Essays in Industrial Organization

by

Xuan Teng

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Economics)
in the University of Michigan
2022

Doctoral Committee:

Associate Professor Ying Fan, Chair
Assistant Professor Zach Brown
Assistant Professor Justin Huang
Professor Jagadeesh Sivadasan

Xuan Teng
tengx@umich.edu
ORCID iD: 0000-0001-7848-0719

DEDICATION

To my parents, Lingfeng Hu and Zhaobin Teng.

ACKNOWLEDGMENTS

This dissertation would not be possible without the support from a number of people. First and foremost, I am deeply indebted to my mentor, Ying Fan, for inspiring me to work on industrial organization, as well as the incredible amount of time, support, and advice she has given to me. I benefited a lot from being her student, research assistant, and advisee. I am also very grateful to my other advisors, Zach Brown, Jagadeesh Sivadasan, and Justin Huang for their continual encouragement, guidance, and support. I learned a lot from each of them. I would also like to thank my coauthor Travis Bruce Triggs for his invaluable contribution to our joint project. This dissertation also greatly benefited from in-depth conversations with Chenyu Yang and Francine Lafontaine. I gratefully acknowledge the financial support from the Department of Economics Research Grant at the University of Michigan in obtaining the data needed for parts of this dissertation.

I am grateful to numerous seminar participants for their constructive and helpful feedback on parts of this dissertation. At the University of Michigan, I greatly benefited from participants in the Industrial Organization Lunch and Industrial Organization Seminar. I also greatly benefited from seminar participants at Shanghai Jiao Tong University, Chinese University of Hong Kong-Shenzhen, Compass Lexecon, University of Munich, and Stony Brook University.

I want to thank my parents for giving me the freedom and support to become the person I want to be. I am also grateful for the joy my friends gave me at Michigan. Last but not least, my special thanks go to my best friend, great companion, and fiancé, Hai Zhu.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

ABSTRACT

One theme of industrial organization is to understand firm behaviors and its implication on market outcomes. This dissertation studies two firm behaviors that are closely related to competition: self-preferencing in digital platforms and patent licensing relationship between competing manufacturers.

Chapter 1 and Chapter 2 study self-preferencing on Apple App Store, a dominant dual-role platform for mobile applications. It is widely perceived that dual-role platforms, which sell both third-party and own products to consumers, may give preferential treatment to own products in search results to obtain competitive advantage over third-party suppliers. However, platforms deny it. Meanwhile, there is an intense antitrust debate over whether and how to regulate self-preferencing of dominant platforms. To shed light on the debate, this study addresses two research questions. Does there exist any self-preferencing on digital platforms? What are the consequences of self-preferencing on product quality and social welfare? I argue that self-preferencing leads to lower product quality and hurts both consumers and third-party suppliers. Meanwhile, the effect of self-preferencing is limited when there is confounded consumer preference.

In Chapter 1, I study demand for mobile applications and their search rankings on Apple App Store, in order to identify the existence of self-preferencing. To that end, I compile a new dataset of app characteristics, installation price, search ranking, downloads, revenues, conversion rates and search volume for population mobile applications on Apple App Store from April 2018 to February 2020. To motivate the demand model, I firstly examine the effect of an un-anticipated search algorithm change that dropped some Apple's apps from top search results. I find that the algorithm change improves the search ranking, downloads, and update frequency of third-party apps that are competing with Apple's apps in the same categories, compared to third-party apps in categories without Apple's apps. Then, I develop a demand model explicitly accounts for consumer search. It returns an app quality index capturing consumers preference conditional on prices. The search ranking model controls for the app quality index and captures potential self-preferencing. Estimation results direct to self-preferencing: Apple's apps are significantly more likely to show up in top search results compared to third-party applications, conditional on app quality, installation price, and title match with search terms. Counterfactual simulations show that eliminating the identified self-preferencing alone, without supply-side adjustment, shrinks the difference in search

rankings between Apple's apps and third-party apps by 71 percent.

The reduced-form evidence that update frequency is significantly affected by the search algorithm change motivates incorporating supply-side adjustment into the equilibrium analysis surrounding self-preferencing. To that end, Chapter 2 develops and estimates an empirical model of update competition in the presence of potential self-preferencing. Then, based on the estimates in Chapter 1 and Chapter 2, I conduct counterfactual simulations to examine the equilibrium welfare effect of self-preferencing. I find that shutting down the identified self-preferencing increases average update frequency by 0.4 percent, consumer surplus by 0.2 percent, and third-party profits by 0.7 percent. One reason for the modest effects of self-preferencing is consumer preference: consumers are estimated to prefer Apple's apps, which limits the extent of identified self-preferencing and thus its welfare effect.

Chapter 3 studies how unobserved patent licensing between competing manufacturers leads to bias in predicted merger effects. Patent licensing between competing manufacturers is typically hard to observe by researchers and thus not considered in classical pricing competition models for empirical merger analysis. Nevertheless, patent licensing introduces alignment incentives in firms' pricing decisions since licensors collect royalty revenues proportional to competing licensees' sales. Omitting these incentives leads to over-estimated marginal costs and potentially introduces bias in the prediction of merger effects. How large are the estimation and prediction biases? We conduct numerical simulations using a Bertrand competition model that incorporates patent licensing between competitors to assess the biases for both licensor-licensee mergers and licensee-licensee mergers in various simulated markets. We find that omitting patent licensing leads to predicted merger effects that are i) sometimes opposite to the true merger effects and ii) typically over-predicted.

CHAPTER 1

An Empirical Model of Self-preferencing and Demand for Mobile Applications

1.1 Introduction

Search algorithm shapes the competition landscape on digital platforms, and is designed by the platform. This raises up the concerns that dual-role platforms, which present and/or sell third-party products and service as well as those owned by the platforms, may give preferential treatment to their own products in search results, so that their products are ranked higher and thus more accessible to consumers compared to competing independent products. While the press covers multiple stories indicating the existence of such self-preferencing behaviors, the platforms deny it.[1] To shed light on the debate, this chapter develops an empirical model of demand and search ranking to statistically identify whether and to what extent self-preferencing exists on Apple App Store, a dominant platform for mobile applications.

Apple App Store provides an ideal setting for studying self-preferencing for three reasons. First, Apple App Store is a dual-role platform – Apple develops mobile applications and sells them on the app store in a variety of categories such as utilities, music and entertainment. Second, search is an important channel on the platform. While consumers may download an app listed in the editorial contents, 70% of visitors on the platform use search to find their next app, and 65% of all downloads on the platform happen after search during 2020.[2] Lastly, it is a well-tracked platform

[1]Example stories include (1) Dougherty, Conor. 2017. "Inside Yelp's Six-Year Grudge Against Google." The New York Times, July 1.https://www.nytimes.com/2017/07/01/technology/yelp-google-european-union-antitrust.html, (2) Mattioli, Dana. 2019. "Amazon Changed Search Algorithm in Ways That Boost Its Own Products" The Wall Street Journal, Sept. 16. https://www.wsj.com/articles/amazon-changed-search-algorithm-in-ways-that-boost-its-own-products-11568645345, (3) Nicas, Jack and Collins, Keith. 2019. "How Apple's Apps Topped Rivals in the App Store It Controls". The New York Times, Sept. 9. https://www.nytimes.com/interactive/2019/09/09/technology/apple-app-store-competition.html. As an example of how platforms deny the usage of self-preferencing, article (3) wrote "They (Apple's executives) said, Apple apps generally rank higher than competitors because of their popularity and because their generic names are often a close match to broad search terms".

[2]Apple. 2021. "Be discovered." https://searchads.apple.com/

such that there is rich aggregate-level data on consumer search and purchase, as well as historical search rankings.

Observed higher search rankings of platform-owned products may be due to preferential treatment, but may as well be explained by consumers' preferences for these products. To disentangle the two channels, I develop an empirical model of demand and search ranking. The demand model separately identifies search costs and consumer preferences. It returns an app quality index that captures consumers' evaluation of different apps in the same market, conditional on prices. Then, the search ranking model takes the app quality index as a control variable, so that the coefficient on platform ownership in the search ranking model captures potential self-preferencing.

To motivate the empirical demand model, I start with a descriptive analysis of an unanticipated search algorithm change on the Apple App Store. The search algorithm change dropped some of Apple's apps from top search results in July 2019, and was firstly reported by New York Times in September 2019.[3] Using a difference-in-differences (DiD) approach where I compare independent apps in categories containing Apple's apps with independent apps in categories without Apple's apps, I find that the search algorithm change boosted up independent apps by 3.6% in search results, leading to a 22% increase in their installations. However, it did not significantly affect their conversion rates. The result indicates that search rankings do not affect consumer choice conditional on search, which motivates me to assume that search ranking does not affect product values and only affects search cost.[4] This is a useful assumption that helps with separate identification of search costs and consumer preferences.

Next, I develop a demand model that explicitly accounts for consumer search. In the model, consumers sequentially search for apps and choose the most preferred app among the searched apps in the style of Weitzman (1979). This model allows consumers to form different search paths based on their different preferences for update frequency, prior knowledge on apps, and search costs. To obtain a closed-form choice probability, I borrow a parametric assumption on the distribution of consumer-product specific search costs from Moraga-González, Sándor and Wildenbeest (2018). To deal with endogenous app characteristics such as price, ratings, and update frequency, I construct excluded instruments based on multiple-category developers and the search algorithm change in July 2019.

To approximate the search ranking data, I fit it with a rank-ordered logistic regression model. The model allows flexible effects of well-accepted influential factors by including category-specific and month-specific effects of price, text relevance, and app ratings. Conditional these flexible effects and the constructed app quality index, I use monthly-specific coefficients on Apple-ownership

[3]Nicas, Jack and Collins, Keith. 2019. "How Apple's Apps Topped Rivals in the App Store It Controls". The New York Times, Sept. 9. https://www.nytimes.com/interactive/2019/09/09/technology/apple-app-store-competition.html

[4]Similarly, Ursu (2018) exploits experimental data on an online hotel platform and finds that rankings affect what consumers search, but conditional on search, do not affect purchases.

indicator to capture the dynamics of potential self-preferencing.

The model is estimated using aggregate data from AppTweak, a third-party research company in the mobile application industry. The data covers a wide variety of non-game and game apps on Apple App Store between April 2018 and February 2020 in 38 categories.[5] My main dataset contains information on estimated downloads and revenues[6], and actual app characteristics and search rankings for each app/category/month combination. In addition, category-month level conversion rates and keyword-month level search volume are observed to augment the study on consumer search. [7]

Estimation of the empirical model shows that while consumers significantly prefer Apple's apps over independent apps, there remains preferential treatment for Apple's apps in the search algorithm. In particular, estimation of the search ranking model shows that, from April to August in 2019, Apple's enjoyed significantly higher probability to be ranked in top search results compared to competing independent apps, conditional on consumer preference and apps' title match with search terms. As a cross validation, I also find that the preferential treatment on Apple's apps in the search algorithm significantly decreased after July 2019, which is consistent with the timing and purpose of the search algorithm change.

To shed light on the demand-side effect of the identified self-preferencing, I conduct counterfactual simulations based on the estimated consumer preference, search cost, and ranking probability. Specifically, I shut down the identified self-preferencing by setting the coefficient on Apple indicator in the search ranking model to be zero. I focus on the two months with the strongest identified preferential treatment, i.e., June and July 2019, and four main categories with Apple's apps: health & fitness, utilities, music, and entertainment. I find that the identified self-preferencing explains 71 percent of observed difference in search rankings between Apple's apps and independent apps. Without supply-side adjustment, eliminating the identified self-preferencing increases consumers surplus by 2.1 million dollars per month, which is a 0.2 percent improvement; while the variable profits of independent developers increase by 1.6 million dollars per month, which is a 0.6 percent rise. This changes the incentive for independent developers to update their apps and thus the equilibrium welfare effect, which will be a focus of Chapter 2.

[5]Specifically, the data covers all non-game app categories and 16 out of 18 game app categories. Table F.1 lists the covered app categories.

[6]Because actual downloads and revenues are highly confidential in the industry, literature on the mobile application industry commonly relies on estimated downloads and revenues. Section 1.2 shows that the estimated downloads and revenues fit the actual data well.

[7]In this study, conversion rate is the number of installations divided by the number of views. Search volume is an integer index between 5 and 100 describing the volume of consumers searching for a keyword on the platform, constructed by Apple.

1.1.1 Related Literature

This paper contributes to four strands of literature. First and foremost, this paper is related to the broad literature on information frictions and product competition, dated back to Stigler (1961) and Diamond (1971).[8] Information frictions have been theoretically and empirically shown to be able to shape the competition landscape in a large variety of markets.[9] A common focus of this literature is universal changes of information frictions for some or all products without affecting information frictions for others.[10] In contrast, this paper, by studying self-preferencing, focuses on differential changes of information frictions that reduce the search costs for some products but increase the search costs for others. Such differential changes of information frictions are commonly seen in papers studying platform design. Examples include Yao and Mela (2011), Arnosti, Johari and Kanoria (2014), Ghose, Ipeirotis and Li (2014), Nosko and Tadelis (2015), Fradkin (2017), Santos, Hortaçsu and Wildenbeest (2017), Dinerstein et al. (2018), Huang (2018), Choi and Mela (2019), Lam (2021), and Lee and Musolff (2021). In terms of topics, this paper is closely related to Dinerstein et al. (2018). However, there are three differences between Dinerstein et al. (2018) and this paper: design of search result display *v.s.* self-preferencing, homogeneous products *v.s.* differentiated products, pricing response *v.s.* product quality response. Despite this growing literature, to my knowledge, there is no evidence on the equilibrium welfare effects of platform design with endogenous non-price product attributes, and thus this paper complements the literature.

By quantifying consumer welfare with aggregate product-level data in the context of consumer search, this paper also contributes to the literature on empirical consumer search. This literature typically tests or estimates empirical consumer search models with two different types of data: i) individual-level data on search and purchase[11]; ii) aggregate product-level data that sometimes are augmented with moments of purchase and search behaviors[12]. In terms of methodology, the pa-

[8]For related review, please see Goldfarb and Tucker (2019), Anderson and Renault (2018) and Ratchford (2009).

[9]Theoretical work and empirical work have studied information frictions and competition in markets for health care, airlines, mutual funds, personal computers, automobiles, trade waste, books, and consumer electronics. Some examples include Dranove et al. (2003), Brown (2017), Brown (2019), Orlov (2015),Hortaçsu and Syverson (2004), Goeree (2008), Tadelis and Zettelmeyer (2015), Salz (2020), Bar-Isaac, Caruana and Cuñat (2012), Ellison and Ellison (2018), and Baye, Morgan and Scholten (2004).

[10]For example, Fishman and Levy (2015) theoretically examines the effects of search costs on investment in qualities. Brown (2017) examines how price transparency affects equilibrium prices and welfare in the medical imaging services market and finds a 22 percent reduction in prices if all patients had full information. Allen, Clark and Houde (2019) quantifies the role of search costs and brand loyalty for market power and finds that search frictions reduce consumer surplus by $12/month/consumer in mortgage markets.

[11]Examples include Roberts and Lattin (1991), Mehta, Rajiv and Srinivasan (2003), De los Santos, Hortaçsu and Wildenbeest (2012), Seiler (2013), Honka (2014), Koulayev (2014), Pires (2016), Honka, Hortaçsu and Vitorino (2017), Honka and Chintagunta (2017), Santos, Hortaçsu and Wildenbeest (2017), Ursu (2018), Murry and Zhou (2020), Jolivet and Turon (2019), and Compiani et al. (2021).

[12]Examples include Hong and Shum (2006), Kim, Albuquerque and Bronnenberg (2010), Kim, Albuquerque and Bronnenberg (2017), Moraga-González and Wildenbeest (2008), Moraga-González, Sándor and Wildenbeest (2015),

per is built on Moraga-González, Sándor and Wildenbeest (2018), which provides the theoretical foundation for the consumer search model in the paper. In particular, Moraga-González, Sándor and Wildenbeest (2018) uses a parametric search cost distribution for which a closed-form expression for choice probabilities is obtained. I apply the search cost distribution to consumer search in a different industry. The closed-form choice probability makes the estimation of the consumer search model of similar difficulty as most standard discrete choice models of demand. It also allows me to explicitly deal with price endogeneity by instruments, which is typically hard to achieve in the literature. However, Moraga-González, Sándor and Wildenbeest (2018) does not quantify consumer welfare, whereas I go one step further by simulating the consumer search model at consumer level given the estimated search cost distributions and preferences, and thus quantify the consumer welfare.[13]

By studying the upgrading decisions of app developers, this paper is also related to the literature on endogenous product choices.[14] The literature has studied endogenous product choices in a variety of industries, including retail video (Seim, 2006), retail eyeglasses (Watson, 2009), ice-cream (Draganska, Mazzeo and Seim, 2009), TV (Chu, 2010; Crawford and Yurukoglu, 2012; Crawford, Shcherbakov and Shum, 2019), CPU (Nosko, 2010), newspapers (Fan, 2013), home PC (Eizenberg, 2014), movie (Orhun, Venkataraman and Chintagunta, 2016), ratio (Berry, Eizenberg and Waldfogel, 2016), smartphones (Wang, 2017; Fan and Yang, 2020*a*), trucks (Wollmann, 2018), vendor allowances contracts (Hristakeva, 2019), and retail craft beer (Fan and Yang, 2020*b*). However, none of them has examined endogenous product choices in the mobile application industry; neither is there much evidence on the effects of platform design on endogenous product choices. This paper fills the gap.

Finally, this paper is also related to the emerging literature on mobile applications. Examples include Ghose and Han (2014), Bresnahan, Orsini and Yin (2014), Mendelson and Moon (2016), Mendelson and Moon (2018), Huang (2018), Leyden (2019), Ershov (2020), Li, Bresnahan and Yin (2016), Wang, Li and Singh (2018), Leyden (2021), Allon et al. (2021), Singh, Hosanagar and Nevo (2021) and Janssen et al. (2021). In terms of methodology, this paper is most closely related to Leyden (2019) which also develops an empirical model to rationalize app developers' upgrading decisions for a different research question. However, whereas Leyden (2019) assumes that consumers have perfect information about apps, my model allows consumers to have imperfect information about match values with apps. While imperfect information is not necessary for Leyden (2019) to study the effects of digitization, it is a necessary assumption to study the effect

Galenianos and Gavazza (2017), Moraga-González, Sándor and Wildenbeest (2018), Giulietti, Waterson and Wildenbeest (2014), Abaluck and Curto (2018), and Coey, Larsen and Platt (2020).

[13] Ursu (2018) uses a similar simulation approach to calculate consumer welfare, where the model is estimated with individual-level data.

[14] For related review, please see Crawford (2012).

of self-preferencing in this paper. In terms of topics, this paper is most closely related to Ershov (2020) which studies the effect of reduced consumer discovery costs on product entry and consumer welfare after re-categorization of game apps in the Google Play Store. However, whereas Ershov (2020) focuses on congestion effects and does not explicitly model consumer search, this paper builds on an explicit model of consumer search to separate search costs from utilities. The separation is crucial for measuring consumer welfare. Therefore, I complement these papers by studying the equilibrium welfare effects of self-preferencing with explicit consumer search model and endogenous upgrade decisions.

1.1.2 Roadmap

The remainder of the chapter uses "self-preferencing" and "preferential search ranking" exchangeably, and is organized as follows. Section 1.2 describes the data. Section 1.3 provides background on the search algorithm change in the U.S. market on the Apple App Store and presents the descriptive evidences. Section 1.4 describes the empirical model of demand and search ranking. Section 1.5 describes the estimation procedure and presents the estimation results. Section 1.6 presents counterfactual simulations. Section 1.7 concludes.

1.2 Data

My main data come from AppTweak. It covers popular apps and keywords in 38 four-digit categories on Apple App Store in the US market between April 2018 and February 2020. For every app in every month during the sample period, I observe the estimated downloads and revenues and actual characteristics such as installation prices, ratings, and versions. Furthermore, I observe the daily search ranking in the search results for every app and keyword in the sample. Additionally, I include two auxiliary datasets on app conversion rates and keyword search volumes.

