. It follows that $|\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)]|1(\Omega_{n,2}) \to 0$ *a.s.*, completing the proof.

E.2.2.2 Proof of Lemma E.6

First, suppose $p_n \to p \in [0, 1)$. Then, since $\hat{\vartheta}_{T_n}(\phi) \leq 1$ *a.s.* (see (E.5) and the following argument), $\mathrm{Var}_n(\hat{\vartheta}_{T_n}(\phi)) \leq \mathbb{E}_n\hat{\vartheta}_{T_n}(\phi)^2 \leq \mathbb{E}_n\hat{\vartheta}_{T_n}(\phi)$. Furthermore, since $T_n \to \infty$ by (A4), the fact that $p_n \to p \in [0, 1)$ means only the case $T_n(1 - p_n) \to \infty$ can occur. In this case, since $\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)]1(\Omega_{n,2}) \to 0$ *a.s.* by Lemma E.5, we immediately obtain from $\mathrm{Var}_n(\hat{\vartheta}_{T_n}(\phi)) \leq \mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)]$ that $\mathrm{Var}_n(\hat{\vartheta}_{T_n}(\phi))1(\Omega_{n,2}) \to 0$ *a.s.* as well. Hence, it only remains to prove the lemma in the case $p_n \to 1$, which we assume to hold for the remainder of the proof.

Towards this end, letting $\mathcal{D}$ denote the degree sequence and $\mathcal{T}$ denote the set of random variables defining the tree structure (as in Appendix E.2.2.1), we have

$$\mathrm{Var}_n(\hat{\vartheta}_{T_n}(\phi)) = \mathbb{E}_n[\mathrm{Var}(\hat{\vartheta}_{T_n}(\phi)|\mathcal{D}, \mathcal{T})] + \mathrm{Var}_n(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{D}, \mathcal{T}]). \tag{E.30}$$

We next consider the two summands in (E.30) in turn. In particular, we aim to show that each summand multiplied by $1(\Omega_{n,2})$ tends to zero *a.s.* as n tends to infinity.

For the first summand in (E.30), we use the fact that the signals are i.i.d. Bernoulli(θ) given the tree structure, as well as Lemma E.4, to write

$$\begin{aligned}\mathrm{Var}(\hat{\vartheta}_{T_n}(\phi)|\mathcal{D}, \mathcal{T}) &= \frac{1}{T_n^2}\sum_{t=0}^{T_n-1}\sum_{l=0}^{t}\sum_{\imath\in\hat{A}_l}\mathrm{Var}(\hat{s}_{T_n-t}(\imath)|\mathcal{D}, \mathcal{T})\left(\binom{t}{l}\eta^l(1-\eta)^{t-l}\prod_{j=0}^{l-1}d_{in}(\imath|j)^{-1}\right)^2 \\ &= \frac{1}{T_n^2}\sum_{t=0}^{T_n-1}\sum_{l=0}^{t}\sum_{\imath\in\hat{A}_l}\theta(1-\theta)\left(\binom{t}{l}\eta^l(1-\eta)^{t-l}\prod_{j=0}^{l-1}d_{in}(\imath|j)^{-1}\right)^2 \\ &\leq \frac{1}{T_n^2}\sum_{t=0}^{T_n-1}\sum_{l=0}^{t}\binom{t}{l}\eta^l(1-\eta)^{t-l}\sum_{\imath\in\hat{A}_l}\prod_{j=0}^{l-1}d_{in}(\imath|j)^{-1} \leq \frac{1}{T_n},\end{aligned}$$

where in the final step we have used $\sum_{\imath\in\hat{A}_l}\prod_{j=0}^{l-1}d_{in}(\imath|j)^{-1} \leq 1$ and $\sum_{l=0}^{t}\binom{t}{l}\eta^l(1-\eta)^{t-l} = 1$.

It immediately follows that $0 \leq \mathbb{E}_n[\mathrm{Var}(\hat{\vartheta}_{T_n}(\phi)|\mathcal{D},\mathcal{T})]1(\Omega_{n,2}) \leq 1/T_n$ *a.s.* Hence, because $T_n \to \infty$ as $n \to \infty$ by (A4), analysis of the first summand in (E.30) is complete.

For the second summand in (E.30), we first use the argument of (E.24) to write

$$\begin{aligned}\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{D},\mathcal{T}] &= \frac{\theta}{T_n}\sum_{t=0}^{T_n-1}\sum_{l=0}^{t}\binom{t}{l}\eta^l(1-\eta)^{t-l}\sum_{\imath\in\hat{A}_l}\prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1} \\ &= \frac{\theta}{T_n}\sum_{l=0}^{T_n-1}\sum_{\imath\in\hat{A}_l}\prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}\sum_{t=l}^{T_n-1}\binom{t}{l}\eta^l(1-\eta)^{t-l} \triangleq \frac{\theta}{T_n}\sum_{l=0}^{T_n-1} Y_l u_{T_n,l},\end{aligned}$$

where we have defined $Y_l = \sum_{\imath\in\hat{A}_l}\prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}$ and $u_{T_n,l} = \sum_{t=l}^{T_n-1}\binom{t}{l}\eta^l(1-\eta)^{t-l}$. Thus,

$$\mathrm{Var}_n(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{D},\mathcal{T}]) = \frac{\theta^2}{T_n^2}\left(\sum_{l=0}^{T_n-1} u_{T_n,l}^2\mathrm{Var}_n(Y_l) + 2\sum_{l=0}^{T_n-1} u_{T_n,l}\sum_{l'=l+1}^{T_n-1} u_{T_n,l'}\mathrm{Cov}_n(Y_l,Y_{l'})\right) \tag{E.31}$$

It remains to compute the variance and covariance in (E.31). First, for any $l, l' \in \mathbb{N}$,

$$\begin{aligned}\mathbb{E}_n[Y_l Y_{l'}] &= \mathbb{E}_n\left[\mathbb{P}(X_l^1 \in \hat{A}|\mathcal{D},\mathcal{T})\mathbb{P}(X_{l'}^2 \in \hat{A}|\mathcal{D},\mathcal{T})\right] \\ &= \mathbb{E}_n\left[\mathbb{P}(X_l^1 \in \hat{A}, X_{l'}^2 \in \hat{A}|\mathcal{D},\mathcal{T})\right] = \mathbb{P}_n(X_l^1 \in \hat{A}, X_{l'}^2 \in \hat{A}),\end{aligned} \tag{E.32}$$

where we have used the argument of (E.25) and the fact that $\{X_i^1\}_{i=1}^\infty$ and $\{X_i^2\}_{i=1}^\infty$ are independent random walks given the tree structure. By a similar argument, $\mathbb{E}_n[Y_l] = \mathbb{P}_n(X_l^1 \in \hat{A})$. Hence, using Lemmas E.7 and E.8, and assuming for the moment that $l > 1$, we have

$$\begin{aligned}\mathrm{Var}_n(Y_l) &= \mathbb{P}_n(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}) - (\mathbb{P}_n(X_l^1 \in \hat{A}))^2 \\ &= \tilde{r}_n^*\tilde{p}_n^{2(l-1)} + \sum_{t=2}^{l}\tilde{q}_n^*\tilde{q}_n^{t-2}\tilde{r}_n\tilde{p}_n^{2(l-t)} + \tilde{q}_n^*\tilde{q}_n^{l-1} - (\tilde{p}_n^*\tilde{p}_n^{l-1})^2 \\ &= \frac{1}{\tilde{p}_n^2}\left(\tilde{r}_n^* + \frac{\tilde{q}_n^*\tilde{r}_n}{\tilde{p}_n^2 - \tilde{q}_n} - (\tilde{p}_n^*)^2\right)\tilde{p}_n^{2l} + \frac{\tilde{q}_n^*}{\tilde{q}_n}\left(1 - \frac{\tilde{r}_n}{\tilde{p}_n^2 - \tilde{q}_n}\right)\tilde{q}_n^l.\end{aligned} \tag{E.33}$$

Next, using (E.7) and Jensen's inequality, we have

$$\tilde{r}_n \geq \left(\sum_{j\in\mathbb{N},k\in\mathbb{N}_0}\frac{j}{j+k}\sum_{i\in\mathbb{N}} f_n(i,j,k)\right)^2 - \tilde{q}_n = \tilde{p}_n^2 - \tilde{q}_n, \tag{E.34}$$

and so $1 - \tilde{r}_n/(\tilde{p}_n^2 - \tilde{q}_n) \leq 0$, i.e. the second term in (E.33) is non-positive, so $\forall\ l > 1$,

$$\mathrm{Var}_n(Y_l) \leq \frac{1}{\tilde{p}_n^2}\left(\tilde{r}_n^* + \frac{\tilde{q}_n^*\tilde{r}_n}{\tilde{p}_n^2 - \tilde{q}_n} - (\tilde{p}_n^*)^2\right)\tilde{p}_n^{2l}. \tag{E.35}$$

In the case $l = 1$, we have (again by Lemmas E.7 and E.8)

$$\mathrm{Var}_n(Y_l) = (\tilde{r}_n^* + \tilde{q}_n^*) - \tilde{p}_n^* \leq \tilde{r}_n^* + \frac{\tilde{q}_n^* \tilde{r}_n}{\tilde{p}_n^2 - \tilde{q}_n} - (\tilde{p}_n^*)^2 = \frac{1}{\tilde{p}_n^2}\left(\tilde{r}_n^* + \frac{\tilde{q}_n^* \tilde{r}_n}{\tilde{p}_n^2 - \tilde{q}_n} - (\tilde{p}_n^*)^2\right)\tilde{p}_n^{2l},$$

where the inequality is (E.34) and $\tilde{p}_n^* \leq 1$; hence, (E.35) holds for $l = 1$ as well. Finally, since $Y_0 = 1$ *a.s.*, it is immediate that (E.35) also holds for $l = 0$. We next analyze the covariance terms in (E.31). First, if $l' > l > 0$, we can use (E.32) and Lemmas E.7 and E.8 to obtain

$$\begin{aligned}
\mathbb{E}_n[Y_l Y_{l'}] = \mathbb{P}_n(X_l^1 \in \hat{A}, X_{l'}^2 \in \hat{A}) = \tilde{p}_n^{l'-l}\mathbb{P}_n(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}) = \tilde{p}_n^{l'-l}\mathbb{E}_n[Y_l^2], \\
\mathbb{E}_n[Y_{l'}] = \mathbb{P}(X_{l'}^2 \in \hat{A}) = \tilde{p}_n^* \tilde{p}_n^{l'-1} = \tilde{p}_n^* \tilde{p}_n^{l-1}\tilde{p}_n^{l'-l} = \mathbb{P}(X_l^1 \in \hat{A})\tilde{p}_n^{l'-l} = \mathbb{E}_n[Y_l]\tilde{p}_n^{l'-l}, \\
\Rightarrow \mathrm{Cov}_n(Y_l, Y_{l'}) = \tilde{p}_n^{l'-l}\left(\mathbb{E}_n[Y_l^2] - (\mathbb{E}_n[Y_l])^2\right) = \tilde{p}_n^{l'-l}\mathrm{Var}_n(Y_l).
\end{aligned}$$

On the other hand, if $l' > l = 0$, we have $Y_l = 1$ *a.s.*, so $\mathrm{Cov}_n(Y_l, Y_{l'}) = 0 = \tilde{p}_n^{l'}\mathrm{Var}_n(Y_0)$. Hence, combined with (E.35), we have argued

$$\mathrm{Cov}_n(Y_l, Y_{l'}) = \tilde{p}_n^{l'-l}\mathrm{Var}_n(Y_l) \leq \frac{1}{\tilde{p}_n^2}\left(\tilde{r}_n^* + \frac{\tilde{q}_n^* \tilde{r}_n}{\tilde{p}_n^2 - \tilde{q}_n} - (\tilde{p}_n^*)^2\right)\tilde{p}_n^{l+l'} \ \forall\, l \in \mathbb{N}_0, l' > l. \tag{E.36}$$

Hence, combining (E.31), (E.35), and (E.36), we obtain

$$\begin{aligned}
&\mathrm{Var}_n(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{D}, \mathcal{T}]) \\
&\quad \leq \frac{1}{\tilde{p}_n^2}\left(\tilde{r}_n^* + \frac{\tilde{q}_n^* \tilde{r}_n}{\tilde{p}_n^2 - \tilde{q}_n} - (\tilde{p}_n^*)^2\right)\frac{\theta^2}{T_n^2}\left(\sum_{l=0}^{T_n-1} u_{T_n,l}^2 \tilde{p}_n^{2l} + 2\sum_{l=0}^{T_n-1} u_{T_n,l}\sum_{l'=l+1}^{T_n-1} u_{T_n,l'}\tilde{p}_n^{l+l'}\right) \\
&\quad \leq \frac{1}{\tilde{p}_n^2}\left(\tilde{r}_n^* + \frac{\tilde{q}_n^* \tilde{r}_n}{\tilde{p}_n^2 - \tilde{q}_n} - (\tilde{p}_n^*)^2\right)\frac{1}{T_n^2}\left(\sum_{l=0}^{T_n-1} u_{T_n,l}^2 + 2\sum_{l=0}^{T_n-1} u_{T_n,l}\sum_{l'=l+1}^{T_n-1} u_{T_n,l'}\right) \\
&\quad = \frac{1}{\tilde{p}_n^2}\left(\tilde{r}_n^* + \frac{\tilde{q}_n^* \tilde{r}_n}{\tilde{p}_n^2 - \tilde{q}_n} - (\tilde{p}_n^*)^2\right)\left(\frac{1}{T_n}\sum_{l=0}^{T_n-1} u_{T_n,l}\right)^2 = \frac{1}{\tilde{p}_n^2}\left(\tilde{r}_n^* + \frac{\tilde{q}_n^* \tilde{r}_n}{\tilde{p}_n^2 - \tilde{q}_n} - (\tilde{p}_n^*)^2\right),
\end{aligned}$$

where the second inequality is simply $\theta, \tilde{p}_n \leq 1$, the first equality is immediate, and the second equality holds by definition of $u_{T_n,l}$. It clearly follows that

$$\mathrm{Var}_n(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{D}, \mathcal{T}])1(\Omega_{n,2}) \leq \frac{1}{\tilde{p}_n^2}\left(\tilde{r}_n^* + \frac{\tilde{q}_n^* \tilde{r}_n}{\tilde{p}_n^2 - \tilde{q}_n} - (\tilde{p}_n^*)^2\right)1(\Omega_{n,2}), \tag{E.37}$$

and so we aim to show the right side of (E.37) tends to zero *a.s.* Clearly, the right side is zero if $\omega \notin \Omega_{n,2}$; we aim to also show that, given $\gamma > 0$, $\exists$ N s.t. for $n > N$ and $\omega \in \Omega_{n,2}$,

$$\frac{1}{\tilde{p}_n(\omega)^2}\left(\tilde{r}_n^*(\omega) + \frac{\tilde{q}_n^*(\omega)\tilde{r}_n(\omega)}{\tilde{p}_n(\omega)^2 - \tilde{q}_n(\omega)} - \tilde{p}_n^*(\omega)^2\right) < \gamma. \tag{E.38}$$

To prove (E.38), we first recall that by (A3), we have for $\omega \in \Omega_{n,2}$, $\tilde{p}_n^*(\omega) \geq \tilde{p}_n(\omega) > p_n - \delta_n$.

Hence, since we are assuming $p_n \to 1$, and since $\delta_n \to 0$ by (A3), we have for $\gamma' > 0$, n sufficiently large, and such ω, $\tilde{p}_n(\omega)^2, \tilde{p}_n^*(\omega)^2 > 1 - \gamma'$. Thus, for n large and $\omega \in \Omega_{n,2}$,

$$\frac{1}{\tilde{p}_n(\omega)^2}\left(\tilde{r}_n^*(\omega) + \frac{\tilde{q}_n^*(\omega)\tilde{r}_n(\omega)}{\tilde{p}_n(\omega)^2 - \tilde{q}_n(\omega)} - \tilde{p}_n^*(\omega)^2\right) < \frac{1}{1-\gamma'}\left(\tilde{r}_n^*(\omega) + \frac{\tilde{q}_n^*(\omega)\tilde{r}_n(\omega)}{1-\gamma' - \tilde{q}_n(\omega)} - (1-\gamma')\right). \tag{E.39}$$

To further upper bound the right side of (E.39), we note $\tilde{r}_n \leq 1 - \tilde{q}_n$ *a.s.* by the first equality in (E.34). The same argument gives $\tilde{r}_n^* \leq 1 - \tilde{q}_n^*$ *a.s.* Note, however, that to use the second bound, we must ensure $1 - \gamma' - \tilde{q}_n(\omega) > 0$. To this end, recall that $\tilde{q}_n(\omega) < 1 - \xi$ for $\omega \in \Omega_{n,2}$ by (A3). Hence, assuming we choose $\gamma' < \xi$, we obtain $1 - \gamma' - \tilde{q}_n(\omega) > 0$ for such ω. Thus,

$$\begin{aligned}
&\frac{1}{\tilde{p}_n(\omega)^2}\left(\tilde{r}_n^*(\omega) + \frac{\tilde{q}_n^*(\omega)\tilde{r}_n(\omega)}{\tilde{p}_n(\omega)^2 - \tilde{q}_n(\omega)} - \tilde{p}_n^*(\omega)^2\right) \\
&\quad < \frac{1}{1-\gamma'}\left((1 - \tilde{q}_n^*(\omega)) + \frac{\tilde{q}_n^*(\omega)(1-\tilde{q}_n(\omega))}{1-\gamma'-\tilde{q}_n(\omega)} - (1-\gamma')\right) \\
&\quad = \frac{\gamma'}{1-\gamma'}\left(\frac{\tilde{q}_n^*(\omega)}{1-\gamma'-\tilde{q}_n(\omega)} + 1\right) < \frac{\gamma'}{1-\gamma'}\left(\frac{\tilde{q}_n^*(\omega)}{\xi - \gamma'} + 1\right) \leq \frac{\gamma'}{1-\gamma'}\left(\frac{1}{\xi - \gamma'} + 1\right)
\end{aligned} \tag{E.40}$$

where the first inequality uses (E.39) and the bounds from the previous paragraph, the equalities are straightforward, the second inequality uses $\tilde{q}_n(\omega) < 1 - \xi$ for $\omega \in \Omega_{n,2}$ by (A3), and the third uses $\tilde{q}_n^*(\omega) \leq 1$ (recall we have chosen $\gamma' < \xi$). Finally, it is straightforward to see the final bound in (E.40) tends to zero with γ'. Hence, for sufficiently small γ', (E.38) follows, completing the proof.