Actual downloads and revenues of apps are highly confidential data. I use estimated downloads and revenues from AppTweak.[15] Figure G.1 shows that the fitness of the estimated downloads is satisfactory. Each download is an installation by a unique new consumer. Hereafter, "downloads" and "revenues" mean the estimated monthly downloads and estimated monthly revenues. Notice that the revenues include installation revenues and in-app-purchase and in-app-subscription revenues but do not cover in-app-advertising revenues.

Update is another important object of this study. I define an update as the release of a new version, which comes with a release note explaining the changes by the update. However, not every

[15] AppTweak estimates app daily downloads and revenues based on a machine learning algorithm by linking top charts rankings with actual downloads and revenues they observe from connected app developers. AppTweak reports that 42% of the top 50 grossing apps are their active customers.

version is of the same importance. For example, bug-fixing versions typically only contains "Bug fix" in the release note, while major versions typically come with long release notes describing the new features introduced in the new version. To capture such an idea, I give higher weights to updates with longer release notes within categories.[16] Then, update frequency of an app in a month is the weighted number of released versions of that app in the month. Appendix A.2 gives more details on variable construction.

I restrict my sample to popular apps and keywords to focus on the most important impacts of self-preferencing. The popular apps are selected based on multiple criteria, including rankings in category-specific top-grossing charts and top-paid charts[17], and annual downloads in 2019.[18] The category-specific popular keywords are selected based on how many popular apps in the category show up in the search results. Appendix A.1 gives more details on the sample selection process. In the end, the sample covers 3,110 apps and 2,265 keyword/category pairs. Figure G.2 plots weighted average residual downloads against search ranking. It shows that the selected popular keywords work well: downloads are negatively correlated with the search rankings in the search results of these keywords.[19] It shows that downloads are negatively correlated with the search rankings in the search results of these selected keywords, which indicates that the popular keywords work well.

Unlike some other contexts of consumer search[20], historical search results in Apple App Store are not available[21]. Therefore, instead of defining a market as a query on the App Store, I define a market as a category/month pair. To keep the model tractable, I aggregate the app-keyword-day level search ranking data to app-category-month level average search ranking, weighted by search volumes. To focus on the more important search results, the aggregation is conditional on top50 search results. [22] Hereafter, the "search ranking" of an app in a category in a month means the weighted average top50 search rankings of the app across popular keywords in the category and days in the month.

In the end, my sample consists of 56,570 observations, each of which is an app/cate-

[16]Leyden (2019) classifies updates into bug-fix updates and feature updates based on natural language processing and machine learning techniques, where he also exploits release note information.

[17]To ensure enough price variation for identifying the coefficient on price in the demand model, I added apps selected based on top-paid charts.

[18]Kim, Albuquerque and Bronnenberg (2017) similarly uses data covering 200 best-selling camcorders on Amazon.com.

[19]For each keyword in the sample, I additionally observe the daily search volume, which is an integer score that indicates how many consumers search for the keyword on that day.

[20]For example, Ursu (2018) uses the data from consumers' queries for hotels on Expedia.

[21]"An Apple spokeswoman said the company [. . .] did not keep a record of historical search results." See Nicas,Jack and Collins,Keith. 2019. "How Apple's Apps Topped Rivals in the App Store It Controls". The New York Times, Sept. 9. https://www.nytimes.com/interactive/2019/09/09/technology/apple-app-store-competition.html.

[22]I track the situation where an app/category/month combination never shows up in top50 search results in any keyword on any day with an indicator.

gory/month. Table 1.1 presents the summary statistics on the downloads, revenues, product characteristics, and search rankings. [23] An average app in my sample has 60,000 unique new consumers downloading it in a month and receives \$0.37 million from installation, in-app purchase and subscription. There is a large variation across the observations. The standard deviation of downloads is 3.8 times its mean, and the standard deviation of revenues is 4.9 times its mean. There is also a sizable variation in prices across observations of paid apps: among the 42% observations of paid apps, the average installation price is \$4.15, with a standard deviation of \$4.69. The variation in prices of paid apps helps with identifying price sensitivities in the mobile application markets. One percent of observations are from Apple's non-preinstalled apps. I also observe app characteristics, including update frequency, average ratings, age, file size, number of screenshots, length of description, and in-app-purchase availability. Across these app characteristics, their standard deviations are about 13 percent to 1.8 times their corresponding means, indicating a wide variety of apps in the sample.

As for search rankings, 62 percent of observed app/category/months have ever ranked top50 in the search results of at least one category-specific popular keyword on at least one day in the month. Among these observations, the average search ranking is 23.6, with a standard deviation of 11.6. This standard deviation is 49 percent of the mean, and the range covers the top and bottom of the top50 interval, indicating a wide variety of search rankings in the sample. It also shows that the search ranking of an average Apple's app is 7.5 positions higher than an average independent app. However, this might be due to higher qualities of Apple's apps or iPhone users' preferences for Apple's apps, instead of preferential treatment on Apple's apps in the search algorithm. I will disentangle the two in Section 1.5.

I focus on update frequency as developers' strategic response to self-preferencing for Apple's apps. One of the motivation roots in the over-time stickiness of product characteristics, which is shown in the right panel of Table 1.1. The first column of the right panel shows the average within-app standard deviation of the row app characteristic across months. While the average within-app over-time standard deviation of update frequency is 48 percent of its overall standard deviation, the same ratio for file size is just 4 percent. Similar patterns show up when comparing the average within-app range to the overall range. The second column in the right panel shows that, during the sample period, an average app's file size may change up to 49.4MB, accounting for 1% overall range of file sizes. In contrast, an average app's update frequency may change up to 1.6 weighted updates, accounting for 15% overall range of update frequencies. Stickiness also shows up in other app characteristics, such as prices and average ratings. The result indicates flexibility of update levels over time, compared to other app characteristics.

[23] Table F.3 presents summary statistics on other variables used in estimation, including market sizes measure by the number of iPhone users from Comscore.

Table 1.1: Summary Statistics

	Overall				Over-time Variation	
Variable	Mean	SD	Min	Max	Avg SD[c]	Avg range[d]
Downloads (million)	0.06	0.23	10^{-6}	7.00	0.03	0.09
Revenues (million \$)	0.37	1.81	0	56.51	0.07	0.22
Apple	0.01	0.07	0	1	0	0
Paid Installation	0.42	0.49	0	1	0	0
Paid Price	4.15	4.69	0[a]	99.99	0.32	0.82
Update Frequency	0.68	1.00	0	11	0.48	1.60
Average Rating	4.40	0.55	1	5	0.09	0.23
Age (month)	51.16	32.66	1	140	4.65	14.51
File Size	225.70	411.00	0.73	4096	17.84	49.41
#Screenshots	5.54	1.96	0	10	0.56	1.56
Description Length	2.21	1.04	0	4.00	0.15	0.40
Offer In-App-Purchase	0.74	0.44	0	1	0	0
Ever Top50 in Search Results	0.62	0.49	0	1	0.09	0.22
Search Ranking|Ever Top 50	23.57	11.58	1	50	4.83	14.11
Apple's apps[b]	16.14	12.01	1	50		
Independent apps	23.59	11.59	1	50		
Number of app/category/months		56,570				
Number of apps					3,110	

a. 9.87% paid apps have ever reduced their prices to 0 in the sample.
b. Include both pre-installed and non-pre-installed Apple's apps, which constitutes 644 app/category/month observations. Pre-installed apps are not included when calculating the summary statistics for other row variables. Search rankings of pre-installed apps are from Sensor Tower. The data for all the other variables come from public information and/or AppTweak.
c. Average of X_j, where $X_j = ($ standard deviation of x_{jt} across months indexed by $t)$. For search ranking related variables, x_{jt} is average x_{jgt} across categories indexed by g.
d. Average of X_j, where $X_j = ($ range of x_{jt} across months indexed by $t)$. For search ranking related variables, x_{jt} is average x_{jgt} across categories indexed by g.

As will be explained later in Section 1.5.1, my identification relies on developers that develop apps in multiple categories and their portfolio variation across categories. Table 1.2 show such variations. In an average market, there are 52.7 developers, 20.5 of whom are multiple-category developers.[24] This constitutes 7,261 multiple-category developer/month observations in the sample. Given a multiple-category developer/month pair, I first calculate the mean of each app characteristic on the row within each category, then calculate the standard deviation of this mean across

[24] An average multiple-category developer develops 2.3 apps in 2.5 categories. The larger average number of apps than the average number of categories reflects the industry feature that some apps operate in multiple categories simultaneously. In such cases, the characteristics of the same app in the other categories will not be used to construct instruments.

Table 1.2: Summary Statistics on App Characteristic Dispersion based on Multiple-Category Developers

Variable	Average cross-category SD within Developer/Month[a]	Average SD of developer's cross-category SD within Category/Month[b]
Update Frequency	0.73	0.52
Price ($)	0.34	0.75
Average Rating	0.09	0.13
Ever Top50	0.15	0.17

a. Average of X_{ft}, where $X_{ft} =$ (standard deviation of $(1/\#\mathscr{J}_{fgt})\sum_{j\in\mathscr{J}_{fgt}} x_{jgt}$ across categories indexed by g), where $\mathscr{J}_{fgt}$ is the set of apps owned by developer f in category g/month t.
b. Average of Y_{gt}, where $Y_{gt} =$ (standard deviation of $X_{f(j)t}$ across apps $j \in \mathscr{J}_{gt}$), where $X_{f(j)t}$ is the X_{ft} of the developer of app j, and $\mathscr{J}_{gt}$ is the set of apps in category g/month t.

categories. This standard deviation captures the cross-category variation within developer/months. The first column reports the mean of this standard deviation across all multiple-category developer/months. It indicates that there is significant portfolio variation across categories within the same developer/month. In the case of single-category developers, my identification relies on the within-market variation of the cross-category changes of product portfolios of the same developer. In order to show the variation, I match each app with its developer's cross-category changes of product portfolios captured by the standard deviation calculated in the first column. Then, I calculate the standard deviation of these cross-category changes of product portfolios across apps in each market. The second column reports the average of this standard deviation across markets. It indicates that there is a significant within-market variation of developers' cross-category changes of product portfolios.

1.3 Descriptive Evidence

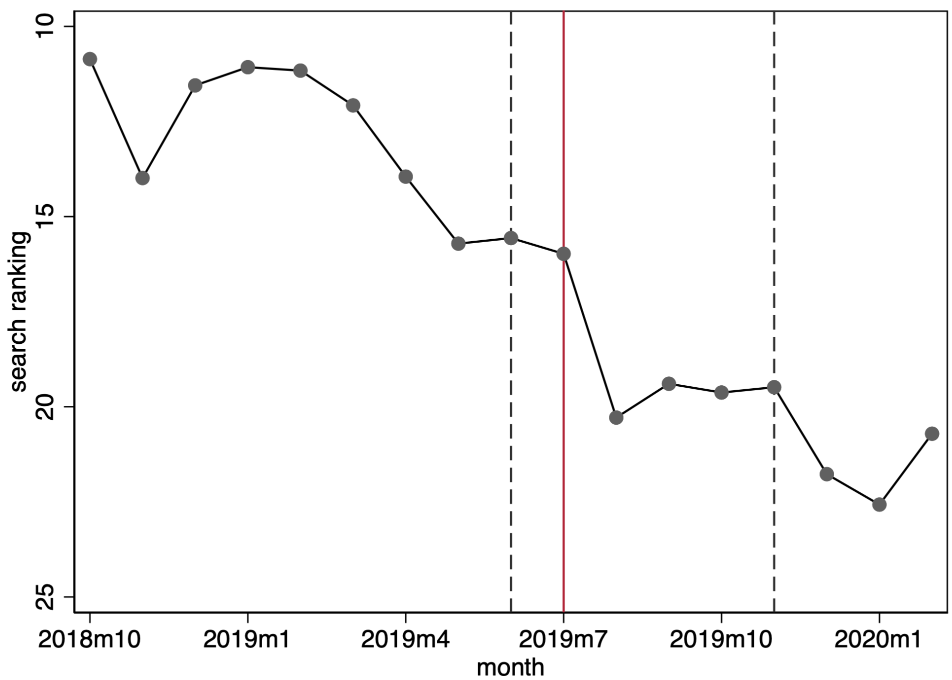
This section exploits an unanticipated search algorithm change on Apple App Store to provide reduced-form evidence on the effect of self-preferencing and to motivate the empirical model of mobile application industry.

1.3.1 Search Algorithm Change on Apple App Store

To motivate the empirical model, I exploit an unanticipated search algorithm change on Apple App Store in July 2019 as an exogenous variation to study the effects of self-preferencing on independent apps. The algorithm change is claimed to tweak "a feature of the App Store search engine that sometimes grouped apps by maker" so that "Apple apps would no longer look as if they were receiving special treatment".[25] The search algorithm change was firstly reported by New York Times in September 2019, so I argue it is unanticipated by the independent developers.[26].

After the search algorithm change, some of Apple's apps are dropped down from top search results, as shown in Figure 1.1. The vertical axis is the average search rankings of Apple's apps, including non-pre-installed and pre-installed Apple's apps. The horizontal axis is the month. It shows that, on average, search rankings of Apple's apps lower down dramatically after July 2019.[27] To focus on the effect of the search algorithm change, I look into the sample period between June and November of 2019.

Figure 1.1: Average Search Ranking of Apple's Apps around July 2019



To capture the effects of the search algorithm change, I exploit the differences in presence of Apple's apps across categories. In particular, there are 16 non-game categories with Apple's apps (pre-installed and non-pre-installed) and 22 non-game and game categories without Apple's apps. I assume that the apps in the categories with Apple's apps are affected by the search algorithm

[25]Nicas,Jack and Collins,Keith. 2019. "How Apple's Apps Topped Rivals in the App Store It Controls". The New York Times, Sept. 9. https://www.nytimes.com/interactive/2019/09/09/technology/apple-app-store-competition.html

[26]It is plausible that the search algorithm change might be due to continuous public complaints of perceived unfair high rankings of Apple's apps on the side of independent developers. I argue that the independent developers would not know when the search algorithm change would come even if they knew it would come.

[27]The change is even more evident among non-preinstalled Apple's apps, as shown in Figure G.3.

change, while the apps in the categories without Apple's apps are not. Table 1.3 shows the summary statistics on observations in categories with and without Apple's apps, before and after the search algorithm change. Panel A shows that the downloads of both groups of apps are decreasing over time. A simple difference-in-differences estimate using the average downloads implies that the downloads of apps in categories with Apple's apps decrease less by 0.01 million after the algorithm change relative to the downloads of apps in categories without Apple's apps. However, there is a sizable deviation within each group and period, and it is not ensured that the two groups are comparable.

Table 1.3: Summary Statistics Before and After the Search Algorithm Change

	Categories with Apple				Categories without Apple			
	Before Jul.2019		After Jul.2019		Before Jul.2019		After Jul.2019	
Variable	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Panel A. App Data								
Downloads	0.07	0.22	0.06	0.26	0.06	0.23	0.04	0.14
Price	1.95	3.74	2.00	3.67	1.75	3.77	1.90	3.89
Update Level	0.40	0.48	0.39	0.49	0.36	0.45	0.33	0.45
Average Rating	4.32	0.62	4.30	0.63	4.43	0.52	4.40	0.58
File Size	98.49	143.90	93.01	139.30	302.80	503.80	302.60	512.50
#Screenshots	5.60	1.94	5.68	1.96	5.72	1.81	5.83	1.87
Description Length	2.40	1.05	2.38	1.06	2.06	1.00	2.03	1.00
Search Ranking|Ever Top 50	24.08	11.71	23.95	11.82	24.07	11.73	24.20	11.67
Obs with search ranking	1,552		3,103		2,266		4,651	
Obs(app/category/month)	2,709		5,784		3,729		8,219	
Panel B. Benchmark Conversion Rates Data								
Rates	0.06	0.05	0.09	0.09	0.04	0.03	0.06	0.06
Among Paid Apps?	0.35	0.48	0.39	0.49	0.21	0.41	0.28	0.45
Obs(type/category/month)	46		99		58		127	

To capture effects on consumers' choice conditional on search, I augment the main data with benchmark conversion rates data from AppTweak, which is presented in Panel B of Table 1.3. For each combination of type/category/month, the data covers average benchmark rates of downloads over impressions (being seen by consumers) for the type of apps in the category across days in the month.[28]

[28]"The benchmarks are calculated based on data from thousands of apps that have synced their iTunes Connect

1.3.2 Difference-in-Differences Analysis

I use difference-in-differences models to study the causal effects of the search algorithm change on independent apps. I define treatment groups as independent apps in categories with Apple's apps and define control groups as independent apps in categories without Apple's apps. Following the literature, to estimate the average treatment effect on independent apps, I use the following two-way fixed effects specification of the difference-in-differences model:

$$y_{jgt} = \beta(AppleCompetitor_{jg} \times Post_t) + \lambda_{jg} + \lambda_t + \nu_{jgt} \tag{1.1}$$

where $AppleCompetitior_{jg}$ is the indicator for the app j is in a category g with Apple's Apps; $Post_t$ is the indicator for the month t is after July 2019. The coefficient on the interaction term, β, captures the average treatment effect of the search algorithm. The specification includes app-category fixed effects denoted by λ_{jg} and month-fixed effects denoted by λ_t. The idiosyncratic error term ν_{jgt} is assumed to be orthogonal to the other variables on the right-hand side of the equation. I am interested in a variety of outcomes of independent apps, including search rankings, downloads, and various app characteristics, each of which is denoted by y_{jgt} in logarithm.

For motivating the demand model, I am particularly interested in the effects on conversion rates.[29] An essential question for modeling the effect of search ranking is whether search ranking directly affects the product values or not. Inspired by Ursu (2018), I interpret conversion rates as the conditional probability for one app to be chosen by a consumer conditional on the consumer has searched it. Since this conditional choice behavior happens after search, it should only be affected by the attributes that determine product values, not by attributes that only affect search costs. Therefore, if conversion rates are significantly affected by the search algorithm change, then there is evidence that search rankings affect product values.

The essential assumption of the difference-in-differences approach is the parallel pre-trend assumption: treatment groups and control groups are comparable before the treatment. To test the assumption empirically, I run the following dynamic specifications:

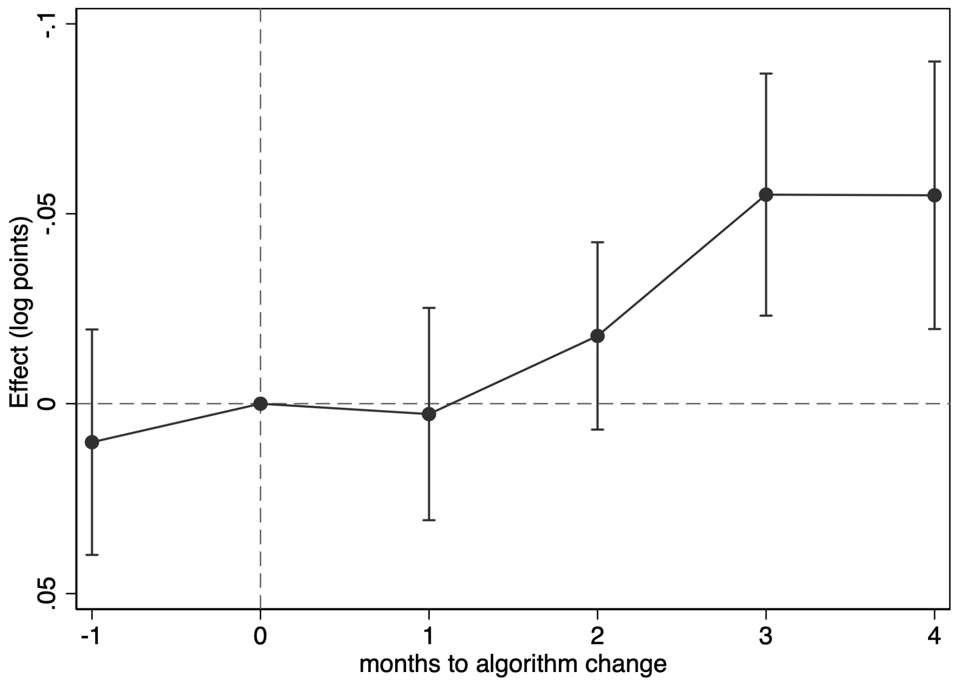
$$y_{jgt} = \sum_{\tau=-1}^{4} \beta_\tau(AppleCompetitor_{jg} \times 1\{t = \tau\}) + \tilde{\lambda}_{jg} + \tilde{\lambda}_t + \tilde{\nu}_{jgt} \tag{1.2}$$

and Google Console with AppTweak.". See Cruyt,Marie-Laure. 2019. "Introducing App Conversion Rate Benchmarks". AppTweak. Jan 21. https://www.apptweak.com/en/aso-blog/mobile-app-store-conversion-rate-benchmarks-per-category. Apart from the plausibility of the benchmark conversion rates, it also indicates that these benchmark conversion rates do not contain the conversion rates of Apple's apps.

[29] When the left-hand-side variable is conversion rates, the index j should be interpreted as type, i.e., paid or free apps, because conversion rates are observed at type/category/month level. Similarly, the fixed effects are type/category fixed effects and month-fixed effects.

where τ is the months to the search algorithm change, and $\tilde{\lambda}_{jg}$ denotes app-category fixed effects, $\tilde{\lambda}_t$ denotes month-fixed effects; $\tilde{v}_{jgt}$ is assumed to be exogenous idiosyncratic error terms. If β_τ is significantly different from zero for $\tau < 0$, then the parallel pre-trend assumption is violated.

Figure 1.2: Test Parallel Pre-trend Assumption: Search Rankings of Independent Apps



Notes. The figure presents point estimates of the effects of the search algorithm on search rankings of independent apps across months.Error bars indicate 95% confidence interval using standard errors robust to heteroscedasticity.

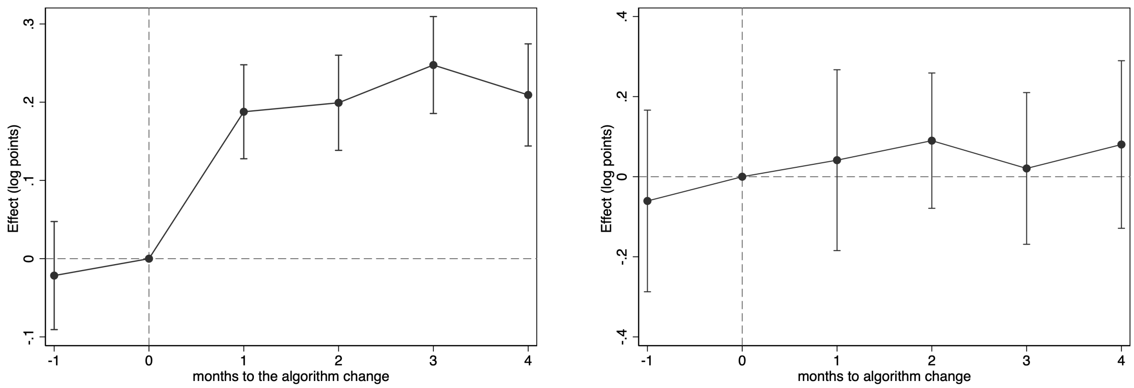
Figure 1.2 shows the verification result for search rankings of independent apps. The vertical axis is estimated β_τ in Equation (1.2) for outcome variable log(search ranking), the horizontal axis is months to the search algorithm change. It does not reject the assumption that the treatment and control groups have a parallel trend in search rankings before the search algorithm change. It also shows that the treated apps see a rising in search rankings after the search algorithm change relative to the controlled apps, confirming the realization of the search algorithm change.

Figure 1.3 shows the verification results for downloads (Panel A) and conversion rates (Panel B) in logarithms. I do not find evidence of violating the parallel pre-trend assumptions for neither of the outcome variables. It also illustrates that the installations of Apple-competing independent apps immediately increase after the search algorithm change relative to the other independent apps in categories without Apple's apps. However, there is no significant change in the conversion rates. It indicates that the search algorithm change increases both the number of consumers who see the independent apps and the number of consumers who download the independent apps, resulting in

Figure 1.3: Test Parallel Pre-trend Assumption: Downloads and Conversion Rates of Independent Apps

Panel A. Download

Panel B. Conversion Rate



Notes. The figure presents point estimates of the effects of the search algorithm on downloads(panel A) and conversion rates(panel B) of independent apps across months.Error bars indicate 95% confidence interval using standard errors robust to heteroscedasticity.

constant conversion rates before and after the search algorithm change. Based on the interpretation of conversion rates as conditional choice probability, the results motivate me to assume that search rankings only affect search costs and do not affect product values in the demand model.

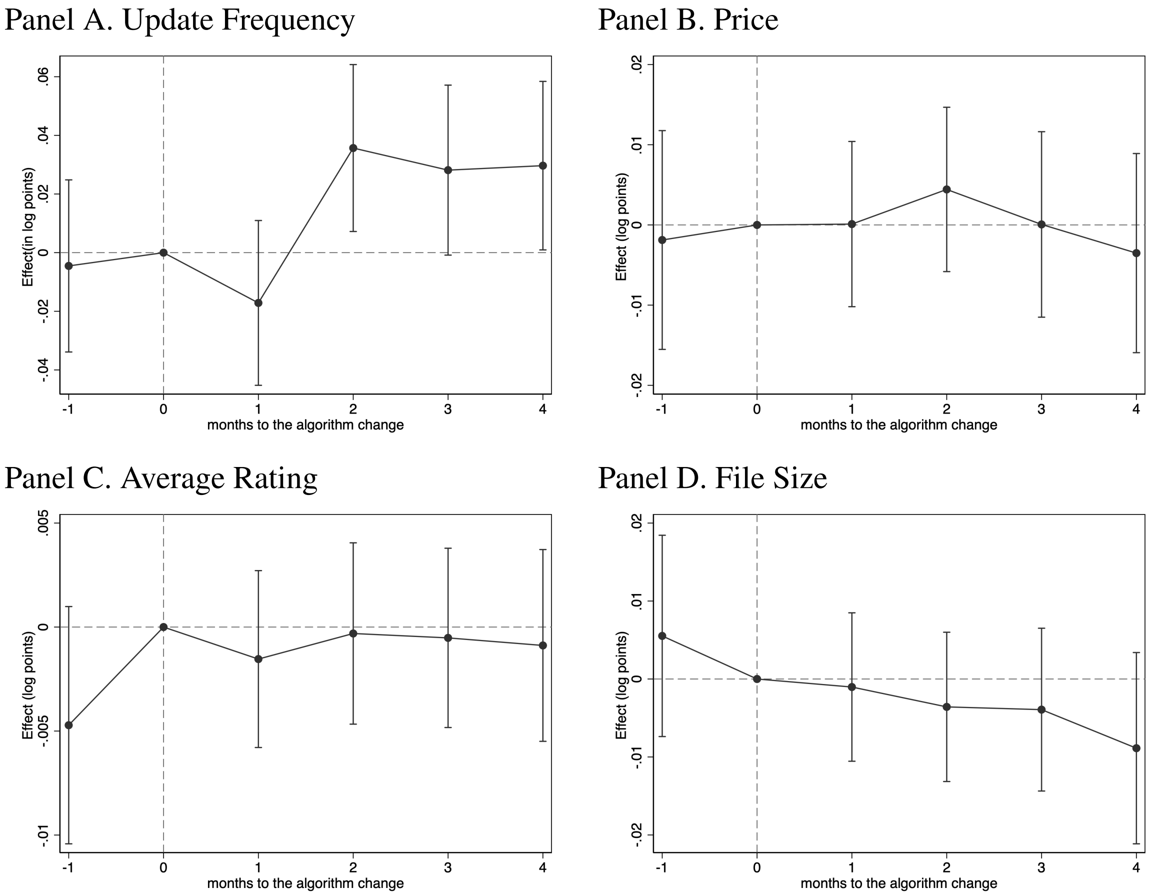
Figure 1.4 shows the verification results for product characteristics including update frequency (Panel A), price (Panel B), average ratings (Panel C) and file size (Panel D) in logarithms. I do not find evidence of violating the parallel pre-trend assumptions for any of the outcome variables. It also shows that update frequencies of Apple-competing independent apps significantly increased in the second month after the search algorithm change.[30] At the same time, there was no significant change in price, average rating, and file size.[31] The results are consistent with the summary statistics on the over-time variation of product characteristics within apps in Table 1.1. It motivates me to focus on update frequencies as developers' primary strategic response to self-preferencings in the supply-side model.

The estimated average treatment effects from Equation1.1 are reported in Table 1.4. It shows that with lower search rankings of Apple's apps after the search algorithm change, independent apps' search rankings significantly rose by 3.6%, installations significantly increased by 22.1%, and update frequencies significantly increased by 2.1%.

[30]While people might think that the higher update frequencies in the second month after the search algorithm change (Sep. 2019) is due to the release of new iPhones in September every year, I do not find a significant increase in update frequencies in September 2018.

[31]I think that the 90% significance of the negative ATE on file size found in Table 1.4 is mainly due to the overall downward-sloping trend of file sizes shown in Figure 1.4 Panel D.

Figure 1.4: Test Parallel Pre-trend Assumption: Product Characteristics of Independent Apps



Notes. The figure presents point estimates of the effects of the search algorithm on characteristics of independent apps across months.Error bars indicate 95% confidence interval using standard errors robust to heteroscedasticity.

The descriptive evidence suggests that self-preferencing is influential: lower search rankings of Apple's apps positively affect independent apps' installations and update frequencies.[32] To identify the self-preferencing and quantify its equilibrium welfare effects, I develop an empirical model for consumer search and update competition in the mobile application markets.

1.4 Modeling Demand and Search Ranking

This section presents empirical models of (1) demand for mobile applications that explicitly incorporates consumer search, and (2) approximates search ranking algorithm with potential self-

[32]Figure G.4 shows that there was not significant effect on entry of competing independent apps due to the search algorithm change.

Table 1.4: Effects of the Search Algorithm Change on Independent Apps: Difference-in-Differences Estimates

Outcome Variable	Estimated ATE	SE	Obs	Adj. R^2	FE	Mean Level
log(Search Ranking)	−0.036***	0.010	11,642	0.86	A	24.17
log(Downloads)	0.221***	0.021	20,423	0.95	A	0.055
log(Conversion Rates)	0.088	0.068	330	0.95	B	0.065
log(1+Update Frequency)	0.021 **	0.009	20,423	0.62	A	0.63
log(1+Price)	0.001	0.004	20,423	0.98	A	1.91
log(Avg.Rating)	0.002	0.002	20,423	0.94	A	4.37
log(File Size)	−0.007 *	0.004	20,423	0.99	A	216.30

Notes: Estimated ATE is the estimate of β in Equation 1.1 for the outcome variable on the row. FE: (A) app/category FE, month FE; (B) type/category FE, month FE. Mean level is not in logarithms. SE is robust standard errors. *** p<0.01, ** p<0.05, * p<0.1.

preferencing.

1.4.1 Demand

I use a random-coefficient discrete choice model with optimal sequential consumer search to describe mobile application demand. A defining feature of mobile application markets is the considerable number of products. For example, in the year 2019, there are 3.96 million apps available on Apple App Store.[33] Therefore, search is a primary way for consumers to find and download an app. Specific to the empirical context of this study, Apple reports that 70% of App Store visitors use search to find their next app, and 65% of all downloads happen after search during 2020.[34] The demand model describes how consumers search and download apps in mobile application markets. I firstly define the market, then explain the indirect utility of downloading an app, and lastly explain the search costs to find an app.

As mentioned in Section 1.2, I define a market as a category/month pair. That is, I assume a consumer may download apps from multiple categories in a month, but within a category, the consumer downloads no more than one app in a month.[35] Then, the market share of an app j that

[33]Statista. 2021. "Number of available apps in the Apple App Store from 2008 to 2021". https://www.statista.com/statistics/268251/number-of-apps-in-the-itunes-app-store-since-2008/

[34]Apple. 2021. "Be discovered." https://searchads.apple.com/

[35]Choice among categories is beyond the scope of this paper. For examples of studies that consider choice among categories, Ghose and Han (2014) uses a random-coefficients nested logit model to incorporate consumers' choices of categories, assuming that a consumer downloads no more than one app on an app store on a day. In addition, Fershtman, Fishman and Zhou (2018) proposes a search model where consumers choose which categories to search,

operates in a category g in month t, is given by $s_{jgt} := Q_{jt}/M_t$, where Q_{jt} is the estimated total downloads of app j in month t and M_t is the number of iPhone users in month t.[36]

Indirect Utility. By downloading an app j in category g and month t, a consumer i receives the following indirect utility:

$$\begin{aligned} u_{ijgt} &= \alpha p_{jt} + x_{jt} \cdot \beta + \tilde{\gamma}_i a_{jt} + \xi_{jgt} + \varepsilon_{ijgt}, \qquad j \neq 0 \\ u_{i0gt} &= \beta_0 I_g + \varepsilon_{i0gt} \end{aligned} \tag{1.3}$$

where p_{jt} is the installation price of app j in month t; a_{jt} is $log(1 + \text{update frequency})$ of app j in month t with a mean-zero random coefficient $\tilde{\gamma}_i$[37]. Hereafter, I call $log(1 + \text{update frequency})$ as update level for exposition. ξ_{jgt} is a partially unobserved (to researchers) quality shifter that are known to consumers before search; x_{jt} is a vector of app characteristics[38]. Before search, all of these app characteristics are known by consumers.[39] Consumers only do not know the idiosyncratic match value, ε_{ijgt}, which they pay search costs to learn. The search costs may include the cost of scrolling down the iPhone screen to see the app, clicking on the app, and checking its detailed information on the view page. Therefore, search costs depend on the search rankings, which will be specified later. I assume that ε_{ijgt}s are i.i.d. T1EV distributed. Consumers know the distribution of the match values.

As for the value of outside options, u_{i0gt}, it depends on whether there are pre-installed apps in the category g, captured by the indicator I_g. In the case of no pre-installed apps, the consumers know their idiosyncratic match values with the outside option (i.e., downloading no apps). When there are pre-installed apps, they are modeled as the outside options, and the consumers still need to search for their idiosyncratic match value with the pre-installed apps. Such search costs may happen when consumers do not know which apps are pre-installed or where are the pre-installed apps on the phone, or whether they have deleted the pre-installed apps or not.

To capture the continuous effects of updates on app qualities over time, I model updates as "incremental innovation" that contributes to the time series of quality improvement instead of one-

---

and firms respond to such more targeted search by strategically choosing the categories in which to list their products. In this paper, the choices of categories on both the demand and supply sides are taken as exogenous.

[36]Following the definition of market shares, apps that operate in multiple categories simultaneously will have the same market shares in multiple category/month pairs. In the data, there are 30% observations from apps that simultaneously operate in more than one category. These observations mostly come from game categories, some from two closely related non-game categories: news and newsstand. On average, an app operates in 1.3 categories at the same time.

[37]Given the large maximum update frequency(11 weighted updates), I put the logarithm function on update frequency to introduce a concave relationship between update frequency and indirect utility.

[38]The vector of app characteristics, x_{jt}, includes month fixed-effects, average rating, an indicator of owned by Apple, an indicator of paid installation, an indicator of offering in-app-purchase, the logarithm of age (in month), the logarithm of file size (in MB), $\log(1 + \text{length of description})$, and indicator of game apps.

[39]Appendix B.1 shows descriptive evidence for consumers at least knowing the observed app characteristics to some extent before search.

shot quality shifters. As in Leyden (2019), quality improvement is modeled as an AR(1) process given by

$$\xi_{jgt} = \rho \xi_{jgt-1} + \gamma a_{jt} + \eta_{jgt}, \qquad \mathbb{E}[\xi_{jgt-1}\eta_{jgt}] = 0 \tag{1.4}$$

where η_{jgt} is unobserved(to researchers) quality shifters. It is assumed to be orthogonal to the lagged partially unobserved quality ξ_{jgt-1}. For example, advertising is unobserved in the data but could affect consumers' perceived value for an app. Notice that there is no random coefficient on updates in Equation (1.4). The heterogeneous consumer tastes over updates are captured by $\tilde{\gamma}_i$ in Equation (1.3), which is assumed to follow the normal distribution $\mathcal{N}(0,\sigma)$. The taste heterogeneity may happen when some consumers like the availability of new features while others dislike the inconvenience of updating the app.

Lastly, I define app j's mean-utility relative to the outside option with the following equation:

$$\delta_{jgt} := \alpha p_{jt} + x_{jt}\beta + \xi_{jgt} - \beta_0 I_g. \tag{1.5}$$

<u>*Search Costs.*</u> Consumers behave as described in the optimal sequential search model (Weitzman, 1979) to search and download apps. Specifically, consumers search apps in the order of reservation values and stop searching when the highest realized utility so far is above the reservation value of the next product to be searched. To have a closed-form choice probability for this search problem, I borrow a distributional assumption on search costs from Moraga-González, Sándor and Wildenbeest (2018). In particular, I assume that the search cost of consumer i to find an app j in market m, c_{ijm}, follows the cumulative distribution function given by

$$F^c_{jm}(c|\mu_{jm}) = \frac{1-\exp(-\exp(-H_0^{-1}(c)-\mu_{jm}))}{1-\exp(-\exp(-H_0^{-1}(c)))} \tag{1.6}$$

where $H_0(r) = \textit{Euler Constant} - r + \int_{\exp(-r)}^{\infty} \frac{\exp(-t)}{t} dt$; and μ_{jm} is the app-market specific location parameter of the search cost distribution.[40] I call the parameter μ_{jm} as *search cost parameter*, which will be determined by the search ranking of app j in market m. Because μ_{jm} has to be positive to give a well-defined distribution function, following Moraga-González, Sándor and Wildenbeest (2018), the specification of μ_{jm} is given by

$$\mu_{jm} = \log\left[1 + \exp(\lambda_1 E_{jm} + \lambda_2 E_{jm} \log(\textit{ranking}_{jm}))\right] \tag{1.7}$$

where $E_{jm} = 1\{\textit{ever top50 in search results}\}$, and $\textit{ranking}_{jm}$ is the search ranking of app j in mar-

[40] To give some explanations of the search cost distribution function, as noted in Moraga-González, Sándor and Wildenbeest (2018), the function allows a mass of consumers to have zero search costs: $F_j(0) = \exp(-\mu_j)$. Similarly, the sequential search model in Stahl (1989) also allows consumers to have zero search costs.

ket m conditional on ever top50 in search results.[41] Intuitively, search costs will be lower for apps that are ranked in top50 search results($E_{jm} = 1$) and have higher (smaller) rankings ($ranking_{jm}$). Therefore, I expect λ_1 to be negative and λ_2 to be positive. Notice that the two search cost shifters only show up in Equation (1.7), and do not show up in Equation (1.3). It assumes that search rankings only affect search costs and do not affect product values. This assumption is supported by the descriptive evidence that conversion rates are not affected by the search algorithm change in July 2019, as shown and explained in Section1.3. It is the crucial assumption for separate identification of product quality and search cost.