E.2.2.3 Notation for proofs of Lemmas E.7 and E.8

In the next two subsections, we prove Lemmas E.7 and E.8. For these proofs, we let $\mathcal{D}$ denote the degree sequence $\{d_{out}(i), d_{in}^A(i), d_{in}^B(i)\}_{i\in[n]}$, and we let D denote a realization of this set. Note that the random variables defined in (E.7) are all functions of $\mathcal{D}$; for a realization D of $\mathcal{D}$, we let e.g. $\tilde{p}_{n,D}$ denote the realization of $\tilde{p}_n$. We similarly define $f_{n,D}, f_{n,D}^*$ for realizations of f_n, f_n^*, defined in (6.7). Finally, letting $g(D) = \mathbb{P}(\cdot|\mathcal{D} = D)$, we have $\mathbb{P}_n(\cdot) = g(\mathcal{D})$ by definition of $\mathbb{P}_n$. Hence, to prove Lemma E.7, it suffices to show

$$\mathbb{P}(X_l \in \hat{A} | \mathcal{D} = D) = \begin{cases} \tilde{p}_{n,D}^* \tilde{p}_{n,D}^{l-1}, & l \in \mathbb{N} \\ 1, & l = 0 \end{cases}.$$

while to prove Lemma E.8, it suffices to show

$$\mathbb{P}(X_l^1 \in \hat{A}, X_{l'}^2 \in \hat{A} | \mathcal{D} = D) = \begin{cases} \mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A} | \mathcal{D} = D)\tilde{p}_{n,D}^{l'-l}, & l \in \mathbb{N} \\ \tilde{p}_{n,D}^* \tilde{p}_{n,D}^{l'-1}, & l = 0 \end{cases}, \tag{E.41}$$

$$\mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A} | \mathcal{D} = D) = \begin{cases} \tilde{r}_{n,D}^* \tilde{p}_{n,D}^{2(l-1)} + \sum_{t=2}^{l} \tilde{q}_{n,D}^* \tilde{q}_{n,D}^{t-2} \tilde{r}_{n,D} \tilde{p}_{n,D}^{2(l-t)} + \tilde{q}_{n,D}^* \tilde{q}_{n,D}^{l-1}, & l > 1 \\ \tilde{r}_{n,D}^* + \tilde{q}_{n,D}^*, & l = 1 \\ 1, & l = 0 \end{cases} \tag{E.42}$$

E.2.2.4 Proof of Lemma E.7

The $l = 0$ case is trivial, since $X_0^1 = \phi \in \hat{A}$, so we assume $l \in \mathbb{N}$ moving forward. First, since $\hat{A}^C = \hat{B}$ is an absorbing set, we have $X_l^1 \in \hat{A} \Rightarrow X_{l-1}^1 \in \hat{A}$, so

$$\mathbb{P}(X_l^1 \in \hat{A} | \mathcal{D} = D) = \mathbb{P}(X_l^1 \in \hat{A} | X_{l-1}^1 \in \hat{A}, \mathcal{D} = D)\mathbb{P}(X_{l-1}^1 \in \hat{A} | \mathcal{D} = D). \tag{E.43}$$

For the first term in (E.43), we have

$$\begin{aligned}
&\mathbb{P}(X_l^1 \in \hat{A} | X_{l-1}^1 \in \hat{A}, \mathcal{D} = D) \qquad\qquad (E.44)\\
&= \sum_{j \in \mathbb{N}, k \in \mathbb{N}_0} \mathbb{P}(X_l^1 \in \hat{A} | d_{in}^A(X_{l-1}^1) = j, d_{in}^B(X_{l-1}^1) = k, X_{l-1}^1 \in \hat{A}, \mathcal{D} = D) \\
&\qquad\qquad \times \mathbb{P}(d_{in}^A(X_{l-1}^1) = j, d_{in}^B(X_{l-1}^1) = k | X_{l-1}^1 \in \hat{A}, \mathcal{D} = D) \\
&= \begin{cases} \sum_{j \in \mathbb{N}, k \in \mathbb{N}_0} \frac{j}{j+k} \sum_{i \in \mathbb{N}} f_{n,D}(i,j,k) = \tilde{p}_{n,D}, & l \in \{2, 3, \ldots\} \\ \sum_{j \in \mathbb{N}, k \in \mathbb{N}_0} \frac{j}{j+k} \sum_{i \in \mathbb{N}} f_{n,D}^*(i,j,k) = \tilde{p}_{n,D}^*, & l = 1 \end{cases},
\end{aligned}$$

where the second equality holds by Algorithm E.2. Specifically, for $l > 1$, the degrees of X_{l-1}^1 are sampled from $f_{n,D}$ (Line 11) after realizing X_{l-1}^1 (Line 8), yielding the $\sum_{i \in \mathbb{N}} f_{n,D}(i,j,k)$ term; further, X_l^1 is chosen uniformly from the incoming neighbors of X_{l-1}^1 (Line 8) after realizing the degrees of X_{l-1}^1, yielding the $j/(j+k)$ term (the $l = 1$ case is similarly justified). Combining (E.43) and (E.44), and using the fact that $X_0^1 = \phi \in \hat{A}$ by definition, completes the proof in the case $l = 1$. For $l > 1$, we again use (E.43) and (E.44) to obtain

$$\mathbb{P}(X_l^1 \in \hat{A} | \mathcal{D} = D) = \tilde{p}_{n,D} \mathbb{P}(X_{l-1}^1 \in \hat{A} | \mathcal{D} = D) = \cdots = \tilde{p}_{n,D}^{l-1} \mathbb{P}(X_1^1 \in \hat{A} | \mathcal{D} = D) = \tilde{p}_{n,D}^{l-1} \tilde{p}_{n,D}^*.$$

E.2.2.5 Proof of Lemma E.8

We begin by proving the first statement in the lemma, i.e. (E.41). First, we note that for the $l = 0$ case, $X_0 = \phi \in \hat{A}$ by definition, so $\mathbb{P}(X_l^1 \in \hat{A}, X_{l'}^2 \in \hat{A} | \mathcal{D} = D) = \mathbb{P}(X_{l'}^2 \in \hat{A} | \mathcal{D} = D)$, and the statement holds by Lemma E.7. For the $l \in \mathbb{N}$ case, we first write

$$\begin{aligned}
\mathbb{P}(X_l^1 \in \hat{A}, X_{l'}^2 \in \hat{A} | \mathcal{D} = D) &= \mathbb{P}(X_l^1 \in \hat{A}, X_{l'-1}^2 \in \hat{A}, X_{l'}^2 \in \hat{A} | \mathcal{D} = D) \\
&= \mathbb{P}(X_{l'}^2 \in \hat{A} | X_l^1 \in \hat{A}, X_{l'-1}^2 \in \hat{A}, \mathcal{D} = D)\mathbb{P}(X_l^1 \in \hat{A}, X_{l'-1}^2 \in \hat{A} | \mathcal{D} = D),
\end{aligned}$$

where the first equality holds since $\hat{A}^C = \hat{B}$ is an absorbing set (i.e. $X_{l'}^2 \in \hat{A} \Rightarrow X_{l'-1}^2 \in \hat{A}$) and the second rewrites a conditional probability. Next, by the same argument as (E.44),

$$\mathbb{P}(X_{l'}^2 \in \hat{A} | X_l^1 \in \hat{A}, X_{l'-1}^2 \in \hat{A}, \mathcal{D} = D) = \tilde{p}_{n,D},$$

where we used the $l' > 1$ case of (E.44), since $l' > l \geq 1$. Hence,

$$\begin{aligned}
\mathbb{P}(X_l^1 \in \hat{A}, X_{l'}^2 \in \hat{A} | \mathcal{D} = D) &= \tilde{p}_{n,D} \mathbb{P}(X_l^1 \in \hat{A}, X_{l'-1}^2 \in \hat{A} | \mathcal{D} = D) \\
&= \cdots = \tilde{p}_{n,D}^{l'-l} \mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A} | \mathcal{D} = D).
\end{aligned}$$

This establishes (E.41). For the second statement, i.e. (E.42), the $l = 0$ case is trivial, since $X_0^1 = X_0^2 = \phi \in \hat{A}$ by definition, so we assume $l \in \mathbb{N}$ for the remainder of the proof. First, let $\tau = \inf\{t \in \mathbb{N}_0 : X_t^1 \neq X_t^2\}$ denote the first step at which the walks diverge. Note $X_0^1 = X_0^2 = \phi$ by definition, so $\tau \in \mathbb{N}$ *a.s.*; also, due to the tree structure, the walks remain apart forever after diverging, i.e. $X_{\tau+1}^1 \neq X_{\tau+1}^2, X_{\tau+2}^1 \neq X_{\tau+1}^2, \dots$ *a.s.* Next, for $l \in \mathbb{N}$,

$$\begin{aligned}&\mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A} | \mathcal{D} = D) \\ &\quad = \sum_{t=1}^{l} \mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, \tau = t | \mathcal{D} = D) + \mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, \tau > l | \mathcal{D} = D)\end{aligned} \tag{E.45}$$

We begin by computing the second term in (E.45). Here we have

$$\begin{aligned}&\mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, \tau > l | \mathcal{D} = D) \\ &= \mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, X_l^1 = X_l^2 | X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, X_{l-1}^1 = X_{l-1}^2, \dots, X_1^1 = X_1^2, \mathcal{D} = D) \\ &\qquad \times \mathbb{P}(X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, \tau > l-1 | \mathcal{D} = D),\end{aligned} \tag{E.46}$$

where we used the definition of τ and that $\hat{A}^C = \hat{B}$ is an absorbing set. Now for $l > 1$,

$$\begin{aligned}&\mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, X_l^1 = X_l^2 | X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, X_{l-1}^1 = X_{l-1}^2, \dots, X_1^1 = X_1^2, \mathcal{D} = D) \\ &\quad = \mathbb{P}(X_l^1 \in \hat{A}, X_l^1 = X_l^2 | X_{l-1}^1 \in \hat{A}, X_{l-1}^1 = X_{l-1}^2, \mathcal{D} = D) \\ &\quad = \sum_{j \in \mathbb{N}, k \in \mathbb{N}_0} \frac{j}{j+k} \frac{1}{j+k} \sum_{i \in \mathbb{N}} f_{n,D}(i,j,k) = \tilde{q}_{n,D},\end{aligned} \tag{E.47}$$

where the first equality uses independence and eliminates repetitive events, and the second follows an argument similar to that following (E.44). Combining (E.46) and (E.47),

$$\begin{aligned}\mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, \tau > l | \mathcal{D} = D) &= \tilde{q}_{n,D} \mathbb{P}(X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, \tau > l-1 | \mathcal{D} = D) \\ &= \cdots = \tilde{q}_{n,D}^{l-1} \mathbb{P}(X_1^1 \in \hat{A}, X_1^2 \in \hat{A}, \tau > 1 | \mathcal{D} = D).\end{aligned} \tag{E.48}$$

Finally, by an argument similar to (E.47), we have

$$\begin{aligned}&\mathbb{P}(X_1^1 \in \hat{A}, X_1^2 \in \hat{A}, \tau > 1 | \mathcal{D} = D) = \mathbb{P}(X_1^1 \in \hat{A}, X_1^1 = X_1^2 | \mathcal{D} = D) \\ &\quad = \sum_{j \in \mathbb{N}, k \in \mathbb{N}_0} \frac{j}{j+k} \frac{1}{j+k} \sum_{i \in \mathbb{N}} f_{n,D}^*(i,j,k) = \tilde{q}_{n,D}^*.\end{aligned} \tag{E.49}$$

Hence, combining (E.48) and (E.49) gives

$$\mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, \tau > l | \mathcal{D} = D) = \tilde{q}_{n,D}^* \tilde{q}_{n,D}^{l-1} \ \forall \ l \in \mathbb{N}. \tag{E.50}$$

For the first term in (E.45), we first consider the $t = l$ summand. For $l > 1$, similar to (E.47),

$$\mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, \tau = l | \mathcal{D} = D)$$

$$
\begin{aligned}
&= \mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, X_l^1 \neq X_l^2, X_{l-1}^1 = X_{l-1}^2, \ldots, X_1^1 = X_1^2 | \mathcal{D} = D) \\
&= \mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, X_l^1 \neq X_l^2, X_{l-1}^1 \in \hat{A}, X_{l-1}^1 = X_{l-1}^2, \ldots, X_1^1 = X_1^2 | \mathcal{D} = D) \\
&= \mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, X_l^1 \neq X_l^2 | X_{l-1}^1 \in \hat{A}, X_{l-1}^1 = X_{l-1}^2, \ldots, X_1^1 = X_1^2, \mathcal{D} = D) \\
&\quad \times \mathbb{P}(X_{l-1}^1 \in \hat{A}, X_{l-1}^1 = X_{l-1}^2, \ldots, X_1^1 = X_1^2 | \mathcal{D} = D) \\
&= \sum_{j \in \mathbb{N}, k \in \mathbb{N}_0} \frac{j}{j+k} \frac{j-1}{j+k} \sum_{i \in \mathbb{N}} f_{n,D}(i,j,k) \mathbb{P}(X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, \tau > l-1 | \mathcal{D} = D) \\
&= \tilde{r}_{n,D} \tilde{q}_{n,D}^{l-2} \tilde{q}_{n,D}^*,
\end{aligned}
$$

where in the final step we have also used (E.50). Similarly, for $l = 1$,

$$
\begin{aligned}
\mathbb{P}(X_1^1 \in \hat{A}, X_1^2 \in \hat{A}, \tau = 1 | \mathcal{D} = D) &= \mathbb{P}(X_1^1 \in \hat{A}, X_1^2 \in \hat{A}, X_1^1 \neq X_1^2 | \mathcal{D} = D) \\
&= \sum_{j \in \mathbb{N}, k \in \mathbb{N}_0} \frac{j}{j+k} \frac{j-1}{j+k} \sum_{i \in \mathbb{N}} f_{n,D}^*(i,j,k) = \tilde{r}_{n,D}^*.
\end{aligned}
$$

To summarize, we have shown

$$
\mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, \tau = l | \mathcal{D} = D) = \begin{cases} \tilde{q}_{n,D}^* \tilde{q}_{n,D}^{l-2} \tilde{r}_{n,D}, & l \in \{2, 3, \ldots\} \\ \tilde{r}_{n,D}^*, & l = 1 \end{cases}. \tag{E.51}
$$

Next, we consider the $t < l$ summands in (E.45) (which are present only for $l > 1$). We have

$$
\begin{aligned}
&\mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, \tau = t | \mathcal{D} = D) \\
&\quad = \mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, X_{l-1}^1 \neq X_{l-1}^2, \tau = t | \mathcal{D} = D) \\
&\quad = \mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A} | X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, X_{l-1}^1 \neq X_{l-1}^2, \tau = t, \mathcal{D} = D) \\
&\qquad \times \mathbb{P}(X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, X_{l-1}^1 \neq X_{l-1}^2, \tau = t | \mathcal{D} = D) \\
&\quad = \prod_{h=1}^{2} \mathbb{P}(X_l^h \in \hat{A} | X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, X_{l-1}^1 \neq X_{l-1}^2, \tau = t, \mathcal{D} = D) \\
&\qquad \times \mathbb{P}(X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, \tau = t | \mathcal{D} = D),
\end{aligned}
$$

where in the first equality we used the fact that $\hat{A}^C = \hat{B}$ is an absorbing set and the fact that once the walks diverge they remain apart; in the second equality we used the fact that X_l^1 and X_l^2 are conditionally independent given the event $X_{l-1}^1 \neq X_{l-1}^2$. Further, for $h \in \{1, 2\}$,

$$
\begin{aligned}
&\mathbb{P}(X_l^h \in \hat{A} | X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, X_{l-1}^1 \neq X_{l-1}^2, \tau = t, \mathcal{D} = D) \\
&\quad = \sum_{j \in \mathbb{N}, k \in \mathbb{N}_0} \frac{j}{j+k} \sum_{i \in \mathbb{N}} f_{n,D}(i,j,k) = \tilde{p}_{n,D},
\end{aligned}
$$

and so, combining the previous two equations and applying recursively yields

$$
\mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A}, \tau = t | \mathcal{D} = D) = \tilde{p}_{n,D}^2 \mathbb{P}(X_{l-1}^1 \in \hat{A}, X_{l-1}^2 \in \hat{A}, \tau = t | \mathcal{D} = D) \tag{E.52}
$$

$$= \cdots = \tilde{p}_{n,D}^{2(l-t)} \mathbb{P}(X_t^1 \in \hat{A}, X_t^2 \in \hat{A}, \tau = t | \mathcal{D} = D)$$
$$= \begin{cases} \tilde{q}_{n,D}^* \tilde{q}_{n,D}^{t-2} \tilde{r}_{n,D} \tilde{p}_{n,D}^{2(l-t)}, & t \in \{2, 3, \ldots, l-1\} \\ \tilde{r}_{n,D}^* \tilde{p}_{n,D}^{2(l-1)}, & t = 1 \end{cases},$$

where the final equality uses (E.51). Combining (E.45), (E.50), (E.51), and (E.52),

$$\mathbb{P}(X_l^1 \in \hat{A}, X_l^2 \in \hat{A} | \mathcal{D} = D) = \begin{cases} \tilde{r}_{n,D}^* \tilde{p}_{n,D}^{2(l-1)} + \sum_{t=2}^{l} \tilde{q}_{n,D}^* \tilde{q}_{n,D}^{t-2} \tilde{r}_{n,D} \tilde{p}_{n,D}^{2(l-t)} + \tilde{q}_{n,D}^* \tilde{q}_{n,D}^{l-1}, & l > 1 \\ \tilde{r}_{n,D}^* + \tilde{q}_{n,D}^*, & l = 1 \end{cases}.$$