Following the Proposition 1 in Moraga-González, Sándor and Wildenbeest (2018), the distributional assumption on search costs in Equation (1.6) rationalizes a closed-form choice probability. Specifically, the probability for consumer i downloading app j in market m is given by

$$s_{ijm}(\theta^D, \tilde{\gamma}_i) = \frac{\exp(\delta_{jm} + \tilde{\gamma}_i \tilde{a}_{jt(m)} - \mu_{jm} + I_g \mu_{0m})}{1 + \sum_{l \in \mathscr{J}_m} \exp(\delta_{lm} + \tilde{\gamma}_i \tilde{a}_{lt(m)} - \mu_{lm} + I_{g(m)} \mu_{0m})} \tag{1.8}$$

where $t(m)$ is the month of market m and $g(m)$ is the category of market m, $\mathscr{J}_m$ is the set of apps in market m. θ^D denotes the vector of consumer preference parameters, containing $(\alpha, \beta, \rho, \gamma, \sigma, \lambda)$. Then, I aggregate the consumer-level choice probability to market level to obtain the market share of app j in market m as below:

$$s_{jm} = \int s_{ijm}(\theta^D, \tilde{\gamma}_i) dF_{\tilde{\gamma}}(\tilde{\gamma}_i), \qquad \tilde{\gamma} \sim \mathscr{N}(0, \sigma). \tag{1.9}$$

Because the choice probability belongs to the Berry, Levinsohn and Pakes (1995)(BLP) framework, I apply the BLP inversion to back out the search-cost-augmented relative mean-utility, $\tilde{\delta}_{jm}$, given a guess of σ. It equals the relative mean-utility in Equation (1.5) net of the search cost parameter relative to the outside option's search cost parameter, which is given by

$$\tilde{\delta}_{jm}(s_m; \sigma) = \alpha p_{jt(m)} + x_{jt(m)} \beta + \xi_{jm} - \beta_0 I_g - \mu_{jm}(\lambda) + I_{g(m)} \mu_{0m}(\lambda) \tag{1.10}$$

where s_m is the vector of market shares in market m. Substituting Equation (1.4) into the above

[41] Recall that a market is a category-month pair. For an app that simultaneously operates in multiple categories, its search rankings might differ across categories because the category-specific popular keywords are different. For example, one app in the data is operating in both the News category and the Newsstand category in April 2018. The keyword "politics" was a popular keyword in the News category but not in the Newsstand category. In contrast, "magazines" was a popular keyword in the Newsstand category but not in the News category. The app gets ranked at position 14 in the search results for "politics" and ranked 10 in the search results for "magazines" on the same day. The difference causes the app to have different search rankings in the News category and the Newsstand category in the same month.

equation, I obtain the main estimation equation for the demand model, which is given by

$$\tilde{\delta}_{jgt}(s_{gt};\sigma) = \alpha p_{jt} + x_{jt}\beta + \rho\xi_{jgt-1} + \gamma a_{jt} - \beta_0 I_g - \mu_{jgt}(\lambda) + I_g\mu_{0gt}(\lambda) + \eta_{jgt} \tag{1.11}$$

Notice (ρ,λ) are non-linear parameters in Equation (1.11), since ξ_{jgt-1} is unobserved and $\mu_{jgt}(\cdot)$ is non-linear. To speed up estimation, I get around the non-linearity of ρ by subtracting $\rho\tilde{\delta}_{jgt-1}$ from $\tilde{\delta}_{jgt}$, which returns

$$\tilde{\delta}_{jgt}(s_{gt};\sigma) - \rho\tilde{\delta}_{jgt-1}(s_{gt-1};\sigma) = \alpha\dot{p}_{jt} + \dot{x}_{jt}\beta - \beta_0(I_g - \rho I_g) - \dot{\mu}_{jgt}(\lambda) + I_g\dot{\mu_{0gt}}(\lambda) + \gamma a_{jt} + \eta_{jgt}$$

where $\dot{y}_t = y_t - \rho y_{t-1}$. This equation can be estimated with linear regression, given a guess of (σ,ρ,λ). The estimation of the demand model is based on the General-Methods-of-Moments(GMM) with iterated guesses of (σ,ρ,λ). The moments are constructed based on Equation (1.11).

1.4.2 Search Ranking

Taking the estimated app qualities from the demand model as input, I use a flexible search ranking model to identify self-preferencing for Apple's apps.

Apple's search ranking algorithm is proprietary, however I approximate it using a rank-ordered logistic regression model(Beggs, Cardell and Hausman, 1981). The advantage of the rank-ordered logistic model is that it allows correlation between rankings of products in the same market. Imagine a regression equation where search ranking is on the left-hand side, and an additively separable error term is on the right-hand side. Such a regression typically needs to assume that the error terms are independent across observations. In contrast, instead of assuming idiosyncratic error terms directly on search rankings, the model assumes such error terms for a latent variable, ranking score. Then, the products in the same market are ranked according to the random ranking scores, allowing intuitive correlations between search rankings of products in the same market. For example, if the probability of one product being ranked top1 increases, the model predicts that the other products' probability of being ranked top1 decreases.

To explain how I use the model to capture self-preferencing, I start by specifying the ranking score of an app j in market m as below

$$score_{jm} = \sum_{\tau=1}^{\tau=23} \theta_{1,\tau} Apple_j * 1\{t(m) = \tau\} + \theta_2\check{\delta}_{jm} + \theta_3\check{\delta}_{jm}^2 + z_{jm}^s \cdot \vartheta + e_{jm} \tag{1.12}$$

where $\theta_{1,\tau}$ captures the preferential treatment for Apple's apps in month τ; $\check{\delta}_{jm}$ is estimated app quality from the demand model. If there is self-preferencing for Apple's apps in month τ, then

$\theta_{1,\tau}$ will be significantly positive. The idiosyncratic unobserved error terms e_{jm} are i.i.d. T1EV distributed. It may be variation in consumers' usage of apps, un-installations, and retention that is independent of the right-hand side observables. To ensure this assumption, I control for category-specific and month-specific effects of well-accepted search ranking shifters in z^s_{jm}, including price, payment type, lagged ratings, and text relevance, apart from app quality.[42] Appendix B.2 lists the variables in z^s_{jm}. [43]

This specification can directly verify the typical argument for observed high search rankings of Apple's apps on Apple App Store: "Apple's apps are preferred by consumers" because it explicitly includes app quality. Specifically, I define quality, $\check{\delta}_{jm}$, as product value net of installation payment-relevant terms:

$$\check{\delta}_{jm} := \delta_{jm} - \alpha p_{jt(m)} - \beta_1 paid_j \tag{1.13}$$

where β_1 is the coefficient on the paid app indicator in the demand model. During estimation of the search ranking model, the demand estimates of product value(δ_{jm}), price coefficient(α), and paid-app coefficient(β_1) will be used to construct app quality($\check{\delta}_{jm}$). Notice that $\check{\delta}_{jm}$ contains estimated consumers' preference for Apple's apps over independent apps. To improve fitness, I include both the linear and quadratic terms of app quality in the score equation (1.12).

The search ranking model is estimated with Maximum-Likelihood-Estimation(MLE) for the probability of the observed orderings of products. To write down the conditional log-likelihood, I firstly write down the conditional probability of an app j to be ranked top1 in market m:

$$pr_{jm}(\theta|x^s_m) = \frac{\exp(x^s_{jm} \cdot \theta^s)}{\sum_{l \in \mathscr{J}_m} \exp(x^s_{lm} \cdot \theta^s)} \tag{1.14}$$

where $x^s_m = (x^s_{1m}, x^s_{2m}, \cdots, x^s_{J_m m})$, $x^s_{jm} = (Apple_j, \check{\delta}_{jm}, z^s_{jm})$, $\theta^s = (\theta_{1,1}, \cdots, \theta_{1,23}, \theta_2, \theta_3, \vartheta)$, and there are J_m apps in market m.

Let y_m denote the ordering of the J_m products in market m, where $y_m(k)$ is the product that is ranked at position k in market m. Then the conditional log-likelihood for observing the orders of

[42] I think these variables capture quite some important factors, since they match with what Apple says about their search algorithm: "Apple has agreed that its Search results will continue to be based on objective characteristics like downloads, star ratings, text relevance, and user behavior signals.". See Apple. 2021. "Apple, US developers agree to App Store updates that will support businesses and maintain a great experience for users". August 26. https://www.apple.com/newsroom/2021/08/apple-us-developers-agree-to-app-store-updates/

[43] Notice that there is no constant term in the score equation because only within-market relative ranking matters for identification, the exact values of rankings do not matter.

products in the data is given by

$$\begin{aligned} L(\theta|x^s) &= \sum_{m=1}^{M} \log \mathbb{P}[y_m|x^s_m] \\ &:= \sum_{m=1}^{M} \log \left[pr_{y_m(1)} \cdot \frac{pr_{y_m(2)}}{1 - pr_{y_m(1)}} \cdot \dots \cdot \frac{pr_{y_m(J-1)}}{pr_{y_m(J-1)} + pr_{y_m(J)}} \cdot \frac{pr_{y_m(J)}}{pr_{y_m(J)}} \right] \end{aligned} \tag{1.15}$$

where the term in the bracket in the second line is the product of the probability for the ranked-first product to be ranked in position 1, and the probability for the ranked-second product to be ranked in position 2 conditional on that the ranked-first product is ranked in position 1, and the probability for the ranked-third product to be ranked in position 3 conditional on that the top2 products are ranked before it, and so on until the probability for the ranked-last product to be ranked in the last position conditional on that all the other products are ranked before it (which is equal to 1).

1.5 Estimation

This section presents the estimation procedures and results of the model. The results point to self-preferencing: while consumers prefer Apple's apps to independent apps, the preference is not large enough to justify the higher ranking of Apple apps compared to independent apps.

1.5.1 Estimation Procedures

Demand. The estimation of demand is similar to Berry, Levinsohn and Pakes (1995). I construct moments using Equation (1.11) and estimate the parameters using GMM. However, there are richer endogeneity concerns in this paper. Specifically, update level(a_{jm}) is endogenous because developers know the unobserved demand shocks (η_{jm}) when choosing update levels. Moreover, price, search ranking, and average ratings may also be correlated with the unobserved demand shocks. For example, advertising is unobserved and might be correlated with the above app characteristics and directly affects demand.

For the endogeneity concerning app characteristics, following the literature, I firstly construct basic instruments using the characteristics of other apps owned by the same developer in the other categories in the same month. These basic instruments follow the idea of the Hausman-type(Hausman and Bresnahan, 2008) instrument: common unobserved cost shifters among products owned by the same firm. Then, in the case of single-category developers, I construct instruments based on the basic instruments. In particular, I calculate the average basic instrument values of the apps of the competing developers in the same market. These instruments are similar to BLP instruments – they are characteristics of the apps of competing developers. However, they are more indirect than BLP instruments because these characteristics are based on other apps in other markets.[44] The idea is that when a developer chooses app characteristics, s/he considers the cost features of her/his competitors, which are partially captured by their behaviors in other markets.

As for the endogeneity in search rankings, I construct instruments based on the title match of the app with popular keywords in the market. The above estimation strategy relies on three assumptions: i) markets are independent; ii) the unobserved demand shocks are realized after app entry (but before update choices); iii) titles do not directly affect app values. The first two assumptions are commonly used in the literature. As for the third assumption, I control for systematic time effects using month-fixed effects. Therefore it seems reasonable that any app/category/month-specific shocks are uncorrelated with app titles. In addition to the above instruments, I include interaction terms between the post-July-2019 indicator and i)Apple-ownership indicator and ii)Apple-competitor indicator in the instruments.[45] These instruments are based on the exogenous search algorithm change in July 2019 that particularly affects categories with Apple's apps. The first-stage regression results for the above instruments are shown in Appendix B.4.

Search Ranking. The estimation of the search ranking model is based on MLE, where the log-likelihood function is given in Equation (1.15). Because the score equation (1.12) is quite flexible with category-specific and month-specific coefficients, I argue that there is no obvious endogeneity concern.

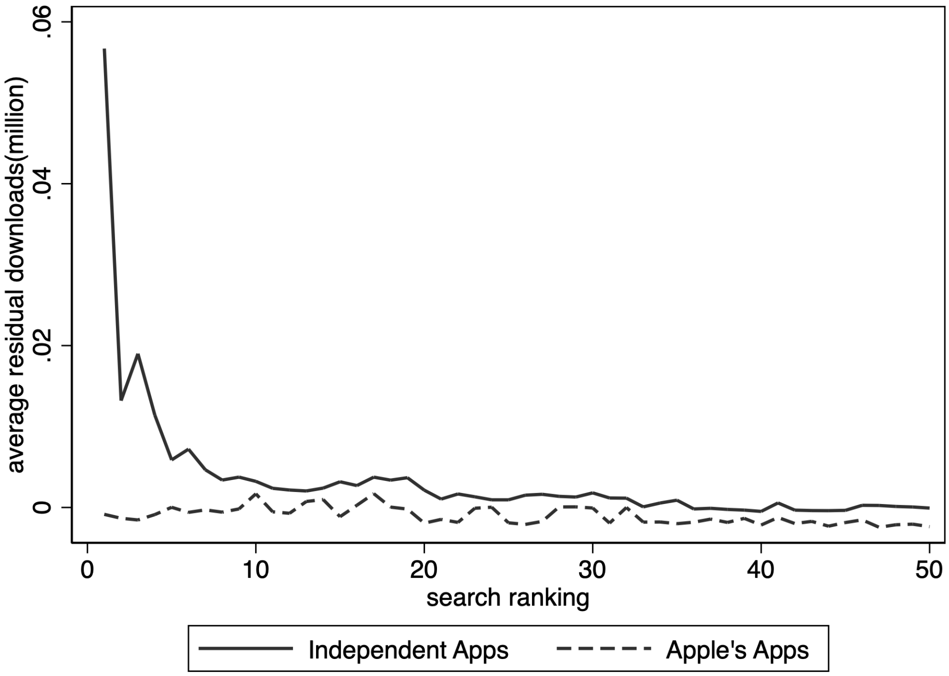
Now I explain the intuitions for identifying the self-preferencing for Apple's apps in more detail. Given the estimated qualities, I identify preferential treatments on Apple's apps from the variation of within-market relative search rankings that are independent of quality, where the variation is across Apple's apps and independent apps. For example, if apps A and B show up in the same market multiple times, and A is frequently ranked higher than B, then app A must have a higher mean-ranking-score, i.e., $\overline{score}_{jm}$, than app B. When both A and B are independent apps, if A has higher quality than B, then the effect of quality on the ranking score is identified as positive. Now, without loss of generality, let us suppose that the identified effect of quality on the ranking score is positive. Then when A is an Apple's app, and B is an independent app with higher quality, it is identified that the higher ranking of A than B is due to Apple's ownership, which is the preferential treatment for platform-owned products.

The identification intuition can be roughly illustrated in Figure 1.5. It shows that independent apps need much higher average residual daily downloads than Apple's apps to reach the same position in the search results for the same keyword. Here, the residual daily downloads are resid-

[44]Because update levels are endogenous, I cannot use classical BLP-type(Berry, Levinsohn and Pakes, 1995) instruments (e.g., average update levels of apps of competing developers).

[45]Because category-fixed effects are included in the excluded instruments, the Apple-competitor indicator is absorbed.

Figure 1.5: Independent Apps *v.s.* Apple's Apps: Residual Downloads at Each Search Ranking



Notes. The figure compares the average residual downloads of independent apps and Apple's apps on the same position in the search results for the same keyword across different days. The residual downloads are from residuals from price, installation payment type, category-fixed effects and daily fixed effects.

uals from price, paid app indicator, category-fixed effects, and daily fixed effects. Moreover, the comparison is conditional on the keywords where both independent apps and Apple's apps show up in the search results. Thus, the residual downloads are not confounded by market-level shocks or installment payments. Then, I take the average of the residual downloads across categories, days, and keywords within Apple's apps and independent apps. The illustrated difference between Apple's apps and independent apps suggests preferential treatments on Apple's apps in the search algorithm.[46] The search ranking model exploits such variation in the data to identify and quantify the extent of the self-preferencing for Apple's apps.[47]

[46]Because the ranking score equation does not explicitly control consumer behaviors such as retention rates and un-installations, there might be alternative explanations like "consumers do not download Apple's apps as much, but they use them a lot". I argue that such behaviors shall be reflected in ratings, which are explicitly and flexibly included in the ranking score equation. For example, high retention rates are likely to generate high star ratings, while un-installations are likely to produce low star ratings.

[47]Figure G.5 shows that the gap in residual daily downloads between Apple's apps and independent apps at the same positions is smaller after the search algorithm change in July 2019. This variation is also exploited in the model by the month-specific preferential treatment parameters.

1.5.2 Estimates of Demand

Table 1.5 reports the estimates for parameters of the demand model. The demand estimation results show that an average consumer prefers apps with higher update levels, Apple ownership, higher average ratings, more experiences, larger file size, fewer screenshots, in-app-purchase availability, and lower installation price. For example, an average consumer is willing to pay $5.2 more for downloading an app developed by Apple than one developed by an independent developer. It might justify the observed high search rankings for Apple's apps and is taken into account by the search ranking model. In addition, the estimated standard deviation for consumers' taste for update levels is about 4.6 times the average taste, suggesting consumers are quite heterogeneous in their tastes for update levels. Lastly, the results show that consumers are more likely to incur higher search costs when searching for an app with lower(i.e., larger) search rankings in the top50 search results or an app absent from the top50 search positions.

Table 1.5: Estimation Results for Demand

Variables	Parameter	Standard error
Quality Coefficients		
Update Level	0.155	0.074
Apple	1.115	0.447
Average Rating	0.527	0.172
log(Age) (month)	0.330	0.143
log(File Size) (MB)	0.422	0.054
#Screenshots	−0.039	0.015
log(1 + Description Length)	0.033	0.121
Offer In-App-Purchase	0.230	0.134
Game	−0.037	0.146
Constant	−13.840	1.019
One-month Lagged Unobserved Mean Relative Utility	0.921	0.004
Outside Value: Exists Pre-installed Apps	−0.255	0.134
Price	−0.216	0.046
Paid	−1.754	0.239
Random Coefficients		
Update Level	0.711	0.105
Search Cost Parameter		
Ever Top50 in Search Results	−5.125	2.601
log(Search Ranking)|Ever Top50	1.374	0.737
Month-FE	YES	
Observations	52,959	

Notes: Update level is $\log(1 + \text{update frequency})$. Description length is in the unit of 1,000 characters.

Table 1.6: Demand Semielasticities with respect to Price and Search Ranking

	Netflix	TikTok	Hulu	Amazon Prime Video
	Panel A. Price Semielasticities			
Netflix	−0.205	0.008	0.004	0.003
TikTok	0.010	−0.209	0.004	0.003
Hulu	0.010	0.007	−0.212	0.003
Amazon Prime Video	0.009	0.006	0.003	−0.214
	Panel B. Ranking Semielasticities			
Netflix	−0.174	0.007	0.004	0.002
TikTok	0.008	−0.175	0.003	0.002
Hulu	0.008	0.006	−0.173	0.002
Amazon Prime Video	0.007	0.005	0.003	−0.174

Notes. The top panel reports percentage change in market share of the column-product with a $1 increase in the row-product's installation price. The bottom panel reports percentage in market share of the column-product with a ten-position decline of the column-product's search ranking.

Table 1.6 shows the price semielasticities(Panel A) and search ranking semielasticities(Panel B) for the top four apps in the entertainment category in July 2019. These apps are Netflix, TikTok, Hulu, and Amazon Prime Video. Panel A shows that a $1 increase in the installation price of an app leads to about 0.2% decrease in its demand.[48] Panel B shows that a ten-position decline in the search ranking of an app leads to about 0.17% decrease in its demand. Unsurprisingly, the own price(ranking) semielasticities are larger than the cross semielasticities in magnitude. Dividing the own ranking semielasticities by the own price semielasticities gives an intuitive measurement for position effect: a ten-position decline in search ranking is equivalent to increasing price by $0.85. This position effect is relatively small compared to those found in the other industries. [49] It suggests that some consumers are likely to know their match values with apps before searching (which is allowed by the assumed distribution of search costs), and thus the average search costs are small across consumers. For example, it may happen when some apps advertise a lot out of the App Store. Furthermore, Figure G.9 illustrates an inverse U-shape curve for estimated position effects across search rankings. It indicates i) inelastic demand for apps with high search rankings and ii) high search costs for apps with low search rankings.

Table 1.7 shows the elasticities of update level for the same top four apps in the same market.

[48] I do not compute price elasticity because these four apps are free. The price semielasticities for the top four paid apps in the same market are similar.

[49] For example, in the online hotel industry, Ursu (2018) finds the effect of 1 position decline between $0.55 and $3.19, Chen and Yao (2017) find it to be $0.21, Koulayev (2014) finds it ranging from $2.93 to $18.78.

Table 1.7: Demand Elasticities with respect to Update Level

	Netflix	TikTok	Hulu	Amazon Prime Video
Netflix	1.395	−0.073	−0.038	−0.024
TikTok	−0.073	0.994	−0.026	−0.016
Hulu	−0.068	−0.048	0.954	−0.015
Amazon Prime Video	−0.043	−0.030	−0.016	0.601

Notes. The table reports percentage change in market share of the column-product with a 1 percent increase in the row-product's update level. Update level is $\log(1 + \text{update frequency})$.