E.2.3 Step 2 for proof of Theorem 6.2

E.2.3.1 Proof of Lemma E.9

We first write

$$\mathbb{P}\left(\left|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]\right| > \varepsilon\right) = \mathbb{E}\left[\mathbb{P}\left(\left|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]\right| > \varepsilon \middle| \mathcal{T}\right)\right]$$
$$= \mathbb{E}\left[\mathbb{P}\left(\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon \middle| \mathcal{T}\right) + \mathbb{P}\left(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] - \hat{\vartheta}_{T_n}(\phi) > \varepsilon \middle| \mathcal{T}\right)\right] \quad \text{(E.53)}$$

where the first equality uses the law of total expectation and the second is immediate. For the first summand in the expectation in (E.53), we fix $\lambda > 0$ and write

$$\mathbb{P}\left(\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon \middle| \mathcal{T}\right) = \mathbb{P}\left(\exp(\lambda(\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}])) > e^{-\lambda\varepsilon} \middle| \mathcal{T}\right)$$
$$\leq e^{-\lambda\varepsilon} \mathbb{E}\left[\exp(\lambda(\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]) \middle| \mathcal{T}\right]$$
$$= e^{-\lambda\varepsilon} \prod_{t=0}^{T_n-1} \prod_{l=0}^{t} \prod_{\imath \in \hat{A}_l} \mathbb{E}\left[\exp\left(\frac{\lambda}{T_n} \binom{t}{l} \eta^l (1-\eta)^{t-l} \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1} (\hat{s}_{T_n - t}(\imath) - \theta)\right) \middle| \mathcal{T}\right]$$
$$\leq e^{-\lambda\varepsilon} \prod_{t=0}^{T_n-1} \prod_{l=0}^{t} \prod_{\imath \in \hat{A}_l} \exp\left(\frac{1}{8}\left(\frac{\lambda}{T_n} \binom{t}{l} \eta^l (1-\eta)^{t-l} \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}\right)^2\right)$$
$$\leq e^{-\lambda\varepsilon} \prod_{t=0}^{T_n-1} \prod_{l=0}^{t} \prod_{\imath \in \hat{A}_l} \exp\left(\frac{\lambda^2}{8T_n^2} \binom{t}{l} \eta^l (1-\eta)^{t-l} \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}\right)$$
$$= \exp\left(-\lambda\varepsilon + \frac{\lambda^2}{8T_n} \frac{1}{T_n} \sum_{t=0}^{T_n-1} \sum_{l=0}^{t} \sum_{\imath \in \hat{A}_l} \binom{t}{l} \eta^l (1-\eta)^{t-l} \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}\right)$$
$$= \exp\left(-\lambda\varepsilon + \frac{\lambda^2}{8T_n\theta} \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]\right) \leq \exp\left(-\lambda\varepsilon + \frac{\lambda^2}{8T_n}\right). \quad \text{(E.54)}$$

Here the first equality holds by monotonicity of $x \mapsto e^{\lambda x}$, the first inequality is Markov's, the second equality holds by (E.14), the second inequality uses Lemma E.13 from Appendix E.2.4, the third inequality uses $\binom{t}{l}\eta^l(1-\eta)^{t-l}, \prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1} \leq 1$, the third equality is immediate, the fourth equality again uses (E.14), and the fourth inequality uses (E.15). Since the preceding argument holds $\forall\ \lambda > 0$, we choose $\lambda = 4\varepsilon T_n$ to minimize the bound.

Upon substituting into (E.54), we obtain $e^{-2\varepsilon^2 T_n}$. The same argument holds for the second summand in the expectation of (E.53). We also note that the bound $e^{-2\varepsilon^2 T_n}$ is non-random, so we may discard the expectation. In summary, we have shown

$$\mathbb{P}\left(\left|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]\right| > \varepsilon\right) \leq 2e^{-2\varepsilon^2 T_n}.$$

Hence, for n sufficiently large, we have by assumption on T_n

$$\mathbb{P}\left(\left|\hat{\vartheta}_{T_n}(\phi) - \mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]\right| > \varepsilon\right) \leq 2e^{-2\varepsilon^2 \mu \log n} = 2n^{-2\varepsilon^2\mu} = O\left(n^{-2\varepsilon^2\mu}\right),$$

which is what we set out to prove.

E.2.3.2 Proof of Lemma E.10

We begin by deriving a bound conditioned on the degree sequence. First, we fix $\tilde{\lambda} > 0$ and use monotonicity of $x \mapsto e^{\tilde{\lambda}x}$ and Markov's inequality to write

$$\mathbb{P}_n(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon) \leq e^{-\tilde{\lambda}\varepsilon}\mathbb{E}_n \exp(\tilde{\lambda}\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]). \tag{E.55}$$

The bulk of the proof will involve bounding the expectation term. For this, we first note

$$\begin{aligned}
\mathbb{E}_n \exp(\tilde{\lambda}\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]) &= \mathbb{E}_n \exp\left(\tilde{\lambda}\frac{\theta}{T_n}\sum_{t=0}^{T_n-1}\sum_{l=0}^{t}\binom{t}{l}\eta^l(1-\eta)^{t-l}\sum_{\imath\in\hat{A}_l}\prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}\right) \\
&= \mathbb{E}_n \exp\left(\frac{\tilde{\lambda}\theta}{T_n}\sum_{l=0}^{T_n-1}\left(\sum_{t=l}^{T_n-1}\binom{t}{l}\eta^l(1-\eta)^{t-l}\right)\left(\sum_{\imath\in\hat{A}_l}\prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}\right)\right) \\
&= \mathbb{E}_n \prod_{l=0}^{T_n-1}\exp(\lambda u_{T_n,l} Y_l),
\end{aligned}$$

where the first equality holds by (E.14), the second rearranges summations, and in the third we defined $\lambda = \tilde{\lambda}\theta/T_n$, $u_{T_n,l} = \sum_{t=l}^{T_n-1}\binom{t}{l}\eta^l(1-\eta)^{t-l}$, and $Y_l = \sum_{\imath\in\hat{A}_l}\prod_{j=0}^{l-1} d_{in}(\imath|j)^{-1}$. For the remainder of the proof, we use $\mathbb{E}_{n,l}$ to denote conditional expectation with respect to the degree sequence *and* the set of random variables realized during the first l iterations of Algorithm E.2 (i.e. the first l generations of the tree). Using this notation, we have

$$\begin{aligned}
\mathbb{E}_n\Big[\prod_{l=0}^{T_n-1}\exp(\lambda u_{T_n,l}Y_l)\Big] &= \mathbb{E}_n\Big[\mathbb{E}_{n,T_n-2}\Big[\prod_{l=0}^{T_n-1}\exp(\lambda u_{T_n,l}Y_l)\Big]\Big] \qquad (E.56)\\
&= \mathbb{E}_n\Big[\prod_{l=0}^{T_n-2}\exp(\lambda u_{T_n,l}Y_l)\mathbb{E}_{n,T_n-2}\Big[\exp(\lambda u_{T_n,T_n-1}Y_{T_n-1})\Big]\Big] \\
&= \mathbb{E}_n\Big[\prod_{l=0}^{T_n-3}\exp(\lambda u_{T_n,l}Y_l)\exp(\lambda(u_{T_n,T_n-2}+u_{T_n,T_n-1}\tilde{p}_n)Y_{T_n-2}) \\
&\quad\times\mathbb{E}_{n,T_n-2}\Big[\exp(\lambda u_{T_n,T_n-1}(Y_{T_n-1}-\tilde{p}_nY_{T_n-2}))\Big]\Big],
\end{aligned}$$

where in the third equality we multiplied and divided $\exp(\lambda u_{T_n,T_n-1}\tilde{p}_n Y_{T_n-2})$. Next, note

$$\begin{aligned} Y_{T_n-1} &= \sum_{\imath'\in\hat{A}_{T_n-2}} \sum_{\imath\in\hat{A}_{T_n-1}:\imath|(T_n-2)=\imath'} \prod_{j=0}^{T_n-2} d_{in}(\imath|j)^{-1} \\ &= \sum_{\imath'\in\hat{A}_{T_n-2}} \sum_{\imath\in\hat{A}_{T_n-1}:\imath|(T_n-2)=\imath'} \prod_{j=0}^{T_n-2} d_{in}(\imath'|j)^{-1} \\ &= \sum_{\imath'\in\hat{A}_{T_n-2}} \prod_{j=0}^{T_n-2} d_{in}(\imath'|j)^{-1} |\{\imath\in\hat{A}_{T_n-1}:\imath|(T_n-2)=\imath'\}| \\ &= \sum_{\imath'\in\hat{A}_{T_n-2}} \prod_{j=0}^{T_n-3} d_{in}(\imath'|j)^{-1} d_{in}(\imath')^{-1} d^A_{in}(\imath'), \end{aligned} \tag{E.57}$$

where in the first equality we rewrote the sum based on the construction of $\hat{A}_{T_n-1}$ in Algorithm E.2, in the second we used the fact that $\imath|j = \imath'|j$ for $j \in \{0,\dots,T_n-2\}$ by Algorithm E.2 (in words, $\imath$ and $\imath'$ share the same ancestry in the tree), in the third we have recognized that the $\imath$-th summand does not depend on $\imath$, and in the fourth we used $\imath'|(T_n-2) = \imath'$ (since $\imath' \in \hat{A}_{T_n-2}$) and the construction of the offspring of $\imath'$ in Algorithm E.2. It follows that

$$\mathbb{E}_{n,T_n-2} Y_{T_n-1} = \sum_{\imath'\in\hat{A}_{T_n-2}} \prod_{j=0}^{T_n-3} d_{in}(\imath'|j)^{-1} \mathbb{E}_{n,T_n-2}(d^A_{in}(\imath')/d_{in}(\imath')) = \prod_{j=0}^{T_n-3} d_{in}(\imath'|j)^{-1}\tilde{p}_n = Y_{T_n-2}\tilde{p}_n,$$

where $\mathbb{E}_{n,T_n-2}(d^A_{in}(\imath')/d_{in}(\imath')) = \tilde{p}_n$ holds by definition of $d^A_{in}(\imath'), d_{in}(\imath')$ in Algorithm E.2 and of $\tilde{p}_n$ from (E.7). In summary, we have argued $\mathbb{E}_{n,T_n-2}(Y_{T_n-1} - Y_{T_n-2}\tilde{p}_n) = 0$. On the other hand, $0 \le Y_{T_n-1} \le Y_{T_n-2} \le \cdots \le Y_0 = 1$, where the first inequality holds since Y_{T_n-1} is a sum of nonnegative terms and the second holds by (E.57) (using $d_{in}(\imath') = d^A_{in}(\imath') + d^B_{in}(\imath') \ge d^A_{in}(\imath')$), and where $Y_0 = 1$ by definition. Hence, we can use Lemma E.13 from Appendix E.2.4 to obtain

$$\mathbb{E}_{n,T_n-2} \exp(\lambda u_{T_n,T_n-1}(Y_{T_n-1} - \tilde{p}_n Y_{T_n-2})) \le e^{\lambda^2 u^2_{T_n,T_n-1}/8}. \tag{E.58}$$

Substituting into (E.56) then yields

$$\begin{aligned} &\mathbb{E}_n\Big[\prod_{l=0}^{T_n-1} \exp(\lambda u_{T_n,l} Y_l)\Big] \\ &\quad\le \mathbb{E}_n\Big[\prod_{l=0}^{T_n-3} \exp(\lambda u_{T_n,l} Y_l) \exp(\lambda(u_{T_n,T_n-2} + u_{T_n,T_n-1}\tilde{p}_n) Y_{T_n-2})\Big] \exp\left(\frac{\lambda^2}{8} u^2_{T_n,T_n-1}\right). \end{aligned} \tag{E.59}$$

We can then iteratively apply the preceding argument. Namely, we have

$$\mathbb{E}_n\Big[\prod_{l=0}^{T_n-3} \exp(\lambda u_{T_n,l} Y_l) \exp(\lambda(u_{T_n,T_n-2} + u_{T_n,T_n-1}\tilde{p}_n) Y_{T_n-2})\Big] \exp\left(\frac{\lambda^2}{8} u^2_{T_n,T_n-1}\right)$$

$$= \mathbb{E}_n\Big[\prod_{l=0}^{T_n-4} \exp(\lambda u_{T_n,l} Y_l) \exp(\lambda(u_{T_n,T_n-3} + u_{T_n,T_n-2}\tilde{p}_n + u_{T_n,T_n-1}\tilde{p}_n^2) Y_{T_n-3})$$

$$\times \mathbb{E}_{n,T_n-3}\Big[\exp(\lambda(u_{T_n,T_n-2} + u_{T_n,T_n-1}\tilde{p}_n)(Y_{T_n-2} - \tilde{p}_n Y_{T_n-3}))\Big]\Big] \exp\left(\frac{\lambda^2}{8} u_{T_n,T_n-1}^2\right)$$

$$\leq \mathbb{E}_n\Big[\prod_{l=0}^{T_n-4} \exp(\lambda u_{T_n,l} Y_l) \exp(\lambda(u_{T_n,T_n-3} + u_{T_n,T_n-2}\tilde{p}_n + u_{T_n,T_n-1}\tilde{p}_n^2) Y_{T_n-3})\Big] \tag{E.60}$$

$$\times \exp\left(\frac{\lambda^2}{8}\left((u_{T_n,T_n-2} + u_{T_n,T_n-1}\tilde{p}_n)^2 + u_{T_n,T_n-1}^2\right)\right) \tag{E.61}$$

$$\leq \cdots \leq \mathbb{E}_n\Big[\exp(\lambda u_{T_n,0} Y_0) \exp\left(\lambda \sum_{l=1}^{T_n-1} u_{T_n,l} \tilde{p}_n^{l-1} Y_1\right)\Big] \exp\left(\frac{\lambda^2}{8} \sum_{l=2}^{T_n-1} \left(\sum_{l'=l}^{T_n-1} u_{T_n,l'} \tilde{p}_n^{l'-l}\right)^2\right). \tag{E.62}$$

(The precise form of the summations in (E.62) can be verified by considering the case $T_n = 4$ in (E.60) and (E.61).) Note that the final step of the iteration is different because the root node has degrees sampled from f_n^* (the uniform distribution) instead of f_n (the size-biased distribution) in Algorithm E.2. Nevertheless, a similar argument holds: here we have $\mathbb{E}_{n,0} Y_1 = \tilde{p}_n^* Y_0$ and $Y_1 \in [0,1]$ *a.s.*, so by an argument similar to that leading to (E.58),

$$\mathbb{E}_n\Big[\exp(\lambda u_{T_n,0} Y_0) \exp\left(\lambda \sum_{l=1}^{T_n-1} u_{T_n,l} \tilde{p}_n^{l-1} Y_1\right)\Big]$$

$$= \mathbb{E}_n\Big[\exp\left(\lambda\left(u_{T_n,0} + \tilde{p}_n^* \sum_{l=1}^{T_n-1} u_{T_n,l} \tilde{p}_n^{l-1}\right) Y_0\right) \mathbb{E}_{n,0}\Big[\exp\left(\lambda \sum_{l=1}^{T_n-1} u_{T_n,l} \tilde{p}_n^{l-1}(Y_1 - \tilde{p}_n^* Y_0)\right)\Big]\Big]$$

$$\leq \mathbb{E}_n\Big[\exp\left(\lambda\left(u_{T_n,0} + \tilde{p}_n^* \sum_{l=1}^{T_n-1} u_{T_n,l} \tilde{p}_n^{l-1}\right) Y_0\right)\Big] \exp\left(\frac{\lambda^2}{8}\left(\sum_{l=1}^{T_n-1} u_{T_n,l} \tilde{p}_n^{l-1}\right)^2\right).$$

Combining the previous inequality with (E.59) and (E.62) then yields

$$\mathbb{E}_n\Big[\prod_{l=0}^{T_n-1} \exp(\lambda u_{T_n,l} Y_l)\Big]$$

$$\leq \mathbb{E}_n\Big[\exp\left(\lambda\left(u_{T_n,0} + \tilde{p}_n^* \sum_{l=1}^{T_n-1} u_{T_n,l} \tilde{p}_n^{l-1}\right) Y_0\right)\Big] \exp\left(\frac{\lambda^2}{8} \sum_{l=1}^{T_n-1} \left(\sum_{l'=l}^{T_n-1} u_{T_n,l'} \tilde{p}_n^{l'-l}\right)^2\right)$$

Next, we recall $Y_0 = 1$ by definition. Additionally, we have

$$u_{T_n,0} + \tilde{p}_n^* \sum_{l=1}^{T_n-1} u_{T_n,l} \tilde{p}_n^{l-1} = \sum_{t=0}^{T_n-1} (1-\eta)^t + \tilde{p}_n^* \sum_{l=1}^{T_n-1} \sum_{t=l}^{T_n-1} \binom{t}{l} \eta^l (1-\eta)^{t-l} \tilde{p}_n^{l-1}$$

$$= \sum_{t=0}^{T_n-1} \left(\sum_{l=1}^{l} \binom{t}{l} \eta^l (1-\eta)^{t-l} \tilde{p}_n^* \tilde{p}_n^{l-1} + (1-\eta)^t \right) = \frac{T_n}{\theta} \mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)],$$

where the first equality uses the definition of $u_{T_n,l}$, the second rearranges summations, and the third uses (E.26). Combining the previous two equations therefore yields

$$\mathbb{E}_n\Big[\prod_{l=0}^{T_n-1} \exp(\lambda u_{T_n,l} Y_l)\Big] \leq \exp\left(\lambda \frac{T_n}{\theta} \mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] + \frac{\lambda^2}{8} \sum_{l=1}^{T_n-1} \left(\sum_{l'=l}^{T_n-1} u_{T_n,l'} \tilde{p}_n^{l'-l} \right)^2 \right).$$

Hence, recalling that $\lambda = \tilde{\lambda}\theta/T_n$, and substituting into (E.55), we have shown

$$\mathbb{P}_n(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon) \leq \exp\left(-\tilde{\lambda}\varepsilon + \tilde{\lambda}\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] + \frac{\tilde{\lambda}^2\theta^2}{8T_n^2} \sum_{l=1}^{T_n-1} \left(\sum_{l'=l}^{T_n-1} u_{T_n,l'} \tilde{p}_n^{l'-l} \right)^2 \right). \quad \text{(E.63)}$$

Clearly, this inequality still holds if we multiply both sides by $1(\Omega_{n,2})$. Additionally, by (A3), $\tilde{p}_n(\omega) < p_n + \delta_n$ for $\omega \in \Omega_{n,2}$, where $p_n \to p$ and $\delta_n \to 0$; since we additionally assume $p < 1$ in the statement of the lemma, we conclude $\tilde{p}_n(\omega) < p_n + \delta_n < 1$ for $\omega \in \Omega_{n,2}$ and n sufficiently large. For such n, we can therefore write

$$\begin{aligned}
&\mathbb{P}_n(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon) 1(\Omega_{n,2}) \\
&\quad \leq \exp\left(-\tilde{\lambda}\varepsilon + \tilde{\lambda}\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] + \frac{\tilde{\lambda}^2\theta^2}{8T_n^2} \sum_{l=1}^{T_n-1} \left(\sum_{l'=l}^{T_n-1} u_{T_n,l'} (p_n+\delta_n)^{l'-l} \right)^2 \right) 1(\Omega_{n,2}) \\
&\quad \leq \exp\left(-\tilde{\lambda}\varepsilon + \tilde{\lambda}\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] + \frac{\tilde{\lambda}^2\theta^2}{8T_n\eta^2(1-(p_n+\delta_n))^2} \right) 1(\Omega_{n,2}),
\end{aligned}$$

where the second inequality uses Lemma E.12 from Appendix E.2.4. Additionally, since $p_n \to p < 1$, we can use the argument leading to (E.29) to obtain $\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)](\omega) < c/T_n$ (for some c independent of n) whenever $\omega \in \Omega_{n,2}$ and n is sufficiently large. For such n, we obtain

$$\mathbb{P}_n(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon) 1(\Omega_{n,2}) \leq \exp\left(-\tilde{\lambda}\varepsilon + \frac{\tilde{\lambda}c}{T_n} + \frac{\tilde{\lambda}^2\theta^2}{8T_n\eta^2(1-(p_n+\delta_n))^2} \right) 1(\Omega_{n,2}), \quad \text{(E.64)}$$

Now since $\tilde{\lambda} > 0$ was arbitrary, we can choose $\tilde{\lambda} = 4T_n\varepsilon\eta^2(1-(p_n+\delta_n))^2/\theta^2$. Upon substituting into the exponent in the previous equation, this exponent becomes

$$\begin{aligned}
&-\tilde{\lambda}\varepsilon + \frac{\tilde{\lambda}^2}{8T_n\eta^2(1-(p_n+\delta_n))^2} + \frac{\tilde{\lambda}c}{T_n} \\
&\quad = -2T_n\varepsilon^2\eta^2(1-p_n)^2/\theta^2 + 2T_n\varepsilon^2\eta^2\delta_n(2(1-p_n)-\delta_n)/\theta^2 + 4c\varepsilon\eta^2(1-(p_n+\delta_n))^2/\theta^2 \\
&\quad \leq -2T_n\varepsilon^2\eta^2(1-p_n)^2/\theta^2 + 4T_n\varepsilon^2\eta^2\delta_n/\theta^2 + 4c\varepsilon\eta^2/\theta^2, \qquad \text{(E.65)}
\end{aligned}$$

where the inequality simply uses $p_n, \delta_n > 0$ and $p_n + \delta_n \in (0,1)$ (for large n). Now note that since $p_n \to p$, we have (for example) $(1-p_n)^2 > (1-p)^2/2$ for n sufficiently large. Additionally, since $\delta_n = o(1/T_n)$, we have (for example) $T_n\delta_n < c/\varepsilon$ for n sufficiently large. Combining these observations, we can upper bound (E.65) as

$$-2\varepsilon^2 T_n \eta^2 (1-p_n)^2/\theta^2 + 4\eta^2\varepsilon^2 T_n\delta_n/\theta^2 + 4\eta^2\varepsilon c/\theta^2 \leq -(\varepsilon\eta(1-p))^2 T_n/\theta^2 + 8c\varepsilon\eta^2/\theta^2.$$

Hence, substituting into (E.64) gives

$$\mathbb{P}_n(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon)1(\Omega_{n,2}) \leq \exp(8c\varepsilon\eta^2/\theta^2)\exp(-(\varepsilon\eta(1-p)/\theta)^2 T_n)1(\Omega_{n,2}). \tag{E.66}$$

Finally, we write

$$\begin{aligned}\mathbb{P}(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon) &= \mathbb{E}[\mathbb{P}_n(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon)1(\Omega_{n,2}) + \mathbb{P}_n(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] > \varepsilon)1(\Omega_{n,2}^C)] \\ &\leq O\left(e^{-(\varepsilon\eta(1-p)/\theta)^2 T_n}\right) + \mathbb{P}(\Omega_{n,2}^C) = O\left(e^{-(\varepsilon\eta(1-p)/\theta)^2\mu\log n} + n^{-\kappa}\right),\end{aligned}$$

where the inequality uses (E.66) and upper bounds a probability by 1, and the second equality uses the assumptions in the statement of the lemma.

E.2.3.3 Where the proof fails in the case $p_n \to 1$

As shown in Appendix E.1.4.1, extending Theorem 6.2 to the case $p_n \to 1$ amounts to showing that for some $\gamma' > 0$,

$$\mathbb{P}(|\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] - L(p_n)| > \varepsilon) = O\left(n^{-\gamma'}\right), \tag{E.67}$$

where $L(p_n)$ is the appropriate limit from (E.19). Here we show (roughly) why the approach from the preceding proof fails to establish (E.67) in the case $p_n \to 1$. To begin, we note we first used the assumption $p_n \to p < 1$ following (E.63). Hence, in the case $p_n \to 1$, we can still follow the approach leading to (E.63) to obtain the (one-sided) bound

$$\begin{aligned}&\mathbb{P}(\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}] - L(p_n) > \varepsilon)1(\Omega_{n,2}) \leq \exp(-\tilde{\lambda}(\varepsilon + L(p_n))\mathbb{E}\exp(\tilde{\lambda}\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}])1(\Omega_{n,2}) \\ &\quad\leq \exp\left(-\tilde{\lambda}\varepsilon + \tilde{\lambda}\left(-L(p_n) + \mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)]\right) + \frac{\tilde{\lambda}^2\theta^2}{8T_n^2}\sum_{l=1}^{T_n-1}\left(\sum_{l'=l}^{T_n-1} u_{T_n,l'}\tilde{p}_n^{l'-l}\right)^2\right)1(\Omega_{n,2}) \\ &\quad\approx \exp\left(-\tilde{\lambda}\varepsilon + \frac{\tilde{\lambda}^2\theta^2}{8T_n^2}\sum_{l=1}^{T_n-1}\left(\sum_{l'=l}^{T_n-1} u_{T_n,l'}\tilde{p}_n^{l'-l}\right)^2\right)1(\Omega_{n,2}),\end{aligned} \tag{E.68}$$

where the approximate equality uses $\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] \approx L(p_n)$ on $\Omega_{n,2}$ by Lemma E.5. We next note

$$\sum_{l=1}^{T_n-1}\left(\sum_{l'=l}^{T_n-1} u_{T_n,l'}\tilde{p}_n^{l'-l}\right)^2 \geq \left(\sum_{l'=1}^{T_n-1} u_{T_n,l'}\tilde{p}_n^{l'-1}\right)^2 = \left(\sum_{l'=1}^{T_n-1}\left(\sum_{t=l'}^{T_n-1}\binom{t}{l'}\eta^{l'}(1-\eta)^{t-l'}\right)\tilde{p}_n^{l'-1}\right)^2$$

$$= (\tilde{p}_n^*)^{-2} \left(\sum_{t=1}^{T_n-1} \sum_{l'=1}^{t} \binom{t}{l'} \eta^{l'} (1-\eta)^{t-l'} \tilde{p}_n^* \tilde{p}_n^{l'-1} \right)^2$$
$$= (\tilde{p}_n^*)^{-2} \left(\frac{T_n}{\theta} \mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] - \frac{1-(1-\eta)^{T_n}}{\eta} \right)^2,$$

where the inequality discards nonnegative terms, the first equality is by definition of $u_{T_n,l'}$, the second rearranges summations and multiplies/divides by $(\tilde{p}_n^*)^2$, and the third uses (E.26). Hence, we have shown (E.68) is (roughly) lower bounded by

$$\exp\left(-\tilde{\lambda}\varepsilon + \frac{\tilde{\lambda}^2}{8} \left(\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] - \frac{\theta(1-(1-\eta)^{T_n})}{T_n \eta} \right)^2 \right) 1(\Omega_{n,2}),$$

where we have also used $\tilde{p}_n^* \approx 1$ for large n on $\Omega_{n,2}$ when $p_n \to 1$ by (A3). Now we consider three cases for the exponent in the previous expression:

- $T_n(1-p_n) \to 0$: Here Lemma E.5 states $\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] \approx \theta$ for large n on $\Omega_{n,2}$; for such n, the exponent is roughly

$$-\tilde{\lambda}\varepsilon + \frac{\tilde{\lambda}^2\theta^2}{8} \left(1 - \frac{\theta(1-(1-\eta)^{T_n})}{T_n\eta} \right)^2 \geq -\tilde{\lambda}\varepsilon + \frac{\tilde{\lambda}^2\theta^2}{16} = -\frac{4\varepsilon^2}{\theta^2},$$

  where the inequality holds for large n (so that $\theta(1-(1-\eta)^{T_n})/(T_n\eta) < 1 - 1/\sqrt{2}$, which holds since $T_n \to \infty$) and the equality holds by choosing the minimizing $\tilde{\lambda}$ (namely, $\tilde{\lambda} = 8\varepsilon/\theta^2$). Since this lower bound is constant in n, (E.68) does not decay as n grows.
- $T_n(1-p_n) \to c \in (0,\infty)$: Here Lemma E.5 states $\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] \approx \theta(1-e^{-c\eta})/(c\eta)$ for large n on $\Omega_{n,2}$. An argument similar to the previous case shows (E.68) does not decay.
- $T_n(1-p_n) \to \infty$ with $p_n \to 1$: Here we consider an example to show (E.68) does not decay sufficiently quickly for the general case: assume $T_n = \bar{c}\log n$ for some constant $\bar{c}$ that satisfies the theorem assumptions and we set $p_n = 1 - (\log n)^{-0.9}$. Then since $\delta_n = o((\log n)^{-1})$ per (A3), we have e.g. $1 - p_n + \delta_n < (1-p_n)/2$ for large n. Hence,

$$\begin{aligned}\mathbb{E}_n[\hat{\vartheta}_{T_n}(\phi)] &\gtrsim \frac{\theta(1-(1-\eta(1-p_n+\delta_n))^{T_n})}{\eta T_n(1-p_n+\delta_n)} \\ &> \frac{\theta(1-(1-(\eta/2)(\log n)^{-0.9})^{\bar{c}\log n})}{(\bar{c}\eta/2)(\log n)^{0.1}} > \frac{\tilde{c}}{(\log n)^{0.1}},\end{aligned}$$

  where the first inequality holds by (E.28) in Appendix E.2.2.1 (where γ_1, γ_2 are arbitrarily small, hence the approximate inequality), the second holds for our chosen T_n, p_n, δ_n, and the third holds for some constant $\tilde{c}$ and for large n. Hence, the exponent is (roughly) lower bounded by

$$-\tilde{\lambda}\varepsilon + \frac{\tilde{\lambda}^2}{8} \frac{\tilde{c}^2}{(\log n)^{0.2}} = -\frac{2\varepsilon^2}{\tilde{c}^2} (\log n)^{0.2},$$

  where the equality holds for the minimizer $\tilde{\lambda} = (4\varepsilon/\tilde{c}^2)(\log n)^{0.2}$. From here it follows

that (E.68) cannot be $O(n^{-\gamma'})$: if it is, then for all large n and for some constant $\tilde{C}$,

$$\exp\left(-\frac{2\varepsilon^2}{\tilde{c}^2}(\log n)^{0.2}\right) < \tilde{C}n^{-\gamma'} \Rightarrow \exp\left(-\frac{2\varepsilon^2}{\tilde{c}^2}(\log n)^{0.2} + \gamma' \log n\right) < \tilde{C}.$$

The final inequality is a contradiction, since $-(2\varepsilon^2/\tilde{c}^2)(\log n)^{0.2} + \gamma' \log n \to \infty$.

E.2.4 Auxiliary results

In this appendix, we collect several auxiliary results used in other proofs. (These results are either cited from other sources, or their proofs are tedious but elementary, so we collect them here to avoid cluttering other parts of our analysis.)

Lemma E.11. For $T_n \to \infty$, $p_n \to 1$, and $\delta_n \to 0$ s.t. $\delta_n = o(1/T_n)$, we have

$$\frac{1-(1-\eta(1-p_n-\delta_n))^{T_n}}{\eta T_n(1-p_n-\delta_n)} \xrightarrow[n\to\infty]{} \begin{cases} 1, & T_n(1-p_n) \xrightarrow[n\to\infty]{} 0 \\ (1-e^{-c\eta})/(c\eta), & T_n(1-p_n) \xrightarrow[n\to\infty]{} c \in (0,\infty) \\ 0, & T_n(1-p_n) \xrightarrow[n\to\infty]{} \infty \end{cases} \tag{E.69}$$

$$\frac{1-(1-\eta(1-p_n+\delta_n))^{T_n}}{\eta T_n(1-p_n+\delta_n)} \xrightarrow[n\to\infty]{} \begin{cases} 1, & T_n(1-p_n) \xrightarrow[n\to\infty]{} 0 \\ (1-e^{-c\eta})/(c\eta), & T_n(1-p_n) \xrightarrow[n\to\infty]{} c \in (0,\infty) \\ 0, & T_n(1-p_n) \xrightarrow[n\to\infty]{} \infty \end{cases} \tag{E.70}$$

Proof. We consider the three cases of (E.69) in turn; the proof of (E.70) follows the same approach. First, suppose $\lim_{n\to\infty} T_n(1-p_n) = \infty$. Then since $T_n\delta_n \to 0$ and $T_n(1-p_n) \to \infty$, we have $T_n\delta_n < 1 < T_n(1-p_n)$ for sufficiently large n, which implies $(1-p_n-\delta_n) > 0$ for such n. Clearly, we also have $(1-p_n-\delta_n) < 1$ for all n. Taken together, it follows that $1-(1-\eta(1-p_n-\delta_n))^{T_n} \in (0,1)$ for n large. For such n, we can then write

$$0 < \frac{1-(1-\eta(1-p_n-\delta_n))^{T_n}}{\eta T_n(1-p_n-\delta_n)} < \frac{1}{\eta T_n(1-p_n-\delta_n)},$$

where we used $(1-p_n-\delta_n) > 0$ in the denominator. Now since $T_n(1-p_n) \to \infty$ and $T_n\delta_n \to 0$, $T_n(1-p_n-\delta_n) \to \infty$, so taking $n \to \infty$ in the above inequality gives the result.

Next, suppose $\lim_{n\to\infty} T_n(1-p_n) = c \in (0,\infty)$. Since $\eta T_n(1-p_n-\delta_n) \to \eta c$ by $T_n(1-p_n) \to c$ and $T_n\delta_n \to 0$, it suffices to show $(1-\eta(1-p_n-\delta_n))^{T_n} \to e^{-\eta c}$ as $n \to \infty$. First, since $T_n(1-p_n) \to c$, $\forall\ \varepsilon_1 > 0\ \exists\ N_1$ s.t. $c-\varepsilon_1 < T_n(1-p_n) < c+\varepsilon_1\ \forall\ n \geq N_1$. Further, since $T_n\delta_n \to 0$, $\forall\ \varepsilon_2 > 0\ \exists\ N_2$ s.t. $-\varepsilon_2 < T_n\delta_n < \varepsilon_2\ \forall\ n \geq N_2$. Hence, $\forall\ n \geq \max\{N_1, N_2\}$,

$$\left(1-\frac{\eta(c+\varepsilon_1+\varepsilon_2)}{T_n}\right)^{T_n} < (1-\eta(1-p_n-\delta_n))^{T_n} < \left(1-\frac{\eta(c-\varepsilon_1-\varepsilon_2)}{T_n}\right)^{T_n}.$$

Moreover, since $(1-x/m)^m \xrightarrow[m\to\infty]{} e^{-x}$, $\forall\ \varepsilon_3 > 0\ \exists\ N_3$ s.t. $\forall\ n \geq N_3$,

$$e^{-\eta(c+\varepsilon_1+\varepsilon_2)} - \varepsilon_3 < \left(1-\frac{\eta(c+\varepsilon_1+\varepsilon_2)}{T_n}\right)^{T_n}, \quad \left(1-\frac{\eta(c-\varepsilon_1)}{T_n}\right)^{T_n} < e^{-\eta(c-\varepsilon_1-\varepsilon_2)} + \varepsilon_3.$$

Combining these arguments, we obtain $\forall\ n \geq \max\{N_1, N_2, N_3\}$

$$e^{-\eta(c+\varepsilon_1+\varepsilon_2)} - \varepsilon_3 < (1-\eta(1-p_n-\delta_n))^{T_n} < e^{-\eta(c-\varepsilon_1-\varepsilon_2)} + \varepsilon_3.$$

Since both bounds converge to $e^{-\eta c}$ as $\varepsilon_1, \varepsilon_2, \varepsilon_3 \to 0$, $(1-\eta(1-p_n-\delta_n))^{T_n} \to e^{-\eta c}$ follows.