Across the four apps, a 1 percent increase in the update level is associated with a 0.6 percent to 1.4 percent increase in market shares. Similarly, the own update level elasticities are larger than the cross elasticities.

1.5.3 Estimates of Self-preferencing

Table 1.8 reports the estimates for parameters of the search ranking model. The results are intuitive: higher-ranked apps are associated with higher quality, lower installation price, more text relevance with popular keywords, and more one-month lagged 5-star ratings. Furthermore, it shows that preferential treatment on Apple's apps exists in the search algorithm of the Apple App Store. Specifically, in June and July 2019, the ranking score of Apple's apps are significantly higher than independent apps by 1.5 and 1.3. To mitigate such a disadvantage in the search algorithm, an average independent app needs to decrease installation price by \$73.9($\approx 1.551/0.021$) in June 2019 or \$60.0($\approx 1.261/0.021$) in July 2019.[50]

Figure 1.6 presents the coefficients on the Apple-ownership indicator in each month. It illustrates that the preferential treatment on Apple's apps evolves through four periods: i) before April 2019, it was weak if any; ii) from April 2019 to June 2019, it starts to increase and reaches the peak; iii) from July 2019 to August 2019, it starts to decrease; iv) after August 2019, it basically disappears.

To understand the four periods of the self-preferencing, I start with the latest two periods: from July 2019 to February 2020. The results are consistent with the search algorithm change studied in Section 1.3. In fact, when normalizing the preferential treatment parameter($\theta_{1,\tau}$) in July 2019

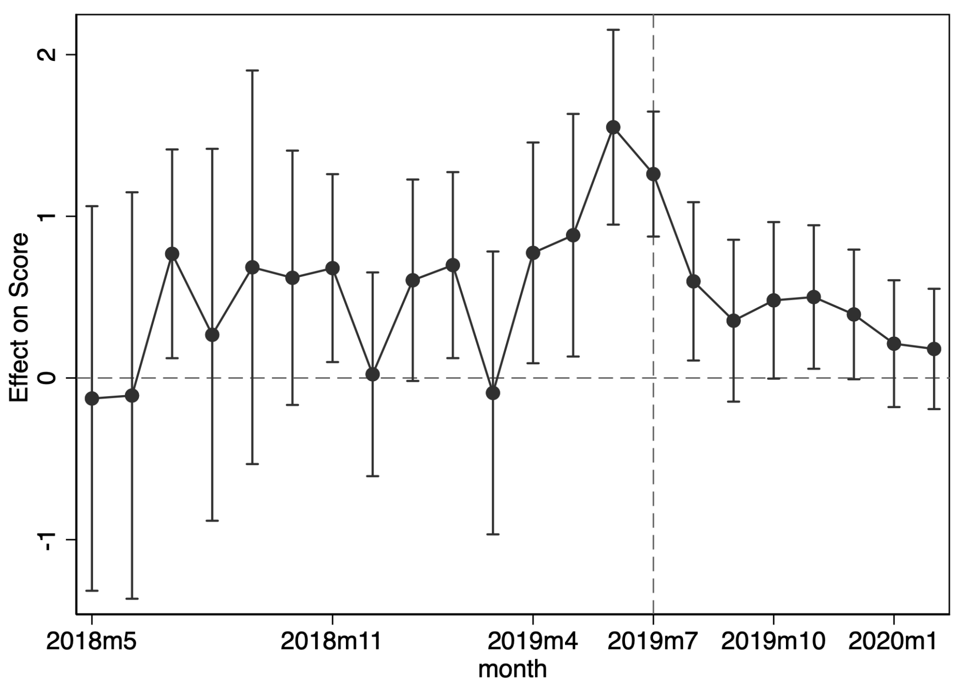
[50]Figure G.10 presents the histogram of static developer-fixed effects across all developers. To estimate the developer-fixed effects, I replace the interaction terms between the Apple-ownership indicator and month indicators in Equation (1.12) with developer-fixed effects, where the single-product developers are normalized as the reference group. Figure G.10 shows that, even when allowing all developers to have their own advantage or disadvantage in the search ranking algorithm, Apple ownership still generates larger advantages than most developers.

Table 1.8: Estimation Results for Search Ranking

Variables	Parameter	Standard error
Apple×1{June 2019}	1.551	0.366
Apple×1{July 2019}	1.261	0.235
Quality	0.167	0.019
Squared Quality	0.002	0.001
Paid	−0.512	0.077
Price	−0.021	0.005
Title Match with Popular Keywords	12.780	1.036
Subtitle Match with Popular Keywords	2.785	1.007
One-month Lagged 5-star Ratings	5.365	0.575
Pre-Install	−0.196	0.635
Full Controls	YES	
Observations	53,245	

Notes: The other interaction terms between Apple and Month indicators are included in the model but not shown here. Appendix B.2 lists variables included in the full controls.

Figure 1.6: Apple self-preferencing Parameters across Months

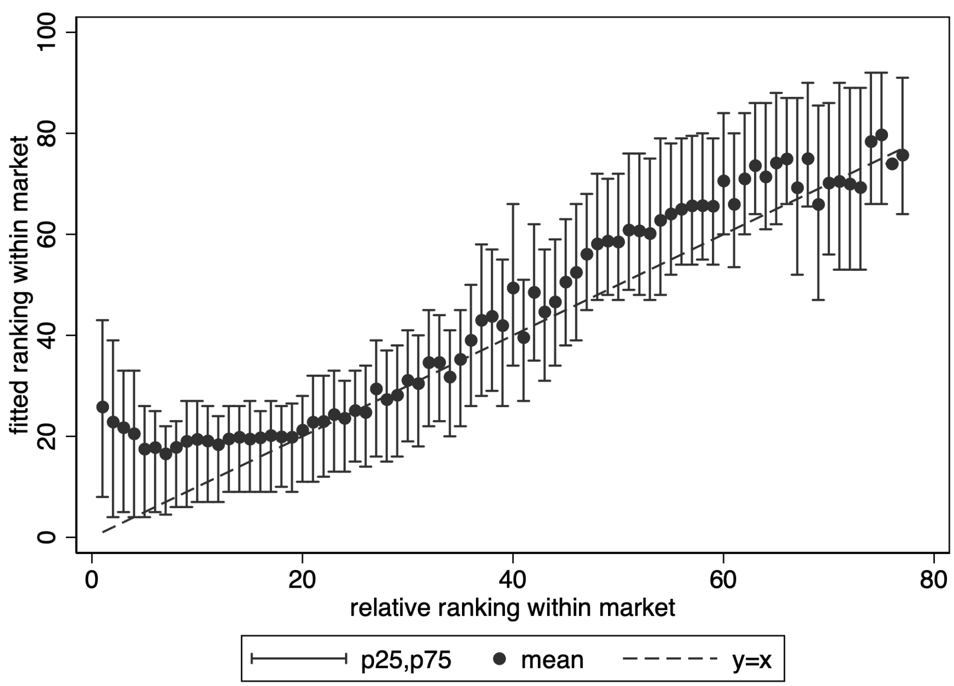


Notes. The figure presents point estimates of the effects of Apple ownership on ranking score in each month during the sample period.Error bars indicate 90% confidence interval using standard errors clustered at the category-month level.

as zero, Figure G.6 shows that the preferential treatment parameters in and after August 2019 are significantly lower than those in June and July 2019. Therefore, the search algorithm change in July 2019 provides good cross-validation for the structural estimates.

The first two periods seem to be inconsistent with the presented average search rankings of Apple's apps in Figure 1.1. Specifically, it suggests that Apple's apps deserve the observed high search rankings during the first period, and as independent apps become stronger competitors for Apple's apps, the self-preferencing kicks in during the second period. This interpretation can be shown by comparing the observed rankings to rankings of residual downloads. In particular, suppose the self-preferencing was stronger in the first period compared to the second period, as suggested by Figure 1.1. In that case, the gap between the observed rankings and rankings of residual downloads should be larger in the first period compared to the second period. However, Figure G.8 shows the opposite. It supports the above interpretation and explains the seemingly inconsistent trends. This discussion suggests the importance of identifying self-preferencing rigorously and the incompleteness of information from only observed rankings.

Figure 1.7: Rank-ordered Logistic Regression Model Fitness



Notes. The figure presents predicted most-likely within-market ranking (y-axis) against observed within-market ranking (x-axis) across markets. Bars indicate the 25 percentile and 75 percentile of fitted within-market rankings across apps that have ranked at the given observed within-market ranking.

Figure 1.7 shows the fitness of the search ranking model. It plots the model-fitted most likely within-market relative rankings against the observed within-market relative rankings.[51] It shows that the model fits the data relatively well. For example, for most of the relative rankings, the average fitted relative rankings are close to the observed ones; and the interval between the first and the third quartiles of the fitted relative rankings covers the observed ones. In addition, Figure G.11 particularly shows the fitness for Apple's apps. App developers' beliefs on possible search rankings are constructed based on the estimated search ranking model.

1.6 Demand-side Effect of Eliminating Self-preferencing

This section uses the estimates from the demand and search ranking models to examine counterfactual simulations. Specifically, I focus on changes in market outcomes when shutting down the identified self-preferencing, *without* endogenous adjustment on the supply side. The changes are used to quantify the demand-side effects of self-preferencing. When shutting down the self-preferencing, the coefficients on the Apple-ownership indicator are set to be zero in the search ranking model.

The counterfactual simulations cover eight markets identified with significant self-preferencing. The markets are the entertainment category, health & fitness category, music category, and utilities category in June and July in 2019. The two months are estimated to have the strongest self-preferencing, i.e., the largest significant coefficients on the Apple-ownership indicator in the search ranking model, during the sample period.

I fix the search costs of outside options throughout the counterfactual simulations. In the model, outside option lumps i) not using any app, ii) using previously downloaded apps, and iii) using pre-installed apps. In the first two cases, there is no search costs. In the last case, search costs for pre-installed apps change with their search rankings. However, the search algorithm change in July 2019 is mainly driven by changes of non-preinstalled Apple's apps' search rankings, which implies the rigidity of pre-installed apps' search rankings.[52] Because the empirical model for estimation does not fix the search rankings of pre-installed apps, for the sake of consistency, the reported status-quo market outcomes with self-preferencing are the results from fixing pre-installed apps' search rankings. For details on the procedures of fixing the pre-installed apps' search rankings, please see Appendix C.2.

Table 1.9 shows the demand-side effects of eliminating the identified preferential treatment on Apple's apps in the search ranking. The first row confirms that eliminating self-preferencings

[51]For example, the search ranking of an app might be 33.5 in a market where there are nine apps whose search rankings are strictly higher (smaller) than 33.5. Then, the within-market relative ranking of this app is 10. Ties are considered in the search ranking model.

[52]Figure G.3 shows the average search ranking of non-preinstalled Apple's apps around July 2019, which sees a more significant change after July 2019 than the average search ranking of all Apple's apps in Figure 1.1.

Table 1.9: Demand-side Effects of Preferential Search Ranking on Search Rankings, Installations and Welfare

	Variable	Status-quo	Shut-down	Mean Δ	Mean $\%\Delta$
(1)	Average Search Rankings	38.92	38.92	0	0
(2)	- Independent apps	39.86	39.23	-0.63	-1.56
(3)	- Apple's apps	14.35	31.85	17.51	142.07
(4)	Total Installations (million)	15.16	15.40	0.24	1.55
(5)	- Independent apps	15.06	15.32	0.26	1.75
(6)	- Apple's apps	0.10	0.08	-0.03	-26.58
(7)	Consumer Surplus (million \$)	321.40	321.94	0.53	0.17
(8)	Variable Profits (million \$)	76.60	77.01	0.41	0.62
(9)	Total Search Costs (million \$)	15.39	15.36	-0.03	-1.51
(10)	Total Realized Utility (million \$)	336.79	337.29	0.50	0.15

Notes: For the status-quo case, there is identified preferential search ranking. For the shut-down case, there is no preferential search ranking. Variable profits are calculated based on independent developers.

does not change average search rankings; it only re-allocates the positions among apps in the same market.[53] The second and third rows show that, in an average market, eliminating the self-preferencing lowers down Apple's apps by 17.5 positions and boosts up independent apps by 0.6 positions in search. [54] Compared to the DiD-estimated average treatment effect on the search ranking of independent apps shown in Table 1.4, i.e., 3.6 percent decrease, the structural change of 1.6 percent is smaller. This reflects that the search algorithm change causes more shifts in search rankings beyond reducing the preferential treatment effect on Apple's apps.[55]

How much of the difference in search rankings between Apple's apps and independent apps is explained by self-preferencing? A back-of-envelope calculation tells us the answer: with self-preferencing, the average difference is 25.51 (=39.86-14.35) position; without self-preferencing, the average difference is 7.38 (=39.23-31.85) position. The difference shrinks by 71 percent,

[53]Therefore, eliminating self-preferencing is a differential change in search costs across products instead of a universal reduction in search costs. In particular, eliminating the identified self-preferencing decreases the search costs for some independent apps while increasing the search costs for Apple's apps. It makes the research question different from most papers in the literature on information frictions and product competition.

[54]It corresponds to a 142 percent decline of search rankings for Apple's apps and a 1.6 percent rise of search rankings for independent apps. The smaller changes in the search rankings of independent apps relative to Apple's apps are due to the large number of independent apps compared to the number of Apple's apps in the markets.

[55]For example, Figure G.7 shows that the effect of lagged number of 5-star ratings became more positive after the search algorithm change. I did not find significant changes of the search ranking parameters on installation price and title match after the search algorithm change. Also recall that the search ranking parameters on quality index are constant over time, currently. In principal, these parameters can change over time and be significantly affected by the search algorithm change.

which is the portion explained by self-preferencing. Also, notice that without the identified self-preferencing, Apple's apps are still ranked higher than independent apps, on average. This is due to consumers' preference for Apple's apps.

Rows four to six in Table 1.9 report the effects on installations. After the elimination, in an average market, the total installations of independent apps increase by 1.8%, while the total installations of Apple's apps decrease by 26.6%. Because Apple's apps only account for a small portion of the products in the markets, the total installation in an average market increase by 1.6%. Again, compared to the DiD-estimated average treatment effect on the downloads of independent apps shown in Table 1.4, i.e., 22.1 percent increase, the structural change of 1.8 percent is much smaller. This is consistent with the previous reflection that the search algorithm change does more than reducing self-preferencing. The rest of changes in the search ranking, together with supply-side adjustment, causes the gap between the two estimates.

Before jumping into the resulting effects on welfare, let me firstly explain the definition of consumer surplus. Thanks to the separation of indirect utility from search cost, the welfare of a consumer in a market is well-defined as the indirect utility of the app chosen by the consumer net of the total search costs incurred by the consumer to find the app in dollars.[56], as shown in the following equation for a consumer i in market m: [57]

$$CS_m := M_m \cdot \mathbb{E}_{(\varepsilon,c,\sigma,y)} \left[-(u_{ij(i)m} - \sum_{l \in \mathscr{S}_i} c_{ilm})/\alpha \right] \tag{1.16}$$

where M_m is the market size of market m, $j(i)$ is the chosen app by consumer i, $\mathscr{S}_i$ is the set of apps that are searched by consumer i, and α is the identified price(income) coefficient in the demand model. The expectation is over i) the vector of consumer-app-specific unobserved match-values (ε), ii) the vector of consumer-app-specific search costs (c), iii) consumer-specific random coefficients over updates (σ), and iv) the vector of app-specific search rankings (y). Changing self-preferencing potentially changes both the chosen app ($j(i)$) and the set of searched apps ($\mathscr{J}_i$). For more details on the computation, please see Appendix C.3.

Now, the welfare effects are reported in the last four rows in Table 1.9. Consumer welfare increases by 0.53 million dollars, which is equal to the sum of 0.03 million dollars of reduction in total search costs and 0.50 million dollars of increase in total matched indirect utilities between consumers and their chosen products.[58] The changes on the demand side increases independent developers' variable profits (and equivalently, profits) by 0.41 million dollars.[59] These numbers

[56] Kim, Albuquerque and Bronnenberg (2010) and Ursu (2018) define consumer welfare in the same way.

[57] All expressions for expected consumer surplus are up to a constant. See Small and Rosen (1981).

[58] Allen, Clark and Houde (2019) finds that 50 percent of consumer surplus gain from reduced search frictions in mortgage markets is associated with reduced search costs, while 22 percent gain is associated with inefficient matching.

[59] Variable profit is measured as the total revenues from installation, in-app purchase and in-app subscription, plus

are small in magnitudes. Together, they imply a 3.8 million dollars of total welfare gain per month across the four categories in simulations. However, the total revenue on Apple app store is $160.5 million dollars per month across the same four categories. One reason is consumers' preference for Apple's apps over independent apps. Such a preference explains part of the observed difference between search rankings of Apple's apps and independent apps, limits the extent of identified self-preferencing, and thus its welfare effect.

Overall, the simulation exercise shows that self-preferencing leads to welfare loss on consumers and independent developers, without supply-side adjustment. The smaller structural estimates compared to the reduced-form estimates from the search algorithm change indicate that reduction in self-preferencing is a partial rather than the whole picture of the change. Combined with estimated demand, the limited magnitudes of the welfare effects imply that the benefit from eliminating self-preferencing might be surprisingly small due to confounded consumer preference.

1.7 Conclusion

This chapter identifies self-preferencing and quantifies its demand-side effects in the Apple App Store. To that end, it exploits product-level data on consumer search and purchase and leverages an un-anticipated search algorithm change. The existence and degree of self-preferencing are identified by comparing search rankings of Apple's apps and independent apps, conditional on app quality, app price, ratings, and text relevance. I then use the estimates to examine the demand-side welfare effects of self-preferencing.

The results show that self-preferencing existed on the Apple App Store in the U.S. market from April to August 2019 and led to a competitive advantage of Apple's apps. It implies that the observed high search rankings of Apple's apps during the self-preferencing period were not purely due to differences in qualities or app titles. Counterfactual simulations show that self-preferencing explains 71 percent of observed difference in search rankings between Apple's apps and independent apps.

I also find that without supply-side adjustment, eliminating the identified self-preferencing increases total installation by 1.6 percent, driven by growth of the downloads of independent apps. As a result, consumer surplus increases by 2.1 million dollars per month, which is a 0.2 percent improvement; while the variable profits of independent developers increase by 1.6 million dollars per month, which is a 0.6 percent rise.

While we learned that self-preferencing directly hurts consumers by increasing search costs and causing worse match between consumers and products, it remains a question that whether supply-

in-app-advertising profits and minus variable update costs. Profit is variable profit minus fixed update costs. Without supply-side adjustment, change in variable profit is equal to changes in profit.

side adjustment will exacerbate or offset the demand-side welfare effect. This question is tackled in Chapter 2.

CHAPTER 2

Update Competition and Equilibrium Welfare Effect of Self-preferencing in Mobile Application Markets

2.1 Introduction

Self-preferencing in search algorithms is of concern to both antitrust authorities and independent suppliers. From an antitrust perspective, search manipulation evokes concern about how to regulate digital economies to ensure healthy competition and, by extension, product/service quality. [1] From the perspective of third-party suppliers, the concern is that self-preferencing hampers innovation under budget constraint.[2] This suggests that it is important to understand how self-preferencing affects product quality and welfare in equilibrium, which is the research question of this chapter.

Theoretically, it is unclear how eliminating self-preferencing impacts independent product quality. On the one hand, a search algorithm without self-preferencing increases the visibility of third-party products and thus potentially increases the marginal revenues from quality. This encourages quality provision. On the other hand, higher rankings bring in consumers that could have been attracted by higher quality and thus potentially decrease the marginal revenues from quality. This discourages quality provision. Therefore, it is an empirical question whether eliminating self-preferencing increases the quality of third-party products.