Finally, suppose $\lim_{n\to\infty} T_n(1-p_n) = 0$. First, we observe

$$\frac{1-(1-\eta(1-p_n-\delta_n))^{T_n}}{\eta T_n(1-p_n-\delta_n)} = \frac{1}{T_n}\sum_{t=0}^{T_n-1}(1-\eta(1-p_n-\delta_n))^t \leq 1, \tag{E.71}$$

where the inequality holds for n s.t. $(1-p_n-\delta_n) > 0$ (which indeed occurs for large n; see proof of $T_n(1-p_n) \to \infty$ case), since then the sum is over T_n terms, each upper bounded by 1. On the other hand, we can use the binomial theorem to write

$$\begin{aligned}\frac{1-(1-\eta(1-p_n-\delta_n))^{T_n}}{\eta T_n(1-p_n-\delta_n)} &= \frac{1-\sum_{t=0}^{T_n}\binom{T_n}{t}(-\eta(1-p_n-\delta_n))^t}{\eta T_n(1-p_n-\delta_n)} \\ &= 1 - \sum_{t=2}^{T_n}\frac{(T_n-1)\cdots(T_n-t+1)(-1)^t(\eta(1-p_n-\delta_n))^{t-1}}{t!}.\end{aligned} \tag{E.72}$$

Next, we observe (assuming $(1-p_n-\delta_n) > 0$) as above)

$$\begin{aligned}&\sum_{t=2}^{T_n}\frac{(T_n-1)\cdots(T_n-t+1)(-1)^t(\eta(1-p_n-\delta_n))^{t-1}}{t!} \\ &< \sum_{t=2}^{T_n}\frac{(T_n-1)\cdots(T_n-t+1)(\eta(1-p_n-\delta_n))^{t-1}}{t!} < \sum_{t=2}^{T_n}\frac{(T_n(1-p_n-\delta_n))^{t-1}}{(t-2)!} \\ &= T_n(1-p_n-\delta_n)\sum_{t=0}^{T_n-2}\frac{(T_n(1-p_n-\delta_n))^t}{t!} < T_n(1-p_n-\delta_n)e^{T_n(1-p_n-\delta_n)},\end{aligned} \tag{E.73}$$

where the first inequality replaces negative terms with positive ones; the second inequality uses $\eta < 1$, $(t-2)! < t!$, and $(T_n - j) < T_n$ for $j > 0$; and the third inequality replaces the upper limit of the summation with infinity. Hence, (E.71), (E.72), and (E.73) yield

$$\begin{aligned}&1 \geq \frac{1-(1-\eta(1-p_n-\delta_n))^{T_n}}{\eta T_n(1-p_n-\delta_n)} > 1 - T_n(1-p_n-\delta_n)e^{T_n(1-p_n-\delta_n)} \\ &\Rightarrow 1 \geq \lim_{n\to\infty}\frac{1-(1-\eta(1-p_n-\delta_n))^{T_n}}{\eta T_n(1-p_n-\delta_n)} \geq 1 - \lim_{n\to\infty} T_n(1-p_n-\delta_n)e^{T_n(1-p_n-\delta_n)} = 1,\end{aligned}$$

where the final equality holds since $T_n(1-p_n), T_n\delta_n \to 0$ by assumption. □

Lemma E.12. Let $u_{T_n,l} = \sum_{t=l}^{T_n-1}\binom{t}{l}\eta^l(1-\eta)^{t-l}$. Then for any $x \in (0,1)$,

$$\sum_{l=1}^{T_n-1}\left(\sum_{l'=l}^{T_n-1} u_{T_n,l'}x^{l'-l}\right)^2 \leq \frac{T_n}{\eta^2(1-x)^2}.$$

Proof. For $l \in \mathbb{N}_0$, define $w_l = \sum_{l'=l}^{T_n-1} u_{T_n,l'} x^{l'-l}$. Then

$$w_l = u_{T_n,l} + x \sum_{l'=l+1}^{T_n-1} u_{T_n,l'} x^{l'-(l+1)} = u_{T_n,l} + x w_{l+1}.$$

Assuming temporarily that $u_{T_n,l'} \geq u_{T_n,l''}$ whenever $l' \leq l''$ (which we will return to prove),

$$w_{l+1} \leq u_{T_n,l} \sum_{l'=l+1}^{T_n-1} x^{l'-(l+1)} = u_{T_n,l} \sum_{l'=0}^{T_n-l-2} x^{l'} \leq u_{T_n,l} \sum_{l'=0}^{\infty} x^{l'} = \frac{u_{T_n,l}}{1-x}.$$

Hence, using the previous two equations, we obtain $w_{l+1} - w_l = (1-x)w_{l+1} - u_{T_n-l} \leq 0$, i.e. the sequence $\{w_l\}$ decreases in l. It is also clearly nonnegative. Therefore,

$$\sum_{l=1}^{T_n-1} \left(\sum_{l'=l}^{T_n-1} u_{T_n,l'} x^{l'-l} \right)^2 = \sum_{l=1}^{T_n-1} w_l^2 \leq T_n w_0^2.$$

To further bound the right hand side, we note

$$\begin{aligned} w_0 &= \sum_{l'=0}^{T_n-1} \left(\sum_{t=l'}^{T_n-1} \binom{t}{l'} \eta^{l'} (1-\eta)^{t-l'} \right) x^{l'} = \sum_{t=0}^{T_n-1} \sum_{l'=0}^{t} \binom{t}{l'} (\eta x)^{l'} (1-\eta)^{t-l'} \\ &= \sum_{t=0}^{T_n-1} (\eta x + (1-\eta))^t = \sum_{t=0}^{T_n-1} (1-\eta(1-x))^t \leq \sum_{t=0}^{\infty} (1-\eta(1-x))^t = \frac{1}{\eta(1-x)}, \end{aligned}$$

where the first line uses the definition of $u_{T_n,l'}$ and rearranges summations, and the second line involves simple calculations. The previous two inequalities then imply the lemma.

We return to prove $u_{T_n,l'} \geq u_{T_n,l''}$ for $l' \leq l''$. First, we claim that $\forall\, t^* \in \mathbb{N}, l \in \{1, \ldots, t^*\}$,

$$\sum_{t=l}^{t^*} \binom{t}{l} \eta^l (1-\eta)^{t-l} - \sum_{t=l+1}^{t^*+1} \binom{t}{l+1} \eta^{l+1} (1-\eta)^{t-(l+1)} = \binom{t^*+1}{l+1} \eta^l (1-\eta)^{t^*+1-l}. \quad \text{(E.74)}$$

We prove (E.74) by induction on t^*. First, when $t^* = 1$, the only case to prove is $l = 1$; when $t^* = l = 1$, it is immediate that both sides of (E.74) equal $\eta(1-\eta)$. Next, assume (E.74) holds for $t^* - 1$. If $l = t^*$, both sides of (E.74) equal $\eta^{t^*}(1-\eta)$. If $l \in \{1, \ldots, t^*-1\}$, we write

$$\begin{aligned} &\sum_{t=l}^{t^*} \binom{t}{l} \eta^l (1-\eta)^{t-l} - \sum_{t=l+1}^{t^*+1} \binom{t}{l+1} \eta^{l+1} (1-\eta)^{t-(l+1)} \\ &\quad = \left(\sum_{t=l}^{t^*-1} \binom{t}{l} \eta^l (1-\eta)^{t-l} - \sum_{t=l+1}^{t^*} \binom{t}{l+1} \eta^{l+1} (1-\eta)^{t-(l+1)} \right) \\ &\qquad + \binom{t^*}{l} \eta^l (1-\eta)^{t^*-l} - \binom{t^*+1}{l+1} \eta^{l+1} (1-\eta)^{t^*+1-(l+1)} \end{aligned}$$

$$= \left(\binom{t^*}{l+1}\eta^l(1-\eta)^{t^*-l}\right) + \binom{t^*}{l}\eta^l(1-\eta)^{t^*-l} - \binom{t^*+1}{l+1}\eta^{l+1}(1-\eta)^{t^*-l}$$
$$= \eta^l(1-\eta)^{t^*-l}\left(\binom{t^*}{l+1} + \binom{t^*}{l} - \eta\binom{t^*+1}{l+1}\right)$$
$$= \eta^l(1-\eta)^{t^*-l}\left(\binom{t^*+1}{l+1} - \eta\binom{t^*+1}{l+1}\right) = \binom{t^*+1}{l+1}\eta^l(1-\eta)^{t^*+1-l},$$

where the first equality simply writes the final summands separately, the second uses the inductive hypothesis on the term in parentheses, the third is immediate, the fourth uses Pascal's rule ($[t^*+1]$ has $\binom{t^*+1}{l+1}$ subsets of cardinality $l+1$; $\binom{t^*}{l}$ that contain 1 and $\binom{t^*}{l+1}$ that do not contain 1), and the fifth is immediate. This establishes (E.74). We then write

$$\begin{aligned}
u_{T_n,l'} - u_{T_n,l'+1} &= \sum_{t=l'}^{T_n-1}\binom{t}{l'}\eta^{l'}(1-\eta)^{t-l'} - \sum_{t=l'+1}^{T_n-1}\binom{t}{l'+1}\eta^{l'+1}(1-\eta)^{t-(l'+1)} \\
&= \sum_{t=l'}^{T_n-1}\binom{t}{l'}\eta^{l'}(1-\eta)^{t-l'} - \sum_{t=l'+1}^{T_n}\binom{t}{l'+1}\eta^{l'+1}(1-\eta)^{t-(l'+1)} \\
&\quad + \binom{T_n}{l'+1}\eta^{l'+1}(1-\eta)^{T_n-(l'+1)} \\
&= \binom{T_n}{l'+1}\eta^{l'}(1-\eta)^{T_n-l'} + \binom{T_n}{l'+1}\eta^{l'+1}(1-\eta)^{T_n-(l'+1)} \\
&= \binom{T_n}{l'+1}\eta^{l'}(1-\eta)^{T_n-(l'+1)} \geq 0,
\end{aligned}$$

where the first equality holds by definition of $u_{T_n,l'}$, the second adds and subtracts a term, and the third uses (E.74). Thus, $u_{T_n,l'} \geq u_{T_n,l'+1}$; iterating gives $u_{T_n,l'} \geq u_{T_n,l''}$ for $l' \leq l''$. □

Lemma E.13. Let Z be a random variable satisfying $\mathbb{E}Z = 0$ and $Z \in [a,b]$ *a.s.*, and let $\lambda > 0$. Then $\mathbb{E}e^{\lambda Z} \leq e^{\lambda^2(b-a)^2/8}$.

Proof. See e.g. [124, Lemma 5.1]. □

E.3 Proof of Theorem 6.4

The proof relies on three lemmas proved at the end of this appendix. Also, throughout the proof, we use $\tilde{\mathbb{P}}_n$ and $\tilde{\mathbb{E}}_n$, respectively, to denote probability and expectation, respectively, conditioned on $\{d_{out}(i), d^A_{in}(i)\}_{i\in[n]}]$ (i.e. the degrees observed by the adversary).

In the first lemma, we solve the relaxed problem. The proof is standard and primarily amounts to verifying the Karush-Kuhn-Tucker (KKT) conditions (see e.g. [128, Section 5.5.3]).

Lemma E.14. The solution of the relaxed problem (6.17) is, almost surely,

$$d^{rel}_n(i) = d^A_{in}(i)\left(\frac{\sqrt{r(i)}}{h^*} - 1\right)_+ \quad \forall\, i \in [n],$$

where $x_+ = x$ for $x > 0$ and $x_+ = 0$ for $x \leq 0$. Furthermore, $\sum_{i=1}^n d_n^{rel}(i) = b_n$.

Proof. See Appendix E.3.1. □

The next lemma shows that also the randomized scheme objective is (in expectation) close to the relaxed solution objective.

Lemma E.15. The following inequalities hold almost surely:

$$\tilde{p}_n\left(d_n^{rel}\right) \leq \tilde{p}_n\left(d_n^{opt}\right) \leq \tilde{\mathbb{E}}_n \tilde{p}_n\left(d_n^{rand}\right) < \frac{1}{2}\left(1 + \tilde{p}_n\left(d_n^{rel}\right)\right) \leq \frac{1}{2}\left(1 + \tilde{p}_n\left(d_n^{opt}\right)\right).$$

Proof. See Appendix E.3.2. □

The third and final lemma provides a tail bound for this randomized scheme objective. The proof shows that an affine transform of this objective is a *self-bounding* function of independent random variables [129, Section 3.3] and uses an existing result for such functions.

Lemma E.16. For any $\delta > 0$, $\exists\ c_\delta > 0$ independent of n such that, almost surely,

$$\tilde{\mathbb{P}}_n\left(\tilde{p}_n\left(d_n^{rand}\right) \geq \frac{1 + \delta + \tilde{p}_n(d_n^{opt})}{2 + \delta}\right) \leq \exp\left(-\frac{c_\delta m_n (1 - \tilde{p}_n(d_n^{opt}))}{\max_{j\in[n]} r(j)}\right).$$

Proof. See Appendix E.3.3 □

With Lemma E.16 in place, we can prove the theorem. First, by (6.20), we can find a sequence $\{x_n\}_{n\in\mathbb{N}} \subset [0, \infty)$ satisfying $x_n \to \infty$ and $\mathbb{P}(\mathcal{E}_{x_n}) \to 1$, where $\mathcal{E}_{x_n}$ is the event

$$\mathcal{E}_{x_n} = \left\{\frac{m_n(1 - \tilde{p}_n(d_n^{opt}))}{\max_{j\in[n]} r(j)} \geq x_n\right\}.$$

Thus, by the law of total expectation and Lemma E.16,

$$\begin{aligned}
\mathbb{P}\left(\tilde{p}_n\left(d_n^{rand}\right) \geq \frac{1 + \delta + \tilde{p}_n(d_n^{opt})}{2 + \delta}\right) &= \mathbb{E}\left[\tilde{\mathbb{P}}_n\left(\tilde{p}_n\left(d_n^{rand}\right) \geq \frac{1 + \delta + \tilde{p}_n(d_n^{opt})}{2 + \delta}\right)\right] \\
&\leq \mathbb{E}\left[\tilde{\mathbb{P}}_n\left(\tilde{p}_n\left(d_n^{rand}\right) \geq \frac{1 + \delta + \tilde{p}_n(d_n^{opt})}{2 + \delta}\right)\middle| \mathcal{E}_{x_n}\right] + \mathbb{P}\left(\mathcal{E}_{x_n}^C\right) \\
&\leq \exp\left(-c_\delta x_n\right) + \mathbb{P}\left(\mathcal{E}_{x_n}^C\right) \xrightarrow[n\to\infty]{} 0.
\end{aligned}$$

E.3.1 Proof of Lemma E.14

First note strict convexity of $y \mapsto 1/y$ for $y \in \mathbb{R}$ implies strict convexity of $\tilde{p}_n$, i.e. for any $d \neq d' \in \mathbb{R}_+^n$ and $\rho \in (0, 1)$, we have

$$\begin{aligned}
\tilde{p}_n(\rho d + (1-\rho)d') &= \sum_{i=1}^n \frac{1}{\rho(d_{in}^A(i) + d(i)) + (1-\rho)(d_{in}^A(i) + d'(i))} \frac{d_{in}^A(i) d_{out}(i)}{m_n} \\
&< \sum_{i=1}^n \left(\frac{\rho}{d_{in}^A(i) + d(i)} + \frac{1-\rho}{d_{in}^A(i) + d'(i)}\right) \frac{d_{in}^A(i) d_{out}(i)}{m_n} = \rho \tilde{p}_n(d) + (1-\rho)\tilde{p}_n(d').
\end{aligned}$$

Also note we can rewrite the relaxed problem (6.17) as

$$\min_{d\in\mathbb{R}^n} \tilde{p}_n(d) \ s.t. \ g(d) \leq 0, g_i(d) \leq 0 \ \forall \ i \in [n], \tag{E.75}$$

where $g(d) = \sum_{i=1}^n d(i) - b_n, g_i(d) = -d(i)$. Given $\lambda, \lambda_i \geq 0$, we also define the Lagrangian

$$L(d, \lambda, \lambda_1, \ldots, \lambda_n) = \tilde{p}_n(d) + \lambda g(d) + \sum_{i=1}^n \lambda_i g_i(d).$$

Finally, we set $\lambda^* = (h^*)^2/m_n, \lambda_i^* = ((h^*)^2 - r(i))_+/m_n$ (clearly, $\lambda^*, \lambda_i^* \geq 0$). Now to prove the theorem, it suffices to establish the following (see e.g. [128, Section 5.5.3]):

1. $g(d_n^{rel}), g_1(d_n^{rel}), \ldots, g_n(d_n^{rel}) \leq 0 \ \forall \ i \in [n]$, i.e. d_n^{rel} is a feasible point of (E.75).
2. $\nabla L(d_n^{rel}, \lambda^*, \lambda_1^*, \ldots, \lambda_n^*) = 0$, i.e. the first-order condition is satisfied.
3. $\lambda^* g(d_n^{rel}) = \lambda_1^* g_1(d_n^{rel}) = \cdots = \lambda_n^* g_n(d_n^{rel}) = 0$, i.e. complementary slackness holds.

We proceed to the proofs of these three statements.

1. Clearly, $g_i(d_n^{rel}) \leq 0 \ \forall \ i \in [n]$. To show $g(d_n^{rel}) \leq 0$, we claim (and will return to prove) that h^* is a fixed point of h, i.e. $h^* = h(h^*)$. Assuming this claim holds, we have

$$\begin{aligned} g\left(d_n^{rel}\right) &= \frac{1}{h^*} \sum_{i\in[n]:r(i)\geq(h^*)^2} d_{in}^A(i)\sqrt{r(i)} - \sum_{i\in[n]:r(i)\geq(h^*)^2} d_{in}^A(i) - b_n \\ &= \frac{1}{h(h^*)} \sum_{i\in[n]:r(i)\geq(h^*)^2} \sqrt{d_{out}(i)d_{in}^A(i)} - \sum_{i\in[n]:r(i)\geq(h^*)^2} d_{in}^A(i) - b_n = 0, \end{aligned} \tag{E.76}$$

   where the last two equalities use the fixed point claim and the definition of h.
2. If $i \in [n]$ satisfies $r(i) > (h^*)^2$, then $d_n^{rel}(i) = -d_{in}^A(i) + d_{in}^A(i)\sqrt{r(i)}/h^*, \lambda_i^* = 0$, and

$$\begin{aligned} \frac{\partial L}{\partial d(i)}\left(d_n^{rel}, \lambda^*, \lambda_1^*, \ldots, \lambda_n^*\right) &= -\frac{d_{out}(i)d_{in}^A(i)}{m_n}\frac{1}{(d_{in}^A(i) + d_n^{rel}(i))^2} + \lambda^* \\ = -\frac{d_{out}(i)d_{in}^A(i)}{m_n(d_{in}^A(i)\sqrt{r(i)}/h^*)^2} + \frac{(h^*)^2}{m_n} &= -\frac{d_{out}(i)d_{in}^A(i)}{m_n\left(\sqrt{d_{out}(i)d_{in}^A(i)}/h^*\right)^2} + \frac{(h^*)^2}{m_n} = 0. \end{aligned}$$

   Next, let $i \in [n]$ satisfy $r(i) \leq (h^*)^2$, so that $d_n^{rel}(i) = 0, \lambda_i = ((h^*)^2 - r(i))/m_n$. Then

$$\begin{aligned} \frac{\partial L}{\partial d(i)}\left(d_n^{rel}, \lambda^*, \lambda_1^*, \ldots, \lambda_n^*\right) &= -\frac{d_{out}(i)d_{in}^A(i)}{m_n}\frac{1}{(d_{in}^A(i))^2} + \lambda^* - \lambda_i^* \\ = -\frac{r(i)}{m_n} + \frac{(h^*)^2}{m_n} - \frac{(h^*)^2 - r(i)}{m_n} &= 0. \end{aligned}$$

3. For any $i \in [n]$, we have

$$\lambda_i^* g_i(d_n^{rel}) = -d_{in}^A(i)\left(\frac{(h^*)^2 - r(i)}{m_n}\right)_+\left(\frac{\sqrt{r(i)}}{h^*} - 1\right)_+.$$

Clearly, the first $(\cdot)_+$ term is zero if $r(i) > (h^*)^2$, the second is zero if $r(i) < (h^*)^2$, and both are zero if $r(i) = (h^*)^2$. Finally, $\lambda^* g(d_n^{rel}) = 0$ holds by (E.76).

We return to establish the fixed point claim. We in fact prove the slightly stronger result

$$h(x) \leq h(h(x)) \ \forall \ x \in \mathbb{R}_+. \tag{E.77}$$

The fixed point claim then follows, since $h^* \geq h(h^*)$ by definition and $h^* \leq h(h^*)$ by (E.77) with $x = x^*$, where x^* is a maximizer of h. Thus, it suffices to prove (E.77). Towards this end, fix $x \in \mathbb{R}_+$. We first assume $x \geq h(x)$ and will return to address the other case. For any $y, z \in \mathbb{R} \cup \{\infty\}$, we define

$$N(y,z) = \sum_{i \in [n]: r(i) \in [y^2, z^2)} \sqrt{d_{out}(i) d_{in}^A(i)}, \quad D(y,z) = \sum_{i \in [n]: r(i) \in [y^2, z^2)} d_{in}^A(i),$$

where by convention $N(y,z) = D(y,z) = 0$ if y, z are such that $\{i \in [n] : r(i) \in [y^2, z^2)\} = \emptyset$ (i.e. if the sums are over empty sets). Then by definition of h, N, and D, we have

$$h(h(x)) = \frac{N(h(x), \infty)}{b_n + D(h(x), \infty)} = \frac{N(x, \infty) + N(h(x), x)}{b_n + D(x, \infty) + D(h(x), x)}.$$

Again by definition of h, N, and D, and recalling $r(i) = d_{out}(i)/d_{in}^A(i)$, we also have

$$\begin{aligned} N(h(x), x) &= \sum_{i \in [n]: r(i) \in [h(x)^2, x^2)} \sqrt{r(i)} d_{in}^A(i) \geq h(x) \sum_{i \in [n]: r(i) \in [h(x)^2, x^2)} d_{in}^A(i) \\ &= h(x) D(h(x), x) = \frac{N(x, \infty)}{b_n + D(x, \infty)} D(h(x), x) \end{aligned}$$

Thus, combining the previous two equations, we obtain

$$h(h(x)) \geq \frac{N(x, \infty) + \frac{N(x,\infty)}{b_n + D(x,\infty)} D(h(x), x)}{b_n + D(x, \infty) + D(h(x), x)} = \frac{N(x, \infty)}{b_n + D(x, \infty)} = h(x).$$

If instead $x \leq h(x)$, we can use the same argument to obtain

$$\begin{aligned} h(h(x)) &= \frac{N(x, \infty) - N(x, h(x))}{b_n + D(x, \infty) - D(x, h(x))}, \quad N(x, h(x)) \leq \frac{N(x, \infty)}{b_n + D(x, \infty)} D(x, h(x)), \\ \Rightarrow h(h(x)) &\geq \frac{N(x, \infty) - \frac{N(x,\infty)}{b_n + D(x,\infty)} D(x, h(x))}{b_n + D(x, \infty) - D(x, h(x))} = \frac{N(x, \infty)}{b_n + D(x, \infty)} = h(x). \end{aligned}$$

E.3.2 Proof of Lemma E.15

The first and fourth inequalities are immediate, since d_n^{rel} is the solution of (6.17), d_n^{opt} is the solution of (6.15), and (6.17) enlarges the feasible set of (6.15). The second inequality is immediate by definition of d_n^{opt}. Thus, it only remains to prove the third inequality.

Towards this end, first recall that for each $i \in [n]$,

$$d_n^{rand}(i) = \sum_{j=1}^{b_n} 1(W_j = i), \quad \tilde{\mathbb{P}}_n(W_j = i) = \frac{d_n^{rel}(i)}{\sum_{k=1}^n d_n^{rel}(k)} = \frac{d_n^{rel}(i)}{b_n} \ \forall\ j \in [b_n], \tag{E.78}$$

where the second equality holds by Lemma E.14. Also note, by definition of m_n and d_n^{rand},

$$\begin{aligned}\tilde{\mathbb{E}}_n\left(1 - \tilde{p}_n\left(d_n^{rand}\right)\right) &= \sum_{i=1}^n \frac{d_{out}(i)}{m_n} \tilde{\mathbb{E}}_n \frac{d_n^{rand}(i)}{d_{in}^A(i) + d_n^{rand}(i)} \\ &= \sum_{i=1}^n \frac{d_{out}(i)}{m_n} \tilde{\mathbb{E}}_n \frac{\sum_{j=1}^{b_n} 1(W_j = i)}{d_{in}^A(i) + \sum_{k=1}^{b_n} 1(W_k = i)} = \sum_{i=1}^n \frac{d_{out}(i)}{m_n} \sum_{j=1}^{b_n} \tilde{\mathbb{E}}_n \frac{1(W_j = i)}{d_{in}^A(i) + \sum_{k=1}^{b_n} 1(W_k = i)}\end{aligned} \tag{E.79}$$

We can then bound the (i,j)-th summand in (E.79) as

$$\begin{aligned}&\tilde{\mathbb{E}}_n \frac{1(W_j = i)}{d_{in}^A(i) + \sum_{k=1}^{b_n} 1(W_k = i)} \\ &= \tilde{\mathbb{E}}_n \tilde{\mathbb{E}}_n \left[\frac{1(W_j = i)}{d_{in}^A(i) + 1(W_j = i) + \sum_{k=1, k\neq j}^{b_n} 1(W_k = i)} \middle| \{W_k\}_{k=1,k\neq j}^{b_n} \right] \\ &= \tilde{\mathbb{E}}_n \frac{d_n^{rel}(i)/b_n}{d_{in}^A(i) + 1 + \sum_{k=1,k\neq j}^{b_n} 1(W_k = i)} \geq \frac{d_n^{rel}(i)/b_n}{d_{in}^A(i) + 1 + \tilde{\mathbb{E}}_n \sum_{k=1,k\neq j}^{b_n} 1(W_k = i)} \\ &= \frac{d_n^{rel}(i)/b_n}{d_{in}^A(i) + 1 + (b_n - 1) d_n^{rel}(i)/b_n} > \frac{1}{2b_n} \frac{d_n^{rel}(i)}{d_{in}^A(i) + d_n^{rel}(i)},\end{aligned}$$

where in the first line we mean $\tilde{\mathbb{E}}_n[\cdot|X] = \mathbb{E}[\cdot|X, \{d_{out}(i'), d_{in}^A(i')\}_{i' \in [n]}]$ for any random variable X, the second equality holds by (E.78), the first inequality is Jensen's, the third equality again uses (E.78), and the second inequality uses $1 \leq d_{in}^A(i)$ (by assumption) and the obvious inequality $(b_n - 1)/b_n < 2$. Substituting into (E.79), we thus obtain

$$\tilde{\mathbb{E}}_n\left(1 - \tilde{p}_n\left(d_n^{rand}\right)\right) > \frac{1}{2} \sum_{i=1}^n \frac{d_{out}(i)}{m_n} \frac{d_n^{rel}(i)}{d_{in}^A(i) + d_n^{rel}(i)} = \frac{1}{2}\left(1 - \tilde{p}_n\left(d_n^{rel}\right)\right),$$

which, after rearranging, completes the proof.

E.3.3 Proof of Lemma E.16

For any $w \in (w_1, \ldots, w_{b_n}) \in [n]^{b_n}$, define

$$g_n(w) = \frac{1}{\max_{j \in [n]} r(j)} \sum_{j=1}^{b_n} \frac{d_{out}(w_j)}{d_{in}^A(w_j) + \sum_{k=1}^{b_n} 1(w_k = w_j)}.$$

Observe that, if W is the $[n]^{b_n}$-valued random vector with i.i.d. coordinates $\{W_j\}_{j\in[b_n]}$ s.t.

$$\tilde{\mathbb{P}}_n(W_j = k) = \frac{d_n^{rel}(k)}{\sum_{k'=1}^n d_n^{rel}(k')} = \frac{d_n^{rel}(k)}{b_n} \ \forall\ j \in [b_n], k \in [n]$$

(where the second equality holds by Lemma E.14), then the random variable $g_n(W)$ satisfies

$$\begin{aligned} g_n(W) &= \frac{1}{\max_{j\in[n]} r(j)} \sum_{j=1}^{b_n}\sum_{i=1}^{n} \frac{1(W_j = i) d_{out}(i)}{d_{in}^A(i) + \sum_{k=1}^{b_n} 1(W_k = i)} \qquad \text{(E.80)} \\ &= \frac{m_n}{\max_{j\in[n]} r(j)} \sum_{i=1}^{n} \frac{d_{out}(i)}{m_n} \frac{\sum_{j=1}^{b_n} 1(W_j = i)}{d_{in}^A(i) + \sum_{k=1}^{b_n} 1(W_k = i)} \\ &= \frac{m_n}{\max_{j\in[n]} r(j)} \sum_{i=1}^{n} \frac{d_{out}(i)}{m_n} \frac{d_n^{rand}(i)}{d_{in}^A(i) + d_n^{rand}(i)} = \frac{m_n}{\max_{j\in[n]} r(j)} \left(1 - \tilde{p}_n\left(d_n^{rand}\right)\right). \end{aligned}$$

Thus, we can analyze $g_n(W)$, then recover $\tilde{p}_n(d_n^{rand})$ by an affine transform. Working with $g_n(W)$ is convenient because g_n is a *self-bounding* function, defined as follows.

Definition E.1. [129, Section 3.3] Let $\mathcal{X}$ be some measurable space, $l \in \mathbb{N}$, and $f : \mathcal{X}^l \to [0, \infty)$. We say f is a *self-bounding* function if there exists auxiliary functions $f_{-i} : \mathcal{X}^{l-1} \to \mathbb{R}, i \in [l]$ such that, for any $x = (x_1, \ldots, x_l) \in \mathcal{X}^l$,

$$0 \le f(x) - f_{-i}(x_{-i}) \le 1 \ \forall\ i \in [l], \quad \sum_{i=1}^{l} (f(x) - f_{-i}(x_{-i})) \le f(x),$$

where $x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_l)\ \forall\ i \in [l]$.

To verify g_n is self-bounding, we use the most obvious choice of auxiliary functions: let

$$g_{n,-i}(w_{-i}) = \frac{1}{\max_{j\in[n]} r(j)} \sum_{j=1, j\neq i}^{b_n} \frac{d_{out}(w_j)}{d_{in}^A(w_j) + \sum_{k=1, k\neq i}^{b_n} 1(w_k = w_j)},$$

where $w_{-i} = (w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{b_n})$ for $w \in (w_1, \ldots, w_{b_n}) \in [n]^{b_n}$, i.e. we simply ignore the i-th coordinate of w. Towards bounding $g_n(w) - g_{n,-i}(w_{-i})$, we first observe

$$\begin{aligned} &\sum_{j=1, j\neq i}^{b_n} d_{out}(w_j) \left(\frac{1}{d_{in}^A(w_j) + \sum_{k=1}^{b_n} 1(w_k = w_j)} - \frac{1}{d_{in}^A(w_j) + \sum_{k=1, k\neq i}^{b_n} 1(w_k = w_j)} \right) \qquad \text{(E.81)} \\ &= \sum_{j=1, j\neq i}^{b_n} \frac{-1(w_i = w_j) d_{out}(w_j)}{(d_{in}^A(w_j) + \sum_{k=1}^{b_n} 1(w_k = w_j))(d_{in}^A(w_j) + \sum_{k=1, k\neq i}^{b_n} 1(w_k = w_j))} \qquad \text{(E.82)} \\ &= \sum_{j=1, j\neq i}^{b_n} \frac{-1(w_i = w_j) d_{out}(w_i)}{(d_{in}^A(w_i) + \sum_{k=1}^{b_n} 1(w_k = w_i))(d_{in}^A(w_i) + \sum_{k=1, k\neq i}^{b_n} 1(w_k = w_i))} \qquad \text{(E.83)} \end{aligned}$$

$$= \frac{-d_{out}(w_i)}{d_{in}^A(w_i) + \sum_{k=1}^{b_n} 1(w_k = w_i)} \times \frac{\sum_{k=1,k\neq i}^{b_n} 1(w_k = w_i)}{d_{in}^A(w_i) + \sum_{k=1,k\neq i}^{b_n} 1(w_k = w_i)} \tag{E.84}$$

$$\in \left(\frac{-d_{out}(w_i)}{d_{in}^A(w_i) + \sum_{k=1}^{b_n} 1(w_k = w_i)}, 0 \right), \tag{E.85}$$

where in (E.82) we computed the difference of fractions in (E.81), in (E.83) we replaced w_j by w_i (which is permitted due to the indicator $1(w_i = w_j)$), and in (E.84) we rearranged the expression; the upper bound in (E.85) is obvious, while the lower bound holds since the second factor in (E.84) is less than 1. Using the upper bound in (E.85), we can then obtain

$$g_n(w) - g_{n,-i}(w_{-i}) = \frac{\sum_{j=1,j\neq i}^{b_n} d_{out}(w_j) \left(\frac{1}{d_{in}^A(w_j) + \sum_{k=1}^{b_n} 1(w_k = w_j)} - \frac{1}{d_{in}^A(w_j) + \sum_{k=1,k\neq i}^{b_n} 1(w_k = w_j)} \right)}{\max_{j\in[n]} r(j)} \tag{E.86}$$

$$+ \frac{1}{\max_{j\in[n]} r(j)} \frac{d_{out}(w_i)}{d_{in}^A(w_i) + \sum_{k=1}^{b_n} 1(w_k = w_i)} \tag{E.87}$$

$$< \frac{1}{\max_{j\in[n]} r(j)} \frac{d_{out}(w_i)}{d_{in}^A(w_i) + \sum_{k=1}^{b_n} 1(w_k = w_i)} < \frac{r(w_i)}{\max_{j\in[n]} r(j)} \leq 1. \tag{E.88}$$

On the other hand, using the lower bound in (E.85), along with (E.86)-(E.87), we obtain $g_n(w) - g_{n,-i}(w_{-i}) > 0$. Together with (E.88), the first condition in Definition E.1 holds. To verify the second condition in Definition E.1, we use the leftmost expression in (E.88) to obtain

$$\sum_{i=1}^{b_n} (g_n(w) - g_{n,-i}(w_{-i})) < \frac{1}{\max_{j\in[n]} r(j)} \sum_{i=1}^{b_n} \frac{d_{out}(w_i)}{d_{in}^A(w_i) + \sum_{k=1}^{b_n} 1(w_k = w_i)} = g_n(w).$$

Having verified that g_n is self-bounding, we aim to show $g_n(W)$ concentrates around its mean. For this, we will use the following concentration inequality.

Theorem E.1. [129, Theorem 6.12] Let $X_1, \ldots, X_l$ be independent $\mathcal{X}$-valued random variables, define $X = (X_1, \ldots, X_l)$, and let $f : \mathcal{X}^l \to [0, \infty)$ be self-bounding. Then $\forall\ t \in (0, \mathbb{E}f(X)]$,

$$\mathbb{P}(f(X) \leq \mathbb{E}f(X) - t) \leq \exp\left(-\frac{t^2}{2\mathbb{E}f(X)} \right).$$

Applying the theorem to our setting, we obtain for any $t \in (0, \tilde{\mathbb{E}}_n g_n(W)]$,

$$\tilde{\mathbb{P}}_n(g_n(W) \leq \tilde{\mathbb{E}}_n g_n(W) - t) \leq \exp\left(-\frac{t^2}{2\tilde{\mathbb{E}}_n g_n(W)} \right). \tag{E.89}$$

Now for $\delta > 0$ define

$$t(\delta) = \frac{\delta}{2+\delta} \frac{m_n}{\max_{j\in[n]} r(j)} \frac{1 - \tilde{p}_n(d_n^{opt})}{2}.$$

Observe that, by Lemma E.15 and (E.80),

$$t(\delta) \leq \frac{\delta}{2+\delta} \frac{m_n}{\max_{j\in[n]} r(j)} \left(1 - \tilde{\mathbb{E}}_n \tilde{p}_n\left(d_n^{rand}\right)\right) = \frac{\delta}{2+\delta} \tilde{\mathbb{E}}_n g_n(W) < \tilde{\mathbb{E}}_n g_n(W).$$

Thus, for any $\delta > 0$, we can set $t = t(\delta)$ in (E.89). Furthermore, we have

$$\begin{aligned}
g_n(W) \leq \tilde{\mathbb{E}}_n g_n(W) - t(\delta) &= \tilde{\mathbb{E}}_n g_n(W) - \frac{\delta}{2+\delta} \frac{m_n}{\max_{j\in[n]} r(j)} \frac{1 - \tilde{p}_n(d_n^{opt})}{2} \\
\Leftrightarrow \frac{\max_{j\in[n]} r(j) g_n(W)}{m_n} &\leq \frac{\max_{j\in[n]} r(j) \tilde{\mathbb{E}}_n g_n(W)}{m_n} - \frac{\delta}{2+\delta} \frac{1 - \tilde{p}_n(d_n^{opt})}{2} \\
\Leftrightarrow 1 - \tilde{p}_n\left(d_n^{rand}\right) &\leq 1 - \tilde{\mathbb{E}}_n \tilde{p}_n\left(d_n^{rand}\right) - \frac{\delta}{2+\delta} \frac{1 - \tilde{p}_n(d_n^{opt})}{2} \\
\Leftarrow 1 - \tilde{p}_n\left(d_n^{rand}\right) &\leq \frac{1 - \tilde{p}_n(d_n^{opt})}{2} - \frac{\delta}{2+\delta} \frac{1 - \tilde{p}_n(d_n^{opt})}{2} = \frac{1 - \tilde{p}_n(d_n^{opt})}{2+\delta} \\
\Leftrightarrow \tilde{p}_n\left(d_n^{rand}\right) &\geq \frac{1 + \delta + \tilde{p}_n(d_n^{opt})}{2+\delta},
\end{aligned}$$

where the second and third implications hold by (E.80) and Lemma E.15, respectively, and the others are simple manipulations. Hence, by monotonicity and (E.89), we obtain $\forall\ \delta > 0$,

$$\tilde{\mathbb{P}}_n \left(\tilde{p}_n\left(d_n^{rand}\right) \geq \frac{1 + \delta + \tilde{p}_n(d_n^{opt})}{2+\delta} \right) \leq \exp\left(- \frac{t(\delta)^2}{2\tilde{\mathbb{E}}_n g_n(W)} \right)$$

Finally, we bound the exponential term. For this, we first note

$$\frac{t(\delta)}{\tilde{\mathbb{E}}_n g_n(W)} = \frac{\delta}{2(2+\delta)} \frac{1 - \tilde{p}_n(d_n^{opt})}{\max_{j\in[n]} r(j) \tilde{\mathbb{E}}_n g_n(W)/m_n} = \frac{\delta}{2(2+\delta)} \frac{1 - \tilde{p}_n(d_n^{opt})}{1 - \tilde{\mathbb{E}}_n \tilde{p}_n(d_n^{rand})} \geq \frac{\delta}{2(2+\delta)},$$

where the first inequality is the definition of $t(\delta)$, the second holds by (E.80), and the inequality holds by definition of d_n^{opt}. Combining the previous two inequalities, we thus obtain

$$\tilde{\mathbb{P}}_n \left(\tilde{p}_n\left(d_n^{rand}\right) \geq \frac{1 + \delta + \tilde{p}_n(d_n^{opt})}{2+\delta} \right) \leq \exp\left(- \frac{\delta^2}{4(2+\delta)^2} \frac{m_n(1 - \tilde{p}_n(d_n^{opt}))}{\max_{j\in[n]} r(j)} \right),$$

so choosing $c_\delta = \delta^2/(4(2+\delta)^2)$ completes the proof.