Built on Chapter 1, this chapter focuses on update frequency of mobile applications as a strategic decision for quality provision, and develops an empirical model of update competition to quan-

[1] Examples include i) one clause (section-2-(a)-(1)) in a proposed bill in the U.S. House providing that certain discriminatory conduct by covered platforms shall be unlawful, see 117th Congress, 2021, "H.R.3816 - American Choice and Innovation Online Act", congress.gov, June 11. https://www.congress.gov/bill/117th-congress/house-bill/3816/text; ii) Google's antitrust cases, see Molla, Rani and Estes, Adam Clark. 2020. "Google's three antitrust cases, briefly explained". Vox, Dec 17. https://www.vox.com/recode/2020/12/16/22179085/google-antitrust-monopoly-state-lawsuit-ad-tech-search-facebook

[2] A quote from an app developer on Apple App Store, reported by Wall Street Journals in 2019: "We want to invest more on that aspect (innovation), however, instead of hiring another two AI Ph.D.s, we have to use that money to just get ranked higher." See Mickle, Tripp. 2019. "Apple Dominates App Store Search Results, Thwarting Competitors", The Wall Street Journal, July 23. https://www.wsj.com/articles/apple-dominates-app-store-search-results-thwarting-competitors-11563897221

tify the equilibrium welfare effect of self-preferencing. In particular, in the reduced-form analysis of the search algorithm change on Apple App Store in Chapter 1, I find that the search algorithm change increased update frequencies of independent apps by 2.1%. However, I find no significant effects on other app characteristics such as price, average ratings, file size, and entry. Moreover, the demand estimation in Chapter 1 shows that consumers prefer apps with larger update frequency on average but with sizable taste heterogeneity, indicating that update frequency is a positive quality shifter.

In the update competition model, developers face uncertainty about search rankings and form beliefs on possible search rankings based on the search ranking model. Moreover, because update affects app quality which affects search ranking, apart from the direct effect of updates on installations, developers consider the indirect effect of updates on download through affecting search rankings. The indirect incentive is typically missing from existing supply models, but it is essential for studying platform design.

It is challenging to capture both direct and indirect incentives of update because the number of possible search rankings is the factorial of the number of products in the market. For example, an average market has 65 apps in the data, implying a belief space of $65!(\approx 8.2 \times 10^{90})$ orderings of products. Thus, allowing developers to consider all possible search rankings is infeasible for computation. To achieve tractability, I use a heuristic algorithm to truncate the set of possible search rankings and use the estimated search ranking model to calculate developers' beliefs on the truncated set. The justification behavioral assumption for the truncation method is that developers only consider some most likely search rankings when making update decisions. Results from within-sample tests show that the algorithm performs well at least in markets with a small number of products.[3]

Supply-side estimation shows that marginal update costs increases in update frequencies. I also obtain bounds on fixed costs for updates to rationalize the no-update observations following the approach taken by Fan and Yang (2020*a*). Specifically, I assume that an app developer's observed set of updated apps is profit-maximizing in a Nash equilibrium. Therefore, upgrading or not upgrading an app should not increase the developer's profit. Based on these conditions, I obtain market-specific upper bounds for apps that are updated and lower bounds for apps that are not updated. Estimation results show that the bounds are positively correlated with app qualities.

Based on the estimated consumer preferences, search costs, ranking probabilities, and update costs, I conduct counterfactual simulations to quantify the equilibrium welfare effects of self-preferencing. Similar to Chapter 1, I eliminate the identified preferential treatment on Apple's apps in the search results, but allow supply-side adjustment on update frequency. I find that re-

[3] In tests with all markets with no more than 10 products, the truncated set of possible search rankings can capture 60% to 100% of the top10 most likely ordering of products, with an average coverage rate of 85%.

moving the identified self-preferencing improves consumer surplus by \$2.2 million and the profit of independent developers by \$1.8 million per month in equilibrium. The resulting per-month gains in total surplus are equivalent to three-month consumers' installation payments or 2 percent of per-month consumer total payments for installations, in-app purchase and subscription. Furthermore, comparing the welfare effects across markets, I find a positive correlation between the welfare gains and how much the product ranking is improved due to eliminating self-preferencing. Overall, these counterfactual simulation results suggest that self-preferencing hurts both consumers and producers.

Zooming into app quality adjustment, the simulation results imply positive but modest average effect of eliminating the identified self-preferencing on update frequency and app quality. Specifically, I find that an average market sees a 0.4% increase in average update frequencies and a 0.01% increase in average app qualities after eliminating the identified self-preferencing. One reason for the small average effects is heterogeneity in the direction of changes. In particular, I find that the expected update frequency of an independent app might decrease up to 20% and might as well increase up to 25% after the elimination, which results in a small average product-level effect of 0.3%. Similar heterogeneity shows up in the effects on market-level average update frequencies and average app qualities. This heterogeneity result is consistent with the previously mentioned theoretical ambiguity on changes of product quality without self-preferencing. Given the positive and modest supply-side effects, while the increased quality provision contributes to the welfare gains, most of the welfare gains come from more efficient match between consumers and apps after shutting down the identified self-preferencing for Apple's apps.

2.1.1 Roadmap

The remainder of the chapter uses "preferential search ranking" and "self-preferencing" exchangeably, and is organized as follows. Section 2.2 describes the empirical model of update competition. Section 2.3 describes the estimation procedure and presents the estimation results. Section 2.4 presents counterfactual simulations. Section 2.5 concludes.

2.2 Modeling Update Competition

This section presents an empirical model of update competition in the presence of potential platform self-preferencing.

The reduced-form evidence in Section 1.3 shows that update frequencies of independent apps significantly increased due to the search algorithm change. It motivates me to focus on upgrading decisions in the supply model. In particular, except for updates, the model takes price, file size, and other app characteristics as exogenous. The reasons are two-fold. First, while app developers may change installation price, file size, and other app characteristics without self-preferencing for platform-owned apps, I do not find significant changes in these aspects in Section 1.3. Second, price and file size are quite sticky relative to updates as shown in Section 1.2.[4]

I use a static two-stage game to describe the update decisions of independent apps. In the first stage, app developers choose which apps are to be updated and form update portfolios. In the second stage, app developers choose positive update levels for those apps chosen to be updated. Due to the continuous effects of update on quality, developers may consider the impacts of current updates on future installations. The model abstracts away from such dynamic incentives for computational reasons.[5] Furthermore, the update decisions of Apple's apps are taken as given because the updates of Apple's apps might consider incentives other than maximizing the profits of the apps. The solution concept of the two-stage game is Nash Equilibrium. Now I describe the two stages in reverse order.

2.2.1 Developers' Choice of Update Levels

In the second stage of the supply model, search rankings are not realized yet. Thus, developers form a belief on the possible search rankings and infer the effect of their updates on the search rankings according to the search ranking model in Section 1.4.2. Then the variable update cost shocks for apps chosen to be updated in the first stage are realized. Developers observe each other's set of apps to be updated and the variable update cost shocks. Then, they pick positive update levels for the chosen apps to maximize expected variable profits, where the expectation is over possible search rankings.

I start with describing variable profits. App developers receive profits from three sources: i) installations, ii) in-app-purchase and in-app-subscription; iii) in-app-advertising. I describe them one by one. For an app j in market m, its revenue from installations is $p_{jm}Q_{jm}$. And its in-app-purchase and in-app-subscription revenue, R_{jm}, is given by

$$\begin{aligned} R_{jm} =& \tau_0 + (\tau_1 + \tau_2 \times game_j) \times Q_{jm} + (\tau_3 + \tau_4 \times game_j) \times Q_{jm}^2 + (\tau_5 + \tau_6 game_j) \times a_{jm} \\ &+ \lambda_g^R + \lambda_t^{[0]} + game_j \times \lambda_t^{[1]} + e_{jm}^R \end{aligned} \tag{2.1}$$

where $game_j$ indicates whether app j is a game app, λ_g^R are category-fixed effects, $\lambda_t^{[0]}$ are month-fixed effects, $\lambda_t^{[1]}$ are month-fixed effects interacted with the game indicator, e_{jm}^R are conditional mean-zero idiosyncratic error terms. Because there might be new in-app-purchase items available

[4]Furthermore, price changes only show up in 9% of the observations for paid apps, and file size changes show up in 25% of the observation. In contrast, version release(i.e., update) shows up in 50% of the observations.

[5]Future work should consider a dynamic problem.

along with updates, it is allowed that update levels, a_{jm}, may have direct effects on the revenues. During estimation, because the sum of revenues from in-app purchase and subscription are observed in the revenue data, the parameters(τ) in the equation will be estimated separately, and treated as data when estimating the supply model. Finally, to prepare for the expected variable profit function, denote the in-app-purchase and in-app-subscription revenues that are variable with respect to update levels as $R(Q_{lm}, a_{lm}; \tau) := (\tau_1 + \tau_2 \times game_j) \times Q_{jm} + (\tau_3 + \tau_4 \times game_j) \times Q_{jm}^2 + (\tau_5 + \tau_6 game_j) \times a_{jm}$.

In-app-advertising profits are not observed in the data. I assume that the in-app-advertising profit that is variable with respect to update is a simple quadratic function of installations:

$$F(Q_{jm}; \psi) = \psi_1 Q_{jm} + \psi_2 Q_{jm}^2 \tag{2.2}$$

On top of the installation revenues and in-app-purchase and in-app-subscription revenues, Apple collects 30% tax.[6]. Additionally, I assume zero marginal distributional cost to serve more consumers, which is reasonable for mobile applications.[7] Therefore, the variable profit of developer f in market m with a vector of search rankings (y_m) is given by

$$\begin{aligned} \pi_{fm}^I(y_m) = \sum_{j \in \mathscr{J}_{fm}} & 0.7 p_{jm} Q_{jm}(y_m) + 0.7 R(Q_{jm}(y_m), a_{jm}; \tau) + F(Q_{jm}(y_m); \psi) \\ & - g(a_{jm}, \omega_{jm}; \phi) \end{aligned} \tag{2.3}$$

where $\mathscr{J}_{fm}$ is the set of apps owned by the developer f in market m. The function $g(a_{jm}, \omega_{jm}; \phi)$ describes the variable update costs at positive level a_{jm} with unobserved cost shock ω_{jm}. Specifically, it is given by

$$g(a_{jm}, \omega_{jm}; \phi) = \phi_1 \exp(a_{jm}) + (z_{jm}^g \phi + \omega_{jm}) a_{jm} - \phi_1 \tag{2.4}$$

where z_{jm}^g contains age and month-fixed effects.[8] Notice that when update level is zero, the variable update cost is zero: $g(0, \omega_{jm}; \phi) = 0$. It is assumed that ω_{lm} is realized in the second stage and

[6]The Apple tax cut on installation revenues and in-app-purchase and subscription revenues are not exactly 30% for every app at all times. For example, for the subscription revenues from a consumer that has subscribed to the app for more than one year, Apple collects 15% of the post-one-year subscription revenues instead of 30%. However, I do not have data on the distribution of new and old consumers at the app level. Therefore, I use the 30% tax cut as an approximation of the actual tax cut.

[7]Li, Bresnahan and Yin (2016) studies app developers buying downloads to get their apps on the top chart and find that the median value of one organic download is 70% the cost of buying one download. Moreover, Armstrong and Zhou (2011) theoretically investigates multiple methods that firms can pay to become prominent and thereby influence the order in which consumers consider options. Such behavior on the supply-side is assumed away in this paper.

[8]Following the literature, I assume that marginal variable update cost is convex in update level (I expect ϕ_1 to be positive) so that the profit function is concave in update level.

not known by developers in the first stage. Therefore, ω_{lm} is independent of the decision on which apps are updated. This is the key assumption for instrument construction later in Section 1.5.1.

Now I explain developers' beliefs on search rankings. Computing beliefs on search rankings is challenging in multi-product markets because the number of possible orderings of products increases factorial in the number of products. For example, in a market of 5 products, the number of possible orderings is $5! = 120$; when the number of products increases to 10, the number of possible orders becomes $10!(\approx 3.6 \times 10^6)$. In the data, the number of products is 64.7 on average and could be as large as 102. Thus, it is infeasible to compute developers' beliefs on search rankings when developers consider all possible orders of search rankings. Neither is it feasible to approximate the set of all possible orders.[9]

To deal with the computational challenge, I assume that developers only consider most likely orderings (denoted by $\mathscr{B}_a$) when making update decisions. Then I approximate the most likely orderings with a method detailed in Appendix B.3. I test the method with all markets that have no more than 10 products. Out of the ten most-likely-to-happen orders, the truncated set of possible orders captures 8.5 top10 orderings on average, and captures at least 6 top10 orderings, and always captures the most likely ordering. Notice that the most likely ordering ranks products according to the mean ranking scores, which depend on the quality shifter, i.e., updates. Therefore, the truncated set of possible orderings depends on update levels, which is denoted by the subscript a. Given the truncated set of possible orders, $\mathscr{B}_a$, we can write down the probability measure on it based on the search ranking model in Section 1.4.2. Specifically, it is the probability of an order conditional on that it is in the set $\mathscr{B}_a$, which is given by

$$\tilde{\mathbb{P}}[y|a] = \mathbb{P}[y|a] \big/ \left(\sum_{y' \in \mathscr{B}_a} \mathbb{P}[y'|a] \right), \qquad \forall y \in \mathscr{B}_a \tag{2.5}$$

where $\mathbb{P}[y|a]$ is the probability of an order y defined in Equation (1.15), given the vector of update levels a.

The truncated set of possible orders, $\mathscr{B}_a$, along with its probability measure, $\tilde{\mathbb{P}}$, constitutes well-defined beliefs on search rankings for developers. Now, we can write down the expected

[9] I test with a market consisting of 10 products in the data. Denote the probability of an order y with $\mathbb{P}[y]$. To to reach $\sum_{y \in \mathscr{B}} \mathbb{P}[y] \geq 0.2$, I need at least 1% of all possible orders in the set of orders $\mathscr{B}$, which corresponds to 36,288 orders.

variable profits of a developer f in market m with the vector of update levels a_m as below

$$\begin{aligned}
&\pi^{II}_{fm}(a_m, \omega_m) := \mathbb{E}_y[\pi^I_{fm}(y)|a_m, \omega_m] \\
&= \sum_{y \in \mathscr{B}_{a_m}} \pi^I_{fm}(y)\tilde{\mathbb{P}}[y|a_m] \\
&= \sum_{y \in \mathscr{B}_{a_m}} \left\{ \sum_{l \in \mathscr{J}_{fm}} 0.7 p_{lm} Q_{lm}(a_m, y) + 0.7 R(Q_{lm}(a_m, y), a_{lm}; \tau) + F(Q_{lm}(a_m, y); \psi) \right\} \cdot \tilde{\mathbb{P}}[y|a_m] \\
&\quad - \sum_{l \in \mathscr{J}_{fm}} g(a_{lm}, \omega_{lm}; \phi)
\end{aligned} \tag{2.6}$$

where $\mathscr{B}_{a_m}$ is the truncated set of possible orders given a_m; $\tilde{\mathbb{P}}[\cdot]$ is the probability measure defined in Equation (2.5).

With the objective function in Equation (2.6), now I derive the necessary conditions for observed positive update levels in Nash Equilibrium. Specifically, they are the first-order conditions that the marginal expected variable profit with respect to update level equals zero, given the update levels of other developers' apps. Let R'_{jm} denote the first-order derivative of R_{jm} with respect to Q_{jm}, and D_{jm} denote the indicator of app j is updated in market m. The first-order conditions are given by (market index m is omitted for exposition)

$$MB_j(a_j, \psi) = \phi_1 \exp(a_j) + z^g_j \phi + \omega_j, \qquad \forall j \text{ s.t. } D_j = 1 \tag{2.7}$$

where the left-hand-side term marginal benefit of updates, $MB_j(a_j, \psi)$, is given by

$$\begin{aligned}
MB_j(a_j, \psi) &= MB^{[0]}_j + MB^{[1]}_j \psi_1 + MB^{[2]}_j \psi_2, \\
MB^{[0]}_j &= \sum_{y \in \mathscr{B}_a} \left\{ \sum_{l \in \mathscr{J}_{f(j)}} (0.7 p_l + 0.7 R'_l) \frac{\partial Q_l}{\partial a_j} + 0.7(\tau_5 + \tau_6 game_j) \right\} \tilde{\mathbb{P}}[y|a] \\
&\quad + \sum_{y \in \mathscr{B}_a} \left\{ \sum_{l \in \mathscr{J}_{f(j)}} (0.7 p_l Q_l + 0.7 R_l) \right\} \frac{\partial \tilde{\mathbb{P}}[y|a]}{\partial a_j} \\
MB^{[1]}_j &= \sum_{y \in \mathscr{B}_a} \left\{ \sum_{l \in \mathscr{J}_{f(j)}} \frac{\partial Q_l}{\partial a_j} \right\} \tilde{\mathbb{P}}[y|a] + \sum_{y \in \mathscr{B}_a} \left\{ \sum_{l \in \mathscr{J}_{f(j)}} Q_l \right\} \frac{\partial \tilde{\mathbb{P}}[y|a]}{\partial a_j} \\
MB^{[2]}_j &= \sum_{y \in \mathscr{B}_a} \left\{ \sum_{l \in \mathscr{J}_{f(j)}} 2 Q_l \frac{\partial Q_l}{\partial a_j} \right\} \tilde{\mathbb{P}}[y|a] + \sum_{y \in \mathscr{B}_a} \left\{ \sum_{l \in \mathscr{J}_{f(j)}} Q_l^2 \right\} \frac{\partial \tilde{\mathbb{P}}[y|a]}{\partial a_j}
\end{aligned}$$

where $MB^{[0]}_j$ is the marginal benefit of update from the installation and in-app-purchase and subscription revenues; the sum of $MB^{[1]}_j$ and $MB^{[2]}_j$ is the marginal benefit of update from the in-app-advertising profit. In particular, the first term in $MB^{[0]}_j$ is an example for the direct incentive to update: to directly increase downloads and revenues; while the second term in $MB_j^{[0]}$ is an example for the indirect incentive to update: to indirectly increase expected profits through improving the ordering likelihood($\tilde{\mathbb{P}}[\cdot]$). The partial derivatives $\frac{\partial \tilde{\mathbb{P}}[y|a]}{\partial a_j}$ are simulated based on parameters in the search ranking model. As the last piece of notation, R_l' is the derivative of the in-app-purchase and subscription revenue function $R(Q_l, a_l; \tau)$ with respect to downloads Q_l, and $R_l = R(Q_l, a_l; \tau)$.

There is a technical assumption underlying the first-order-condition approach: the objective function is locally differentiable with respect to the update levels. However, update levels might discretely affect the expected variable profits by changing the truncated set of possible search rankings. Therefore, I assume that marginal changes in update level a_j does not change the truncated set of possible search rankings $\mathscr{B}_a$. This assumption is not strong for two reasons. Firstly, when $\mathscr{B}_a$ equals the set of all possible search rankings, this assumption is a fact. Secondly, because the rank-ordered logistic regression model only requires higher ranking score to be ranked higher, there is typically a space for update level a_j to change the ranking scores without changing the rankings (but it will change the ordering likelihood).[10]

The first-order conditions imply ambiguous effect of self-preferencing on update. Assuming that an independent app j is boosted up in search results due to eliminating self-preferencing, then the demand curve is shifted up, which increases the marginal download from increasing update, i.e., higher $\frac{\partial Q_j}{\partial a_j}$. At the same time, the shifted-up demand curve moves the developer rightward along the marginal revenue curve, which will decrease marginal revenue from downloads, i.e., lower $(0.7p_j + 0.7R_j' + \psi_1 + 2\psi_2 Q_j)$, if the revenue curve is concave in quantity. Appendix D discusses the ambiguity implication in more details.

The first-order conditions in Equation (2.7) can be estimated linearly because all the three variables in marginal benefits of update level, i.e., $MB_j^{[0]}$, $MB_j^{[1]}$, and $MB_j^{[2]}$, can be computed with observed data and estimates from the demand model, search ranking model, and in-app-purchase and in-app-subscription revenue equation. Therefore, the supply model will be estimated using GMM and the moments are constructed based on the first-order conditions. Lastly, Equation (2.7) also defines the second-stage equilibrium positive update levels: it is a function of variable update cost shocks (ω) and the update portfolios (D) from the first-stage game, denoted by $a^+(D, \omega)$.