E.4 Proof of Corollary 6.1

Define $\hat{\vartheta}_{T_n}^{opt}(\phi)$ and $\hat{\vartheta}_{T_n}^{rand}(\phi)$ as in (E.4) but using the sequences $\{d_{out}(i), d_{in}^A(i), d_n^{opt}(i)\}_{i\in[n]}$ and $\{d_{out}(i), d_{in}^A(i), d_n^{rand}(i)\}_{i\in[n]}$, respectively. (In words, these are the beliefs of the root nodes in the trees induced by the optimal and randomized bot strategies, respectively.) The proof proceeds in two steps. First, we use the analysis of Theorem 6.1 to show

$$\theta_{T_n}^{opt}(i^*) \xrightarrow[n\to\infty]{\mathbb{P}} 0 \Leftrightarrow \mathbb{E}\hat{\vartheta}_{T_n}^{opt}(\phi) \xrightarrow[n\to\infty]{} 0, \quad \theta_{T_n}^{rand}(i^*) \xrightarrow[n\to\infty]{\mathbb{P}} 0 \Leftrightarrow \mathbb{E}\hat{\vartheta}_{T_n}^{rand}(\phi) \xrightarrow[n\to\infty]{} 0. \tag{E.90}$$

Second, we again leverage the analysis of Theorem 6.1, and also invoke Theorem 6.4, to show

$$\mathbb{E}\hat{\vartheta}^{opt}_{T_n}(\phi) \xrightarrow[n\to\infty]{} 0 \Leftrightarrow \mathbb{E}\hat{\vartheta}^{rand}_{T_n}(\phi) \xrightarrow[n\to\infty]{} 0. \tag{E.91}$$

Combining (E.90) and (E.91) then completes the proof.

We note the proof of (E.90) will specifically use Lemmas E.1, E.2, and E.3 from the Theorem 6.1 analysis; these lemmas require (A1), (A2), and (A4), but not (A3) (hence the assumptions of the corollary). To prove (E.91), we will use the analysis leading to (E.27) in Appendix E.2.2.1; this analysis does not require any of the four assumptions and thus applies.

E.4.1 First step for proof of Corollary 6.1

We only prove the first equivalence in (E.90); the proof does not rely on the choice of $\{d^{opt}_n(i)\}_{i\in[n]}$, so the same logic establishes the second. The proof combines a standard result ($X_n \to 0$ in $\mathbb{P} \Leftrightarrow \mathbb{E}X_n \to 0$ for uniformly-bounded/non-negative random variables $\{X_n\}_{n\in\mathbb{N}}$) with the fact that $\theta^{opt}_{T_n}(i^*)$ and $\hat{\vartheta}^{opt}_{T_n}(\phi)$ behave similarly per the proof of Theorem 6.1.

First, assume $\mathbb{E}\hat{\vartheta}^{opt}_{T_n}(\phi) \to 0$. Then for any $\varepsilon > 0$ and all n large,

$$\begin{aligned}\mathbb{P}\left(\theta^{opt}_{T_n}(i^*) > \varepsilon\right) &\le \mathbb{P}\left(\hat{\vartheta}^{opt}_{T_n}(\phi) > \varepsilon/2\right) + \mathbb{P}\left(\Omega^C_{n,1}\right) + O\left(n^{\zeta-1/2}\right)\\ &\le \frac{2}{\varepsilon}\mathbb{E}\hat{\vartheta}^{opt}_{T_n}(\phi) + \mathbb{P}\left(\Omega^C_{n,1}\right) + O\left(n^{\zeta-1/2}\right) \xrightarrow[n\to\infty]{} 0,\end{aligned}$$

where the first inequality is Lemma E.3 (with $x = 0$), the second is Markov's, and the limit holds by assumption $\mathbb{E}\hat{\vartheta}^{opt}_{T_n}(\phi) \to 0$ and by (A1)-(A2).

Next, assume $\theta^{opt}_{T_n}(i^*) \to 0$ in $\mathbb{P}$. We desire an inequality analogous to Lemma E.3, but pointing in the opposite direction; we derive one using logic similar to the proof of Lemma E.3. First, define $\vartheta^{opt}_{T_n}(i^*)$ as in (E.1) but using $\{d_{out}(i), d^A_{in}(i), d^{opt}_n(i)\}_{i\in[n]}$; in words, $\vartheta^{opt}_{T_n}(i^*)$ is like $\theta^{opt}_{T_n}(i^*)$ but ignores the prior parameters. We can then write the following:

$$\begin{aligned}\mathbb{P}\left(\theta^{opt}_{T_n}(i^*) \ge \varepsilon/2\right) &\ge \mathbb{P}\left(\vartheta^{opt}_{T_n}(i^*) > \varepsilon\right) \ge \mathbb{P}\left(\vartheta^{opt}_{T_n}(i^*) > \varepsilon \middle| \tau^{opt}_n > T_n\right)\mathbb{P}\left(\tau^{opt}_n > T_n\right)\\ &\ge \mathbb{P}\left(\vartheta^{opt}_{T_n}(i^*) > \varepsilon \middle| \tau^{opt}_n > T_n\right)\mathbb{P}\left(\tau^{opt}_n > T_n \middle| \Omega_{n,1}\right)\mathbb{P}(\Omega_{n,1})\\ &= \mathbb{P}\left(\hat{\vartheta}^{opt}_{T_n}(\phi) > \varepsilon\right)\mathbb{P}\left(\tau^{opt}_n > T_n \middle| \Omega_{n,1}\right)\mathbb{P}(\Omega_{n,1}),\end{aligned}$$

Here the first inequality holds for n large by Lemma E.1, the next two hold by monotonicity (here τ^{opt}_n is the first time at which the graph is no longer treelike; see Algorithm E.1), and the equality holds by Lemma E.2 ($\vartheta^{opt}_{T_n}(i^*)$ and $\hat{\vartheta}^{opt}_{T_n}(\phi)$ have the same distribution when the graph is treelike). Now by assumption $\theta^{opt}_{T_n}(i^*) \to 0$ in $\mathbb{P}$, Lemma E.2, and (A1),

$$\lim_{n\to\infty}\mathbb{P}\left(\theta^{opt}_{T_n}(i^*) > \varepsilon/2\right) = 0, \quad \lim_{n\to\infty}\mathbb{P}\left(\tau^{opt}_n > T_n \middle| \Omega_{n,1}\right) = \lim_{n\to\infty}\mathbb{P}(\Omega_{n,1}) = 1.$$

Combining the above, and since $\varepsilon > 0$ was arbitrary, we conclude $\hat{\vartheta}^{opt}_{T_n}(\phi) \to 0$ in $\mathbb{P}$. Hence,

because $\hat{\vartheta}^{opt}_{T_n}(\phi) \in [0,1]$ $a.s.$, we have for any $\varepsilon > 0$ and for all n sufficiently large,

$$0 \leq \mathbb{E}\hat{\vartheta}^{opt}_{T_n}(\phi) \leq \frac{\varepsilon}{2}\mathbb{P}\left(\hat{\vartheta}^{opt}_{T_n}(\phi) \leq \frac{\varepsilon}{2}\right) + \mathbb{P}\left(\hat{\vartheta}^{opt}_{T_n}(\phi) > \frac{\varepsilon}{2}\right) < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

E.4.2 Second step for proof of Corollary 6.1

To prove (E.91), we use the following notation: for $d_n = (d_n(1), \ldots, d_n(n)) \in \mathbb{N}_0^n$, define

$$\tilde{p}^*_n(d_n) = \frac{1}{n}\sum_{i=1}^{n} \frac{d^A_{in}(i)}{d^A_{in}(i) + d_n(i)},$$

which is simply the random variable $\tilde{p}^*_n$ defined in (6.8), viewed as a function of the bot degrees d_n (similar to how we defined $\tilde{p}_n(d)$ in (6.14)). For such d_n, also define

$$\begin{aligned} g(d_n) &= \frac{\theta}{T_n}\sum_{t=0}^{T_n-1}\left(\sum_{l=1}^{t}\binom{t}{l}\eta^l(1-\eta)^{t-l}\tilde{p}^*_n(d_n)\left(\tilde{p}_n(d_n)\right)^{l-1} + (1-\eta)^t\right) \\ &= \frac{\theta\tilde{p}^*_n(d_n)}{\eta\tilde{p}_n(d_n)}\frac{1-(1-\eta(1-\tilde{p}_n(d_n)))^{T_n}}{T_n(1-\tilde{p}_n(d_n))} + \frac{\theta}{T_n}\left(1 - \frac{\tilde{p}^*_n(d_n)}{\tilde{p}_n(d_n)}\right)\frac{1-(1-\eta)^{T_n}}{\eta}, \end{aligned} \tag{E.92}$$

where the second equality follows as in (E.27) from Appendix E.2.2.1; note from the first expression that $g(d_n)$ monotonically increases in $\tilde{p}^*_n(d_n)$ and $\tilde{p}_n(d_n)$. Also, by (E.26),

$$\mathbb{E}\hat{\vartheta}^{opt}_{T_n}(\phi) = \mathbb{E}g\left(d^{opt}_n\right), \quad \mathbb{E}\hat{\vartheta}^{rand}_{T_n}(\phi) = \mathbb{E}g\left(d^{rand}_n\right).$$

Hence, we aim to show $\mathbb{E}g(d^{opt}_n) \to 0 \Leftrightarrow \mathbb{E}g(d^{opt}_n) \to 0$. By the monotonicity observed above, this requires showing $\tilde{p}_n(d^{rand}_n)$ and $\tilde{p}_n(d^{opt}_n)$ are comparable, for which we will invoke Theorem 6.4. In contrast, there is no obvious relationship between $\tilde{p}^*_n(d^{rand}_n)$ and $\tilde{p}^*_n(d^{opt}_n)$ in the general case. However, in the case of a sublinear budget (i.e. $b_n = o(n)$), we can derive useful bounds on these terms. Thus, we begin by restricting to this case; we then return to address the case $b_n = \Omega(n)$.

E.4.2.1 Sublinear budget case

We begin by lower bounding $\tilde{p}^*_n(d_n)$. We claim that for any $\{d_n\}_{n\in\mathbb{N}}$ satisfying $d_n \in \mathbb{N}_0^n$ and $\sum_{i=1}^n d_n(i) \leq b_n$ for each n (note d^{opt}_n, d^{rand}_n both satisfy this),

$$\exists\ N \in \mathbb{N}\ s.t.\ \forall\ n \geq N,\ \tilde{p}^*_n(d_n) \geq 1/2. \tag{E.93}$$

Suppose instead $\forall\ N \in \mathbb{N}\ \exists\ n \geq N$ satisfying $\tilde{p}^*_n(d_n) < 1/2$. For such n, we have

$$\frac{1}{2} > \frac{1}{n}\sum_{i\in[n]}\frac{d^A_{in}(i)}{d^A_{in}(i)+d_n(i)} \geq \frac{1}{n}\sum_{i\in[n]:d_n(i)=0}\frac{d^A_{in}(i)}{d^A_{in}(i)+d_n(i)} = \frac{1}{n}\left|\{i\in[n] : d_n(i) = 0\}\right|,$$

where the second inequality holds as all summands are non-negative. On the other hand,

$$b_n \geq \sum_{i\in[n]} d_n(i) = \sum_{i\in[n]:d_n(i)\in\mathbb{N}} d_n(i) \geq |\{i \in [n] : d_n(i) \in \mathbb{N}\}| = n - |\{i \in [n] : d_n(i) = 0\}|,$$

where we used the fact that $d_n(i) \in \mathbb{N}_0 \ \forall\ i$. Combining the previous two inequalities,

$$\forall\ N \in \mathbb{N}\ \exists\ n \geq N\ s.t.\ b_n \geq n - |\{i \in [n] : d_n(i) = 0\}| > n/2,$$

which contradicts $b_n = o(n)$, completing the proof of (E.93).

We next show $\mathbb{E}g(d_n^{opt}) \to 0 \Rightarrow \mathbb{E}g(d_n^{rand}) \to 0$. First, we claim that for constant $c > 0$,

$$\lim_{n\to\infty} \mathbb{P}\left(1 - \tilde{p}_n\left(d_n^{opt}\right) \leq c/T_n\right) = 0. \tag{E.94}$$

Assume for the sake of contradiction that (E.94) fails. Then for some $\varepsilon > 0$,

$$\begin{aligned}
&\limsup_{n\to\infty} \mathbb{P}\left(1 - \tilde{p}_n\left(d_n^{opt}\right) \leq c/T_n\right) \geq \varepsilon, \\
\Rightarrow &\limsup_{n\to\infty} \mathbb{E}g\left(d_n^{opt}\right) \geq \varepsilon \limsup_{n\to\infty} \mathbb{E}\left[g\left(d_n^{opt}\right) \middle| 1 - \tilde{p}_n\left(d_n^{opt}\right) \leq c/T_n\right].
\end{aligned}$$

Now assume n is such that $\tilde{p}_n^*(d_n^{opt}) \geq 1/2$ (i.e. n is large enough that the lower bound derived above holds). Then $1 - \tilde{p}_n\left(d_n^{opt}\right) \leq c/T_n$ implies

$$\begin{aligned}
g(d_n^{opt}) &\geq \frac{\theta(1/2)}{\eta(1 - c/T_n)} \frac{1 - (1 - \eta c/T_n)^{T_n}}{c} + \frac{\theta}{T_n}\left(1 - \frac{1/2}{1 - c/T_n}\right) \frac{1 - (1-\eta)^{T_n}}{\eta} \\
&\xrightarrow[n\to\infty]{} \frac{\theta}{2\eta} \frac{1 - e^{-\eta c}}{c},
\end{aligned}$$

where we used the fact that $g_n(d_n^{opt})$ is monotone in $\tilde{p}_n^*(d_n^{opt})$ and $\tilde{p}_n(d_n^{opt})$, and for the limit we used $T_n \to \infty$ by (A4). Combining the previous two lines, we obtain

$$\limsup_{n\to\infty} \mathbb{E}g\left(d_n^{opt}\right) \geq \varepsilon \frac{\theta}{2\eta} \frac{1 - e^{-\eta c}}{c} > 0,$$

which contradicts the assumption $\mathbb{E}g(d_n^{opt}) \to 0$. This establishes (E.94).

Next, we prove (E.94) holds with d_n^{opt} replaced by d_n^{rand}. For constants $c, \delta > 0$,

$$1 - \tilde{p}_n\left(d_n^{opt}\right) > \frac{c(2+\delta)}{T_n}, \quad \frac{1 - \tilde{p}_n(d_n^{rand})}{1 - \tilde{p}_n(d_n^{opt})} > \frac{1}{2+\delta} \quad \Rightarrow \quad 1 - \tilde{p}_n\left(d_n^{rand}\right) > \frac{c}{T_n}.$$

Thus, by monotonicity and the inclusion-exclusion principle,

$$\begin{aligned}
\mathbb{P}\left(1 - \tilde{p}_n\left(d_n^{rand}\right) > \frac{c}{T_n}\right) &\geq \mathbb{P}\left(1 - \tilde{p}_n\left(d_n^{opt}\right) > \frac{c(2+\delta)}{T_n}, \frac{1 - \tilde{p}_n(d_n^{rand})}{1 - \tilde{p}_n(d_n^{opt})} > \frac{1}{2+\delta}\right) \\
&\geq \mathbb{P}\left(1 - \tilde{p}_n\left(d_n^{opt}\right) > \frac{c(2+\delta)}{T_n}\right) + \mathbb{P}\left(\frac{1 - \tilde{p}_n(d_n^{rand})}{1 - \tilde{p}_n(d_n^{opt})} > \frac{1}{2+\delta}\right) - 1 \xrightarrow[n\to\infty]{} 1,
\end{aligned}$$

where the limit holds by (E.94) and Theorem 6.4.