2.2.2 Developers' Choice of Update Portfolios

In the first stage of the supply model, developers choose which apps to be updated or not. If an app j is updated in market m, its developer pays a fixed cost C_{jm} and will choose its positive update level in the second stage. Otherwise, the developer pays nothing and has to remain its update level

[10]In fact, in the data, there are only 10 out of 56,570 observations violating this assumption, whose update levels will be treated as exogenous.

at zero in the second stage. These fixed costs may be the expected loss from interrupting users' app usage or the marketing costs for the new feature. In contrast, the second-stage variable update costs may be the number of computer engineers required for the update level.

Now I describe the objective function of app developers in the first-stage game. Because the variable update cost shocks ω are not realized in the first stage, developers balance the expected second-stage variable profits with the fixed costs, where the expectation is over ω. Let J_f denotes the number of apps owned by developer f. Then, the app developer f chooses update portfolio $D_f := (D_1, \cdots, D_{J_f})$ to maximize the following profit function, given the update portfolios of the other developers (D_{-f}):

$$\pi^{III}_{fm}(D_{fm}; D_{-fm}) := \mathbb{E}_{\omega_m}[\pi^{II}_{f(j)m}(a^+_m(D_{fm}; D_{-fm}, \omega_m), \omega_m)|D_{fm}; D_{-fm}] - \sum_{j \in \mathscr{J}_f} C_{jm} D_{jm} \quad (2.8)$$

where ω_m are variable update costs for apps whose $D_{jm} = 1$.

Now I describe the necessary conditions for an app to be updated or not updated in a Nash Equilibrium. I follow the idea of no profitable deviation in Nash Equilibrium as in Fan and Yang (2020*a*).[11] Specifically, when an app j is not updated in market m, it must be that the additional expected second-stage variable profit of its developer $f(j)$ from updating it cannot cover its fixed cost:

$$\mathbb{E}_{\omega_m}[\pi^{II}_{f(j)m}(a^+_m(1, D_{-jm}, \omega_m), \omega_m)|1, D_{-jm}] - \mathbb{E}_{\omega_m}[\pi^{II}_{f(j)m}(a^+_m(0, D_{-jm}, \omega_m), \omega_m)|0, D_{-jm}] \leq C_{jm}, \forall D_{jm} = 0 \quad (2.9)$$

On the other hand, when an app j is updated in market m, it must be that the additional expected second-stage variable profit of its developer $f(j)$ from updating it can fully cover its fixed cost:

$$\mathbb{E}_{\omega_m}[\pi^{II}_{f(j)m}(a^+_m(1, D_{-jm}, \omega_m), \omega_m)|1, D_{-jm}] - \mathbb{E}_{\omega_m}[\pi^{II}_{f(j)m}(a^+_m(0, D_{-jm}, \omega_m), \omega_m)|0, D_{-jm}] \geq C_{jm}, \forall D_{jm} = 1 \quad (2.10)$$

Therefore, Equation (2.9) obtains the lower bounds on the fixed costs for apps that are not updated, and Equation (2.10) obtains the upper bounds on the fixed costs for apps that are updated. I explain how the expected second-stage variable profits are calculated in the Section 1.5.1 and how equilibrium update portfolios (D^*) are computationally found in Section **??**.

[11] Fan and Yang (2020*b*) develops another method to recover fixed costs. The method is based on bounds of conditional choice probabilities and does not require solving a game.

2.3 Estimation

This section presents the estimation procedures and results of the empirical model of update competition.

2.3.1 Estimation Procedures

Update Costs. The estimation of supply has two steps. First, a reduced-form in-app-purchase and subscription revenue function is estimated based on Equation (2.1) in OLS, where invalid observations are dropped from the sample.[12] Then, I construct moments using Equation (2.7) and estimate the parameters using GMM. Notice that only positive update levels are included in the sample for estimating Equation (2.7). Similar to the demand model, there are endogeneity concerns. Specifically, update levels may be endogenous because developers know unobserved (to researchers) marginal update cost shocks(ω_{jm}) when choosing the positive update levels. For example, an app might receive particularly positive and constructive comments in a certain month, and the developer is inspired to think of a straightforward way to improve user experiences. These unobserved comments increase the marginal benefits from the update and decrease the marginal cost of update, leading to higher update levels. Such omitted variables will bias the coefficient on update in Equation (2.7) upwards.

I construct instruments based on category-fixed effects and pre-determined characteristics, including price, file size, and title match with popular keywords. This estimation strategy is based on the timing assumptions that i) update portfolios are determined after unobserved demand shifters(ξ) and the above pre-determined characteristics, and ii) marginal update cost shocks are realized after update portfolios. The first-stage regression results for these supply-side instruments are reported in Appendix B.4. During the estimation, I apply constraints to guarantee profit maximization conditional on the positive update levels of competing apps, which are listed in Appendix B.5.

As for fixed costs, I use the inequalities 2.9 and 2.10 to obtain bounds. Following Fan and Yang (2020*a*), I calculate the bounds by calculating changes in expected second-stage variable profits from adding an un-updated app into update portfolios or dropping an updated app from update portfolios. The expectation is over marginal update cost shocks ω_m and possible search rankings. To compute the expected variable profits, I draw the cost shocks from their empirical distribution. Then, I compute the update equilibrium in the second-stage game for each cost-shock draw, which returns the second-stage (expected) variable profits. Then, I take the average of these second-stage

[12] Specifically, I drop observations that i) have negative in-app-purchase and in-app-subscription revenues; or ii) do not provide in-app-purchase or in-app-subscription services.

variable profits across all cost-shock draws.

There is a computational burden when evaluating the equilibrium second-stage variable profits, mainly due to the high dimension of possible search rankings.[13] To alleviate the computational burden, I restrict the sample for computing the fixed-cost bounds with two steps. First, I focus on relevant markets. Specifically, I only consider apps in categories with Apple's apps during the difference-in-differences sample period (Jun.-Nov.,2019).[14] It reduces the number of markets from 874 to 96. Second, following the literature, I focus on the top5 developers in each category and only allow these top5 developers to change their updates.[15] With these restrictions, there are 556 fixed-cost upper bounds and 176 fixed-cost lower bounds to be estimated.

2.3.2 Estimates of Revenues and Update Costs

Table 2.1 reports the estimates for parameters of the supply model. First, the estimation results for in-app-purchase and in-app-subscription revenues show intuitive results: i) revenues increase with downloads concavely, and ii) update levels directly contribute to revenues, especially for game apps. The estimates imply that an average app receives \$4.3 from in-app-purchase and in-app-subscription with each new download(consumer).

Second, the estimation results for the update-level first-order conditions show that i) in-app-advertising profits insignificantly increase with downloads[1]; and ii) marginal update costs are larger for apps with higher update levels and/or fewer experiences. The results imply that an average app earns \$0.07 from in-app advertising with each new download. The resulting average marginal update cost is \$0.32 million. To give some contexts for the magnitudes, I do the following back-of-envelope calculation. One weighted update is equivalent to $\log(2)$ update levels, and the average salary for a computer engineer is about \$97,000 per year in California. Then, the average marginal update cost is equivalent to hiring 28 ($\approx 12 \times \log(2) \times 0.32/0.097$) computer engineers in a month. This magnitude seems reasonable.

As for fixed costs of updates, I obtain upper bounds for each updated app and lower bounds for each app that is not updated in each market. Figure 2.1 plots the estimated upper bounds(Panel A) and lower bounds(Panel B) on the vertical axes against app quality on the horizontal axis. The

[13]For example, an average market with 65 products has 141 different orderings of products in the truncated set of possible search rankings. So then, there are 9165 ($= 65 \times 141$) market shares to be computed. And such computation is embedded in the evaluation of each new vector of update levels when firms choose their best-response update levels. It is as if computing price equilibrium for a market with 9165 products. For more details on finding the equilibrium positive update levels, please see Appendix B.6.

[14]The purpose of computing the bounds is to simulate counterfactual update equilibrium when shutting down the self-preferencing. Therefore, only the categories with Apple's apps will have different counterfactual equilibrium than status-quo equilibrium.

[15]The category-specific top5 developers are found based on total downloads of owner apps during the post-change difference-in-differences sample period.

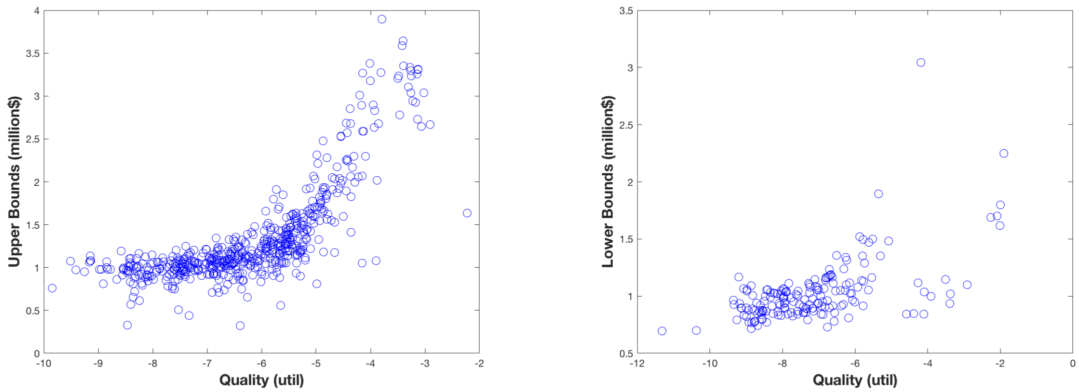
Table 2.1: Supply Model Estimates

Variables	Parameter	Standard error
In-App-Purchase and In-App-Subscription Revenue Parameters		
Downloads (million)	5.619	0.641
Downloads $\times$ Game	-2.765	0.701
Squared Downloads	-0.401	0.196
Squared Downloads $\times$ Game	0.163	0.231
Update Level	0.078	0.030
Update Level $\times$ Game	0.400	0.040
Constant	0.052	0.188
Category-FE	YES	
Month-FE	YES	
Game x Month-FE	YES	
Average Marginal Revenue ($)	4.298	
Observations	37,382	
In-App-Advertising Profit Parameters		
Downloads	0.068	0.429
Squared downloads	-0.003	0.085
Marginal Update Costs Parameters		
exp(Update Level)	1.631	0.052
Age	-0.002	0.000
Constant	-3.731	0.146
Month-FE	YES	
Average Marginal In-App-Advertising Profit ($)	0.067	
Average Marginal Cost (million$)	0.323	
Observations	25,325	

Notes: Update level is $\log(1+\text{update frequency})$. Marginal revenue and marginal profit are with respect to downloads. Marginal update cost is with respect to update levels.

Figure 2.1: Bounds of Fixed Costs of Updates (million \$) *v.s.* Quality

Panel A. Upper Bound Panel B. Lower Bound



average upper bound in Panel A is \$1.34 million, and the average lower bound in Panel B is \$1.23 million.

In addition, as cross-validation of the model fitness, Appendix C.1 compares structural and reduced-form estimates of the average treatment effect of the search algorithm change in July 2019. It shows that both methods generate the same direction and similar magnitudes of changes in updates and search rankings. The structural estimate of the average treatment effect has the same sign as the DiD estimate but with smaller magnitudes, which is explained in the appendix.

2.4 Equilibrium Effects of Self-preferencing

Using the estimates from the supply model, and the estimates from the demand and search ranking models in Chapter 1, I examine counterfactual simulations in this section to quantify the effects of self-preferencing on quality provision and welfare in equilibrium. To that end, I focus on changes in market *equilibrium* when shutting down the identified self-preferencing, *with* supply-side adjustment.

In alignment with section 1.6 in Chapter 1, the same eight markets are covered in counterfactual simulations: entertainment category, health & fitness category, music category, and utilities category in June and July in 2019. Moreover, I also fix the search costs of outside options throughout the counterfactual simulations here, as I did in Chapter 1. It turns out that the changes of equilibrium due to fixing pre-installed apps' search rankings are small, which is detailed in Appendix C.2.

2.4.1 Effects of Self-preferencing on Update Frequency and App Quality

I start with effects on quality provision. To that end, I focus on the changes in update frequency and quality index. Recall that app quality is positively affected by update frequency on average, as shown in the demand estimation. Therefore, changes in update frequencies then lead to changes in app qualities.

I take two main steps to find counterfactual update frequency and app quality. First, following Fan and Yang (2020a), I take simulation draws of fixed costs of update from a range that is consistent with the identified bounds. Specifically, for each updated app in the data, I have obtained an upper bound for the fixed cost of updating it, $\overline{C}_{jm}$. For such an app, I uniformly take five sunk-cost draws from the range $[0.5\overline{C}_{jm}, \overline{C}_{jm}]$. On the other hand, for each app that is not updated in the data, I have obtained a lower bound for the fixed cost of updating it, $\underline{C}_{jm}$. For such an app, I uniformly take five sunk-cost draws from the range $[\underline{C}_{jm}, 5\underline{C}_{jm}]$.

Second, for each simulation draw of fixed costs, an equilibrium set of update portfolios across developers is computed based on best-response iterations. Here, I have two layers of best-response iterations. The top layer is the best-response iterations of update portfolios, which could be challenging with multiple-product firms. For example, one affected developer has up to 12 apps, which creates a choice among $2^{12} (= 4096)$ update portfolios. To alleviate the computational burden, I apply the heuristic algorithm for finding the best-response product portfolio from Fan and Yang (2020a) to find the best-response update portfolio. The bottom layer is the best-response iterations of positive update levels for apps chosen to be updated. [16] Given each equilibrium of update portfolios, I compute the second-stage equilibrium positive update frequency with estimated variable update cost shocks ($\hat{\omega}$) whenever possible. [17]

One last detail on the simulation. As explained in section 2.3.1, only the fixed costs of apps owned by the category-specific top5 independent developers are computed. Therefore, in the counterfactual simulations, only these top5 developers are active players in the update competition game, and the updates of the other apps are holding fixed. Given that, there are ten apps whose updates are subject to changes in an average market, while there are 90.4 independent apps in total. On average, the active apps account for 45.1% of total downloads in a market. In Appendix C.5, I

[16] In the bottom-layer best-response iteration, the variable update cost shocks(ω) are the same as those in the computation of bounds on fixed costs.

[17] Therefore, as a rigorous interpretation, the simulated outcomes are the expected outcomes of the simulated second-stage update competition game, given the simulated first-stage equilibrium update portfolios and the identified variable cost shocks($\hat{\omega}$). The expectation is over draws of variable cost shocks for apps that are not updated in the data but simulated to be updated in the counterfactual. These variable cost shocks are drawn from the empirical distribution of $\hat{\omega}$. This step ensures that the difference between the simulated update levels and observed update levels is not due to different variable cost shocks. Finally, these update levels are transformed into update frequencies using the definition equation that update level equals $\log(1 + \text{update frequency})$ and fed into the definition Equation (1.13) to calculate app qualities.

allow all independent developers to choose update levels but holding update portfolios fixed. The results are robust.

Table 2.2: Effects of Preferential Search Ranking on Update Levels

	Status-quo	Shut-down	Percentage Change(%)			
	mean	mean	mean	min	max	std
Market-Level Results						
Number of Updated Apps	6.98	7.03	0.71	0	2.86	1.26
Average Update Frequencies	1.17	1.17	0.44	−0.02	1.25	0.51
Average App Quality	−6.08	−6.08	0.01	−0.001	0.02	0.01
Product-Level Results						
Probability of Update[a]	0.71	0.71	0.01	−0.20	0.20	0.05
Expected Update Frequency[b]	1.14	1.15	0.28	−19.97	25.14	4.35
Expected App Quality	−6.27	−6.27	0.01	−0.28	0.30	0.06

Notes: Update frequencies are monthly number of updates weighted by length of release notes. Variables are computed for apps owned by the category-specific top5 independent app developers. For the status-quo case, there is identified preferential search ranking. For the shut-down case, there is no preferential search ranking. (a). For probability of update, the reported values in the last four columns are summary statistics on changes instead of percentage changes. (b).For expected update frequency, the summary statistics on percentage change are conditional on upgrading in status-quo. The expectation is over draws of sunk costs.

Table 2.2 shows the counterfactual simulation results for update frequencies and app qualities of independent apps with and without the identified self-preferencing. Overall, I find small average effects of eliminating self-preferencing. The top panel shows that, in an average market, eliminating self-preferencing leads to a 0.7 percent increase in the number of updated independent apps and a 0.4 percent increase in average update frequencies. These changes in updates result in a 0.01 percent increase in average app qualities. The bottom panel shows that, for an average independent app, after eliminating self-preferencing, the probability of update increases by 0.01, and the expected app quality increases by 0.01%. Conditional on updating before eliminating self-preferencing, an average independent app increases expected update frequency by 0.3% after the elimination. [18]

However, large but heterogeneous effects might be hidden behind the small average effects when updates may increase or decrease after eliminating the self-preferencing. Recall that theory predicts ambiguous effects of self-preferencing on product quality, which implies the possibility

[18]Notice that these counterfactual simulations fix pre-installed apps' search rankings, which limits the magnitude of effects. Therefore, the reported effects here are smaller than those estimated from the difference-in-differences analysis(Table1.4), even though the self-preferencing is completely shut down (which is not achieved by the search algorithm).

of heterogeneous effect of self-preferencing on updates across products. Therefore, I examine the heterogeneous effects in the last three columns of Table 2.2. Specifically, compared to the small average effects, the most astonishing result shows up in changes of expected update frequencies, where I find that an independent app's expected update frequency might decrease by 20% and might as well increase by 25%.[19] These effects are large relative to the average effects. Moreover, the standard deviation of the effects on the expected update frequencies is 15.5 times its mean, again indicating sizable heterogeneity in the effects on updates. Similarly, I find both negative and positive effects on the probability of update and expected app quality. Not only does the heterogeneity exists at the product level, but I also find both negative and positive effects on average update frequencies and average app qualities at the market level, except that no market has a decrease in the number of updated apps.[20] Overall, there is sizable heterogeneity in the direction and magnitude of the effects of self-preferencing on update frequency and app quality.[21]

2.4.2 Equilibrium Welfare Effects of Self-preferencing

Given the counterfactual update frequencies and app qualities, I calculate counterfactual search rankings, installations, consumer surplus, and developer profits. I maintain the definition of consumer surplus in Equation (1.16) in Chapter 1.

Table 2.3 shows the effects of eliminating self-preferencing in equilibrium. Regarding the equilibrium effect on search ranking, the first three rows show results that are almost the same as those found in Chapter 1, except for a 0.01 percent larger increase in the value of Apple's apps' search ranking in row (3). Similarly, rows (4) to (6) in Table 2.3 show the equilibrium effect on installation, which is almost the same as those found in Chapter 1, except for a 0.01 percent larger increase in total installation in row (4). These little discrepancies are consistent with the modest effect on update frequency and app quality: the quality improvement is too small to generate noticeable further changes in search rankings and installations.

Regarding equilibrium effects on welfare, rows (7) to (11) show that consumer surplus increase by 0.55 million dollars and profits of independent developers increase by 0.40 million dollars

[19] Moreover, 38 percent of the estimated effects on expected update frequency are positive, and 19 percent of the estimated effects are negative, which constitutes 57 percent of observations that see changes of expected update frequency and quality in the counterfactual simulations. Furthermore, conditional on updating before the elimination, the average positive effect is a 1.3 percent increase of expected update frequency, the average negative effect is a 1.4 percent decrease of expected update frequency. Both of the average positive effect and the average negative effect are larger than the average effect.

[20] All markets see changes in average update frequencies and average app qualities. 62.5 percent of the markets see positive changes, while 37.5 percent of the markets see negative changes. Additionally, one-quarter of the markets see changes in the number of updated apps, and 5 percent of observations see changes in expected update-or-not choice.

[21] Appendix C.4 explains why some independent apps increase update frequency after eliminating the identified self-preferencing while others decrease update frequency.