Finally, we show $\mathbb{E}g(d_n^{rand}) \to 0$. First, we note that by the inclusion-exclusion argument of the previous line, along with (A1), we have for any constant $c > 0$,

$$\lim_{n\to\infty} \mathbb{P}\left(\Omega_{n,1}, 1 - \tilde{p}_n\left(d_n^{rand}\right) > c/T_n\right) = 1.$$

Consequently, since $g(d_n^{rand}) \leq 1$ *a.s.*, and assuming the limits exist,

$$\mathbb{E}g\left(d_n^{rand}\right) \leq \mathbb{E}\left[g\left(d_n^{rand}\right)\middle|\Omega_{n,1}, 1 - \tilde{p}_n\left(d_n^{rand}\right) > c/T_n\right] + \mathbb{P}\left(\left(\Omega_{n,1}, 1 - \tilde{p}_n\left(d_n^{rand}\right) > c/T_n\right)^C\right)$$

$$\Rightarrow \lim_{n\to\infty} \mathbb{E}g\left(d_n^{rand}\right) \leq \lim_{n\to\infty} \mathbb{E}\left[g\left(d_n^{rand}\right)\middle|\Omega_{n,1}, 1 - \tilde{p}_n\left(d_n^{rand}\right) > c/T_n\right],$$

so it suffices to show $\mathbb{E}g(d_n^{rand}) \to 0$ conditioned on $\Omega_{n,1}$ and $1 - \tilde{p}_n(d_n^{rand}) > c/T_n$. Given these events, and using the trivial upper bound $\tilde{p}_n^*(d_n^{rand}) \leq 1$, we have by (E.92),

$$\begin{aligned} g_n\left(d_n^{rand}\right) &\leq \frac{\theta}{\eta(1-c/T_n)} \frac{1-(1-\eta c/T_n)^{T_n}}{c} + \frac{\theta}{T_n}\left(1 - \frac{1}{1-c/T_n}\right)\frac{1-(1-\eta)^{T_n}}{\eta} \\ &\xrightarrow[n\to\infty]{} \frac{\theta(1-e^{-\eta c})}{\eta c}, \end{aligned}$$

where the limit uses $T_n \to \infty$ by (A4). Note the limit can be made arbitrarily small by choosing c sufficiently large. In particular, given any $\varepsilon > 0$, we can choose $c = c_\varepsilon$ such that

$$\lim_{n\to\infty} \mathbb{E}\left[g(d_n^{rand})\middle|\Omega_{n,1}, 1 - \tilde{p}_n\left(d_n^{rand}\right) > c_\varepsilon/T_n\right] < \varepsilon,$$

which completes the proof of $\mathbb{E}g(d_n^{rand}) \to 0$.

The proof of $\mathbb{E}\hat{\vartheta}_{T_n}^{rand}(\phi) \to 0 \Rightarrow \mathbb{E}\hat{\vartheta}_{T_n}^{opt}(\phi) \to 0$ is essentially identical, so for brevity we only outline it. First, we can use $\mathbb{E}\hat{\vartheta}_{T_n}^{rand}(\phi) \to 0$ and the $\tilde{p}_n^*$ lower bound to prove (E.94) with d_n^{opt} replaced by d_n^{rand}. This immediately implies (E.94), simply by definition of d_n^{opt} (i.e. we need not invoke Theorem 6.4). From (E.94), $\mathbb{E}\hat{\vartheta}_{T_n}^{opt}(\phi) \to 0$ follows as above.

E.4.2.2 Linear budget case

We next consider the case $\liminf_{n\to\infty} b_n/n > 0$. The basic idea is as follows. Since average in-degree is constant by (A1), we can find a constant fraction of nodes whose in-degrees are bounded by some constant d. We can then (naively) connect one bot to each of b_n nodes, each with in-degree bounded by d. In this naive strategy, a constant fraction of nodes will have a constant fraction of bot in-neighbors. Consequently, $\tilde{p}_n \to 1$ cannot occur, which will imply the naive strategy drives the typical belief to zero. Finally, since even this naive scheme drives the belief to zero, the randomized and optimal schemes will as well.

More specifically, we will construct a naive choice of bot degrees d_n^{naive} satisfying

$$\exists\, \varepsilon \in (0,1), N \in \mathbb{N}\ s.t.\ \forall\ n \geq N,\ \Omega_{n,1} \Rightarrow \tilde{p}_n(d_n^{naive}) < 1 - \varepsilon. \tag{E.95}$$

We claim (E.95) is sufficient to show $\mathbb{E}g(d_n^{opt}), \mathbb{E}g(d_n^{rand}) \to 0$. Indeed, for d_n^{opt} we have

$$\begin{aligned}\lim_{n\to\infty} \mathbb{E}g\left(d_n^{opt}\right) &\leq \lim_{n\to\infty} \mathbb{E}\left[g\left(d_n^{opt}\right)\middle|\Omega_{n,1}\right] + \lim_{n\to\infty} \mathbb{P}\left(\Omega_{n,1}^C\right) \\ &\leq \lim_{n\to\infty} \left(\frac{\theta}{\eta(1-\varepsilon)} \frac{1-(1-\eta\varepsilon)^{T_n}}{T_n\varepsilon} + \frac{\theta}{T_n}\left(1 - \frac{1}{1-\varepsilon}\right)\frac{1-(1-\eta)^{T_n}}{\eta}\right) + \lim_{n\to\infty} \mathbb{P}\left(\Omega_{n,1}^C\right) = 0,\end{aligned}$$

where the second inequality uses $\tilde{p}_n(d_n^{opt}) \leq \tilde{p}_n(d_n^{naive})$ by definition of d_n^{opt} and $\tilde{p}_n(d_n^{naive}) < 1-\varepsilon$ on $\Omega_{n,1}$ for large n by (E.95), and the trivial inequality $\tilde{p}_n^*(d_n^{opt}) \leq 1$, and the equality holds since $T_n \to \infty$ by (A2) and since $\mathbb{P}(\Omega_{n,1}) \to 1$ by (A1). For d_n^{rand}, we have $\forall\ \delta > 0$,

$$\begin{aligned}\lim_{n\to\infty} \mathbb{E}g\left(d_n^{rand}\right) &\leq \lim_{n\to\infty} \mathbb{E}\left[g\left(d_n^{rand}\right)\middle|\Omega_{n,1}, \tilde{p}_n\left(d_n^{rand}\right) < \left(1+\delta+\tilde{p}_n\left(d_n^{opt}\right)\right)/(2+\delta)\right] \\ &\quad + \lim_{n\to\infty} \mathbb{P}\left(\Omega_{n,1}^C\right) + \lim_{n\to\infty} \mathbb{P}\left(\tilde{p}_n\left(d_n^{rand}\right) \geq \left(1+\delta+\tilde{p}_n\left(d_n^{opt}\right)\right)/(2+\delta)\right) \quad \text{(E.96)} \\ &\leq \lim_{n\to\infty} \left(\frac{\theta}{\eta(1-\varepsilon/(2+\delta))} \frac{1-(1-\eta\varepsilon/(2+\delta))^{T_n}}{T_n\varepsilon/(2+\delta)} + \frac{\theta}{T_n}\left(1 - \frac{1}{1-\varepsilon/(2+\delta)}\right)\frac{1-(1-\eta)^{T_n}}{\eta}\right) \\ &= 0,\end{aligned}$$

where the logic is similar, but we also Theorem 6.4 to equate (E.96) to zero.

It only remains to prove (E.95). Towards this end, we first show that for any $c \in (0,1)$,

$$\exists\ d \in (0,\infty), N \in \mathbb{N}\ s.t.\ \forall\ n \geq N,\ \Omega_{n,1} \Rightarrow \left|\left\{i \in [n] : d_{in}^A(i) \leq d\right\}\right| \geq cn, \tag{E.97}$$

i.e. when n is large and $\Omega_{n,1}$ holds, a constant fraction of nodes have bounded degrees. Suppose instead that (E.97) fails for some $c \in (0,1)$. Let $d = 3\nu_1/(1-c)$, where ν_1 is the limiting mean degree in (A1), and $N = \lceil \nu_1^{-1/\gamma} \rceil$, where γ is the rate of convergence in (A1). Then for $n \geq N$, $\Omega_{n,1}$ implies

$$m_n/n < \nu_1 + n^{-\gamma} \leq \nu_1 + N^{-\gamma} \leq 2\nu_1$$

By assumption, $\exists\ n \geq N$ satisfying $\Omega_{n,1}$ and $|\{i \in [n] : d_{in}^A(i) \leq d\}| < cn$. For such n,

$$2\nu_1 > \frac{m_n}{n} > \frac{\sum_{i\in[n]:d_{in}^A(i)>d} d_{in}^A(i)}{n} > \frac{d|\{i \in [n] : d_{in}^A(i) > d\}|}{n} \geq \frac{d(1-c)n}{n} = 3\nu_1,$$

which is clearly a contradiction. Consequently, (E.97) holds.

Finally, we use (E.97) to prove (E.95). Let $l = \liminf_{n\to\infty} b_n/n > 0$. Then $\exists\ N_1 \in \mathbb{N}$ s.t. $b_n \geq nl/2\ \forall\ n \geq N_1$. If $l > 2$, set $c = 1/2$; otherwise, set $c = l/2$. Then $c \in (0,1)$, so by (E.97) we can find $d \in (0,\infty), N_2 \in \mathbb{N}$ s.t. $\forall\ n \geq N_2$, $\Omega_{n,1} \Rightarrow |\{i \in [n] : d_{in}^A(i) \leq d\}| \geq cn$. Hence, for $n \geq \max\{N_1, N_2\}$ satisfying $\Omega_{n,1}$, we can find $\mathcal{I}_n \subset [n]$ satisfying

$$cn \leq |\mathcal{I}_n| \leq b_n, \quad d_{in}^A(i) \leq d\ \forall\ i \in \mathcal{I}_n. \tag{E.98}$$

For such n, we define $d_n^{naive}(i) = 1(i \in \mathcal{I}_n)$ and observe

$$\begin{aligned}\tilde{p}_n\left(d_n^{naive}\right) &= \sum_{i\in\mathcal{I}_n} \frac{d_{out}(i)}{m_n}\frac{d_{in}^A(i)}{d_{in}^A(i)+1} + \sum_{i\notin\mathcal{I}_n}\frac{d_{out}(i)}{m_n} \le \frac{d}{d+1}\sum_{i\in\mathcal{I}_n}\frac{d_{out}(i)}{m_n} + \sum_{i\notin\mathcal{I}_n}\frac{d_{out}(i)}{m_n} \\ &= 1 - \frac{1}{d+1}\frac{\sum_{i\in\mathcal{I}_n} d_{out}(i)}{m_n} \le 1 - \frac{1}{d+1}\frac{|\mathcal{I}_n|}{m_n} \le 1 - \frac{1}{d+1}\frac{cn}{m_n},\end{aligned}$$

where the first inequality holds by (E.98) and since $y/(y+1)$ increases, the second equality by definition of m_n, the second inequality since $d_{out}(i) \ge 1\ \forall\ i$, and the third inequality by (E.98). Thus, for any $n \ge \max\{N_1, N_2, \nu_1^{-\gamma}\}$, so that $m_n < 2\nu_1 n$ on $\Omega_{n,1}$ as above, we obtain $\Omega_{n,1} \Rightarrow \tilde{p}_n\left(d_n^{naive}\right) \le 1 - \frac{c}{4\nu_1(d+1)}$, so $\varepsilon = c/(4\nu_1(d+1))$ satisfies (E.95).

E.5 Experimental details

The basic workflow of the experiment in Section 6.3.3 proceeded as follows:

- Choose a sequence of time horizons T_n that increase linearly, then set n accordingly.
- Realize the degrees $\{d_{out}(i), d_{in}^A(i), d_{in}^B(i)\}_{i\in[n]}$ after selecting n.
- Define the empirical distributions f_n, f_n^* using the degrees as in (6.7).
- Evaluate quantity of interest $\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]$ empirically via (E.14) using f_n, f_n^*.

We repeated this 400 times to obtain 400 samples of $\mathbb{E}[\hat{\vartheta}_{T_n}(\phi)|\mathcal{T}]$; the plots in Figure 6.2 show empirical means and variances. We used the following parameters:

- We set $\eta = 0.9$ to emphasize the effect of the network.
- We let $d_{in}^A(i) = 1 + \text{Poisson}(\lambda_A - 1)\ \forall\ i \in [n]$, so that $\mathbb{E}[d_{in}^A(i)] = \lambda_A$; we choose $\lambda_A = 2.1$ so that $\mathbb{E}[d_{in}^A(i)] = O(1)$, as required by (A1).
- After realizing $\{d_{in}^A(i)\}_{i\in[n]}$, we assign one outgoing edge to each $i \in [n]$, then assign each of the remaining $\sum_{i\in[A]} d_{in}^A(i) - n$ outgoing edges independently and uniformly. This implies $d_{in}^A(i), d_{out}(i) > 0$ and $\sum_{i\in[n]} d_{in}^A(i) = \sum_{i\in[n]} d_{out}(i)$, as required by (6.5).
- We let $d_{in}^B(i) = \text{Poisson}(\lambda_B)$, with $\lambda_B = \lambda_A(1-p_n)/p_n$, so that

$$\mathbb{E}d_{in}^A(i)/(\mathbb{E}d_{in}^A(i) + \mathbb{E}d_{in}^B(i)) = \lambda_A/(\lambda_A+\lambda_B) = 1/(1-(1-p_n)/p_n) = p_n.$$

- We compare four cases of p_n: $p_n = p$ and $p_n = 1 - c_i T_n^{(-i+1)/2}$ for $i \in \{2,3,4\}$, with p and c_i independent of n. Note that the three latter cases satisfy

$$(1-p_n) \propto T_n^{(-i+1)/2} \in \left\{T_n^{-1/2}, T_n^{-1}, T_n^{-3/2}\right\},$$

  as shown in Figure 6.2. Here p and c_i were chosen so all four cases behaved roughly the same at the smallest value of n (as in Figure 6.2). In particular, we chose

$$p = 0.9, \quad c_2 = 1.3, \quad c_3 = 1.9, \quad c_4 = 2.7.$$

- We let $T_n \in \{2, 3, \ldots, 11\}$; here the minimum of 2 was chosen since $T_n = 1$ is a trivial case and the maximum of 11 was chosen due to computational limitations.
- Given T_n, we let $n = \lceil \lambda_A^{2T_n} \rceil$. Note that this implies $T_n \approx (\log n)/(2\log\lambda_A)$, roughly the upper bound in (A2). With our chosen T_n and λ_A, n ranged from 20 to $\approx 12 \times 10^6$.

Figure E.1 shows an analogue of Figure 6.3 with $b_n = \lceil \tilde{b}|E_n| \rceil$ for $\tilde{b} \in \{\frac{1}{1600}, \frac{1}{800}, \frac{1}{200}, \frac{1}{100}\}$.

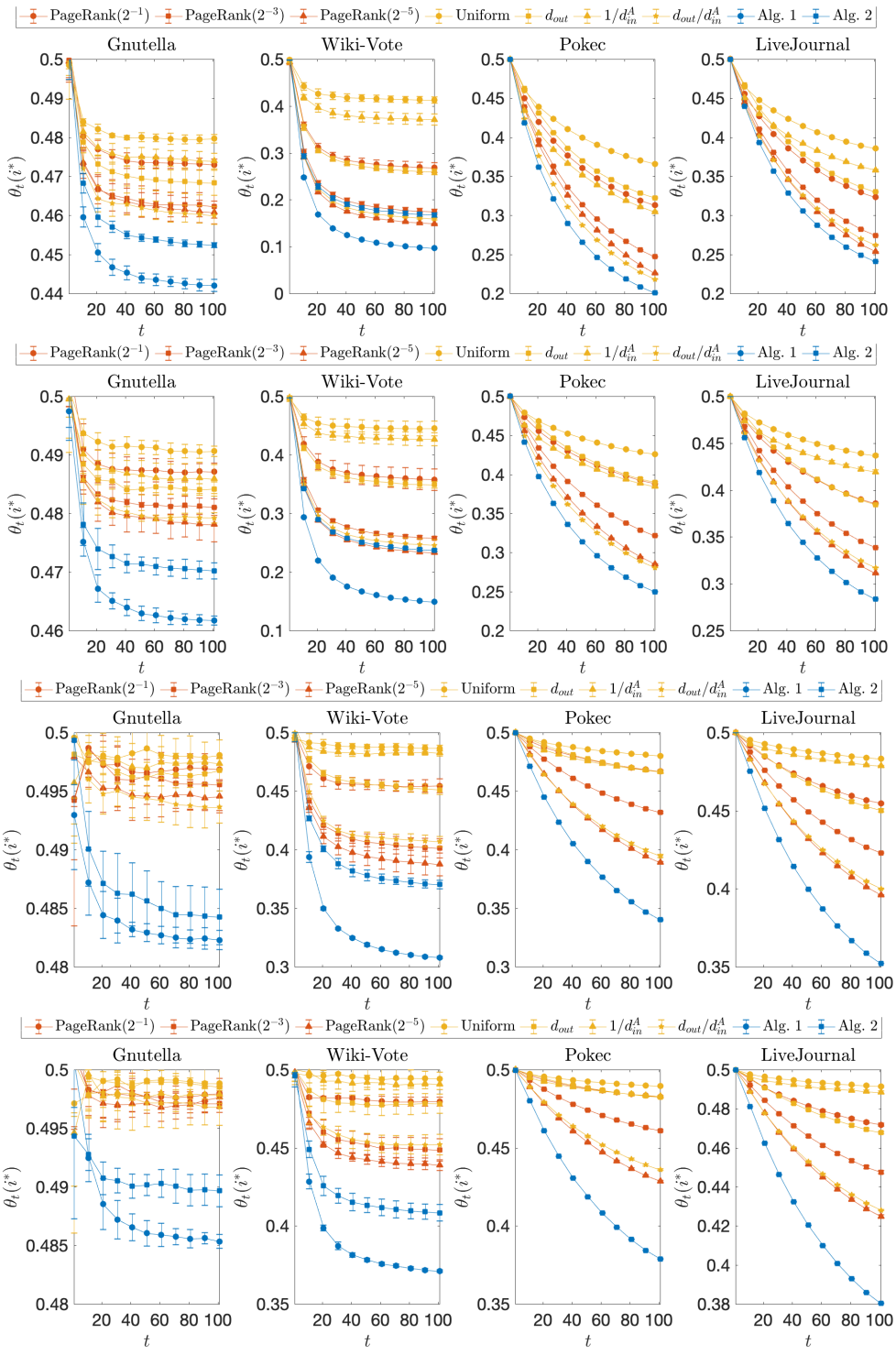


Figure E.1: Analogue of Figure 6.3 for $\tilde{b} = 1/100$, $1/200$, $1/800$, and $1/1600$, respectively.