Table 2.3: Eqruilibrium Effects of Preferential Search Ranking on Search Rankings, Installations and Welfare

	Variable	Status-quo	Shut-down	Mean Δ	Mean %Δ
(1)	Average Search Rankings	38.92	38.92	0	0
(2)	- Independent apps	39.86	39.23	−0.63	−1.56
(3)	- Apple's apps	14.35	31.85	17.51	142.08
(4)	Total Installations (million)	15.16	15.40	0.24	1.56
(5)	- Independent apps	15.06	15.32	0.26	1.75
(6)	- Apple's apps	0.10	0.08	−0.03	−26.58
(7)	Consumer Surplus (million $)	321.40	321.95	0.55	0.17
(8)	Variable Profits (million $)	76.60	77.04	0.45	0.68
(9)	Profits (million $)	69.27	69.66	0.40	0.68
(10)	Total Search Costs (million $)	15.39	15.35	−0.04	−1.53
(11)	Total Realized Utility (million $)	336.79	337.31	0.51	0.15

Notes: For the status-quo case, there is identified preferential search ranking. For the shut-down case, there is no preferential search ranking. Variable profits and profits are calculated based on independent developers.

in an average market.[22] It implies that, after accounting for supply-side adjustment, eliminating self-preferencing benefits both consumers and independent developers, while the magnitude of the benefit remains small (as those found in Chapter 1 without supply-side adjustment).

Comparing the welfare effects with (this Chapter) and without (Chapter 1) supply-side adjustment tells us the contribution of the increased update frequency to the equilibrium welfare gain. Columns (1) and (3) in Table 2.4 reports the comparison result with higher accuracy. It shows that update adjustment does contributes to the equilibrium welfare gain, although modestly. For example, update adjustment raises up the increase in consumer surplus from $0.531 million to $0.548 million, and enlarges the growth of variable profits of independent developers from $0.410 million to $0.448 million.

Column (2) of Table 2.4 reports the welfare effects of eliminating the identified self-preferencing in a partial-game simulation, where I allow all independent app developers to adjust their positive update frequency but holding update portfolios fixed. Comparing columns (2) and (3), it shows that allowing partial update adjustments also improves the gains in consumer surplus. Moreover, comparing columns (2) and (1), it shows that allowing more independent apps to adjust update frequency enlarges the gain in consumer surplus while shrinks the gain in variable profits of independent developers. I interpret such a difference as the effect of competition be-

[22]Profits of Apple's apps decrease by $0.001 million in an average market.

Table 2.4: Welfare Effects with and without Update Adjustment

Mean of	(1) With Update Portfolio Adjustment	(2) With Conditional Update Level Adjustment	(3) Without Update Adjustment
ΔCS (million $)	0.548	0.551	0.531
%ΔCS (%)	0.172	0.173	0.167
ΔVariable Profits (million $)	0.448	0.408	0.410
%ΔVariable Profits (%)	0.678	0.618	0.621

Notes: Update portfolio and level adjustment holds updates of non-top5 independent developers' apps fixed. Conditional update level adjustments allow all independent developers to adjust update levels of apps that are updated in status-quo. Variable profits are calculated based on independent developers.

tween developers because there are more active developers in the partial-game simulation than the portfolio-adjustment simulation in the first column.

All together, the results from this section show that self-preferencing for platform-owned products decreases both consumer surplus and independent developer profits, which is exacerbated when accounting for quality adjustments. This conclusion is further confirmed by the heterogeneity of the welfare gains across markets. In particular, Figure 2.2 compares the welfare gains across markets with different extents of ranking imperfection induced by the identified self-preferencing. [23] As a correlation relationship, Figure 2.2 shows that the larger the ranking imperfection due to self-preferencings, the more the gains in consumer surplus and independent developer profits, again suggesting that self-preferencing for platform-owned products leads to welfare losses.

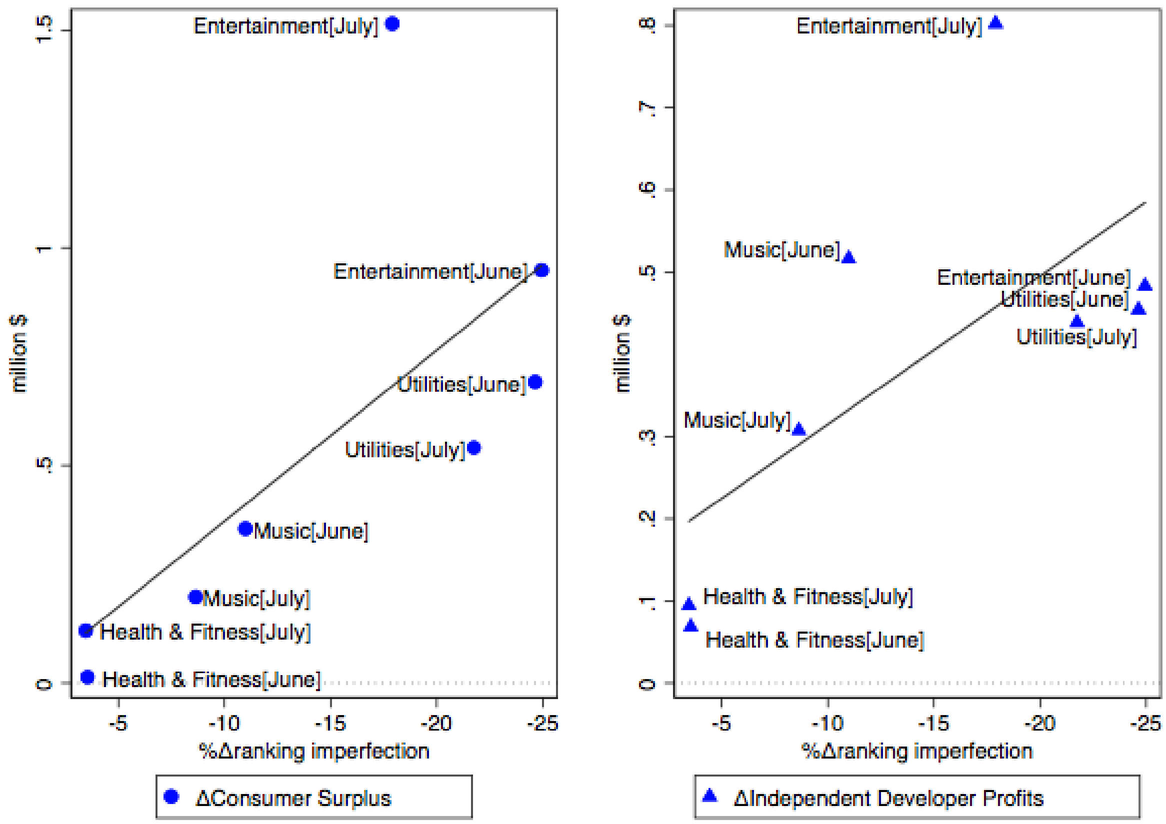
2.5 Conclusion

This chapter develops and estimates an empirical model of update competition with potential self-preferencing in the mobile application industry. Then, this model is used to quantify the equilibrium welfare effect of the identified self-preferencing in Chapter 1. Together, these two chapters contribute to the literature by developing an empirical model of consumer search and update competition in the mobile application industry. As an attractive feature, the empirical model is identifiable with product-level data on consumer search and purchase.

Based on counterfactual simulation, this chapter finds that eliminating the identified self-preferencing increases consumer surplus by $2.2 million and independent developer profits by $1.6

[23] Appendix C.6 explains how the ranking imperfections are calculated. One related factor is the number of Apple's apps. For example, the Health & Fitness category only has one non-preinstalled Apple's apps, while the other three categories have no less than 3 non-preinstalled Apple's apps. It explains why the Health & Fitness category sees the smallest welfare effect.

Figure 2.2: Welfare Effects and Ranking Imperfection due to self-preferencing for Apple's Apps



million per month in equilibrium. Additionally, consistent with theory, I find sizable heterogeneity in both the direction and the magnitude of the effects on update frequency and app quality. For example, after eliminating the identified self-preferencing, an independent app's update frequency might decrease up to 20 percent and might as well increase up to 25 percent. The rich heterogeneity leads to small average effects on update frequency and app quality and modest contribution of supply-side adjustment to the equilibrium welfare gain.

Overall, these results imply that self-preferencing hurt both consumers and producers while negatively but modestly affecting quality provision on average. The approach can be applied to other settings where there might be self-preferencing but confounded by consumer preferences. The effect of self-preferencing may be different in other contexts depending on consumer preference and the extent of self-preferencing.

Lastly, I argue that these estimated welfare gains from eliminating self-preferencing are likely to be lower bounds of real effects. Due to the lack of data on the number of users for pre-installed apps, the qualities of pre-installed apps are over-estimated. Therefore, I fix pre-installed apps' search rankings when quantifying the effects of self-preferencing. However, there might be further improved search efficiency and total welfare from eliminating self-preferencing when the search rankings of pre-installed apps are subject to changes. Moreover, this chapter abstracts away from dynamic incentives of update and finds modest effects on update frequency and app quality in the short-run. However, the effects on updates are potentially larger in the long-run given that app quality is cumulative and update is estimated to positively affect the accumulation process.

As for future research, Chapter 1 and Chapter 2 assume that consumers know all the observable product characteristics before search. While I find some reduced-form evidence supporting the assumption to some extent, it is generally possible that consumers have limited information on some product characteristics before search. Relaxing this assumption could be studied in the future.

CHAPTER 3

Patent Licensing and Bias in Estimation and Prediction of Horizontal Merger Effects (with Travis Bruce Triggs)

3.1 Introduction

In empirical analyses of markets using structural models of pricing competition, high prices can be explained by high marginal costs, given market concentration and consumer preferences. In this paper, we examine a typically unobserved and commonly abstracted-away market feature that additionally contributes to high prices: patent licensing relationships between manufacturers.

Firms can exercise intellectual property rights by implementing technologies in producing goods and services and licensing these rights to other firms, including competitors. By licensing, a firm can recoup private research and development costs, which can incentivize *ex ante* innovation efforts, while publicly disseminating beneficial technologies in the marketplace. In many markets, firms license technologies to competitors[1] in exchange for payments that scale with the quantity sold of the competitor's products that implement the licensed technology.[2]. Under this payment structure, the incentives of the licensor and its competitors become partially aligned: the licensor does not want to price cut their competitors too much, so as to ensure fruitful royalty revenues. Because prices are strategic complements, the licensors' competitors won't charge low prices neither. Therefore, such alignment effects due to the patent licensing relationships softens pricing competition, resulting in higher prices.

When estimating an empirical model of price competition, if a researcher observes who licenses

[1]For example, in the cellphone market, Apple licenses scroll feature to IBM and Nokia. In the electric vehicles market, Toyota licenses hybrid technology to Daimler (Mercedes Benz). In the televisions market, the ATSC patent pool, whose licensors include Panasonic, LG and Samsung, licenses the package of essential patents to not only themselves but also competitors like Sharp (see: https://www.mpegla.com/programs/atsc/licensors/.

[2]Royalty revenue may also be sales-based, i.e., $r_f Q_l p_l$ We follow the theoretical papers in this literature to focus on per-unit royalties(Lerner and Tirole (2004), Lerner, Strojwas and Tirole (2007), Reisinger and Tarantino (2019).).

patents to whom at which royalty rates, the above "alignment effect" can be directly accounted for and estimates of marginal costs can be accurately obtained. Unfortunately, the terms and even the presence of intellectual licensing agreements are often kept confidential between licensors and licensees and are not publicly observable. If a researcher proceeds to estimate a supply model as if these agreements, and thus the associated alignment of incentives, were not actually present, the resulting estimates will be biased in theory. Therefore, unobserved patent licensing relationships introduce mis-specification errors into estimation of marginal costs.

Further, the bias in the marginal cost estimates will propagate through any analysis of the pricing competition, leading to biased predictions of counterfactual market outcomes. This is because the bias in marginal cost estimates are generally not constant between status-quo and counterfactual scenarios, and therefore cannot be "cancelled out" when evaluating the effect of market structure changes. The changes in bias of marginal cost estimates due to not accounting for licensing between status-quo and counterfactual scenarios root in the magnitudes of the alignment effect. For example, how much a licensor cares about its competitors' product sales depends on the elasticity of its competitors' product sales with respect to its own price. If its competitors' product sales are unaffected by its own price, the licensor has no need to charge lower prices to secure royalty revenues. Since price elasticities are market outcomes that depend on many aspects of the market structure, so does the alignment effect. Moreover, the structural difference between pricing competition models with and without patent licensing relationships naturally leads to different predictions of market outcomes.

The common existence in market analyses of counterfactual prediction bias due to unobserved existing patent licensing relationships, along with the typical difficulty of collecting patent licensing data, motivates us to ask and answer the following questions: Does the unobserved licensing behavior always lead to positive or negative bias in marginal cost estimates and subsequently the evaluation of counterfactual scenarios? Under what market conditions and which sorts of counterfactual structural changes would we larger prediction biases? When would we expect opposite predicted changes of market outcomes from models with and without patent licensing?

To answer these questions, we first explain the alignment effects and the resulting bias in estimation and predictions in a theoretical framework. Then, we specify a model of pricing competition with patent licensing relationships between competing manufacturers. We calibrate this model and conduct a series of simulation exercises to assess what features of a market, including royalty rates, product substitution patterns, and market concentration, lead to larger biases in estimation and counterfactual predictions when existing patent licensing relationships are unobserved and assumed-away. We consider two types of counterfactuals: a merger between licensor and licensee and a merger between licensees. This enables us to examine which type of counterfactuals are likely to have larger prediction biases. We use two sets of simulation exercises for two complementary purposes: theoretical simulations for examining how biases change with respect to underlying primitives and guidance simulations for showing, for what types of markets and their respective market characteristics would an analyst be most concerned with biases due to unobserved patent licensing.

We find that estimation bias in marginal costs are increasing with royalty rates and the sum of diversion ratios between licensor and licensees. The guidance simulations show that such bias is typically not ignorable: in a sample of reasonably generated markets, the median estimation bias is 28% of the true marginal cost. We show an alarming finding to researchers: not accounting for existing patent licensing relationships can lead to opposite prediction of merger effects. Such opposite predictions show up in both theoretical simulations and guidance simulations for both licensor-licensee mergers and licensee-licensee mergers. Because the guidance simulations cover a richer set of markets (including asymmetric market) compared to the theoretical simulations that are based on a symmetric benchmark market, opposite predictions show up more frequently in guidance simulations. In particular, in the theoretical simulations, we find 4.3% to 4.4% of cases where the merger effects on prices predicted from a mis-specified model not accounting for existing patent licensing relationships are opposite to the true merger effects predicted from a true model accounting for existing patent licensing relationships. In the guidance simulations, such fraction of opposite prediction cases can be as high as 51.2%. Moreover, both sets of simulations find that, opposite predictions happen when royalty rates are large. In particular, theoretical simulation finds that opposite predictions happen when royalty rates account for about 80% of total marginal costs; while guidance simulation finds that opposite predictions happen at high values of royalty rates in the random sample of markets. We also detail the economic driving forces for opposite predictions in each type of mergers. In licensor-licensee mergers, we find that the ignored saving of royalty payment due to licensor-licensee merger leads to a decrease in prices and an increase of consumer welfare after merger, which causes opposite predictions if one assumes away the patent licensing relationships. In licensee-licensee mergers, we find that ignored decrease in alignment effect due to licensee-licensee merger leads to decrease of licensor's price, which causes opposite predictions if one assumes away the patent licensing relationships. We argue that these two ignored channels when assuming away existing patent licensing relationships are also driving forces for overall prediction biases that are unconditional on opposite predictions.

In both sets of simulations and both types of mergers, we find that market-level merger effects are over-predicted when one assumes away existing patent licensing relationships. We further find that, the prediction biases are smaller in licensee-licensee merger than those in licensor-licensee merger. In particular, the guidance simulation finds that, in licensor-licensee mergers, assuming away existing patent licensing relationships lead to over-predicted increase of share-weighted average price by 2.6% at median, and over-predicted decrease of consumer welfare by 9.3% at median

; in licensee-licensee mergers, the two median predictions biases are 0.5% and 1.4% respectively.

We also examine the relationships between prediction biases, royalty rates and sum of diversion ratios between licensor and licensees. Theoretical simulation shows that prediction biases increase with royalty rates; while guidance simulation also shows positive correlations between the two. On the other hand, theoretical simulation shows non-monotonic relationship between prediction biases and the sum of diversion ratios; while guidance simulation finds that the potentials for prediction biases (i.e. maximum prediction biases) are positively correlated with the sum of diversion ratios.

This paper is related to three strands of the literature. First, we build on the literature studying the role of intellectual property licensing on competition (Reisinger and Tarantino, 2019; Layne-Farrar and Lerner, 2011; Lerner and Tirole, 2004). Second, we apply methods from the literature that uses theoretical and Monte Carlo simulations of structural models to evaluate the implications of modeling techniques, merger diagnostic tools, and different market features (Mazzeo, Seim and Varela, 2018; Sheu and Taragin, 2020; Balan and Brand, 2018; Miller et al., 2016, Miller et al., 2017; Crooke et al., 1999; Domnenko and Sibley, 2020, Das Varma and De Stefano, 2020; Dutra and Sabarwal, 2020). Finally, we contributes to the long literature in industrial organization of estimating parameters in product competition models and predicting counterfactual market outcomes (Berry, Levinsohn and Pakes, 1995; Nevo, 2001; Fan and Yang, 2020*a*). Our contribution lies in examining the consequences of not accounting for existing patent licensing relationship into the pricing competition models and even general supply-side models.

The remainder of the paper proceeds as follows: section 2 illustrates the theoretical framework to analyze estimation bias and prediction bias due to unobserved existing patent licensing relationships; section 3 describes the simulation methods; section 4 shows simulation results; section 5 concludes.

3.2 Theoretical Framework

3.2.1 Alignment Effects on Pricing and Estimation

We begin with explaining the alignment effect on pricing of patent licensing relationships between manufacturers in a Bertrand-pricing competition framework. A set of multiple-product firms compete in one market. Some of the firms are licensors, who license patents to other firms and collect royalty revenues. Some of the firms are licensees, who purchase licenses to legally use the patented technology in production, and pay royalties. Assume that a licensor-manufacturer, f, produces and sells a set of products, $\mathscr{J}_f$, and owns a single patented technology[3]. This firm f chooses product

[3] The model can be easily extended to allow multiple-patents licensor-manufacturers. We focus on single-patent firm for simple exposition.

prices to maximize the following profit function:

$$\pi_f = \sum_{j \in \mathscr{J}_f} (p_j - c_j) Q_j + r_f \sum_{l \in \mathscr{E}_f} Q_l \tag{3.1}$$

where c_j is product j's total marginal costs, Q_j is product j's unit sales, $\mathscr{E}_f$ is the set of other firms' products that use firm f's patented technology, and r_f is the exogenous per-unit royalty rate charged by firm f. The patent licensing relationship described by r_f and $\mathscr{E}_f$ partially aligns firms' pricing incentives: firm f's profit contains royalty payments that depend on other firms' sales, Q_l.

Now, we extend the profit function in Equation (3.1) to nest the profit functions for licensee-manufacturers. When $r_f = 0$ and $\mathscr{E}_f$ is empty, it is as if the firm f is a licensee. Moreover, we allow the total marginal cost, c_j, to contain royalty rates whenever necessary. Therefore, $c_j \geq r_f$ when j is a licensee-product. Under the assumption of Bertrand competition, profit-maximizing prices are characterized by the following first-order conditions[4]:

$$p_j = c_j - \left(\frac{\partial Q_j}{\partial p_j}\right)^{-1} \left(Q_j + \sum_{i \in \mathscr{J}_f \setminus \{j\}} (p_i - c_i) \frac{\partial Q_i}{\partial p_j}\right) - r_f \sum_{l \in \mathscr{E}_f} \frac{\partial Q_l / \partial p_j}{\partial Q_j / \partial p_j}, \qquad \forall j \tag{3.2}$$

Denote $\mu_j := -r_f \sum_{l \in \mathscr{E}_f} \frac{\partial Q_l / \partial p_j}{\partial Q_j / \partial p_j}$. We call μ_j as the *alignment effect* of the existing patent licensing relationship on the price of product j. We note that μ_j is always non-negative, and in particular always positive for licensors whose $r_f > 0$[5]. Therefore, with patent licensing relationships, the licensor has incentive to charge higher prices compared to the case without patent licensing relationships. Because prices are strategic complements, the other firms are incentivized to charge higher prices in response. We also note that Equation (3.2) nests the case of no patent licensing relationships by setting $r_f = 0$ for all f.

Now we consider how the lack of patent licensing data leads to estimation bias. Without patent licensing data, the royalty rates, r_f, the licensed products, $\mathscr{E}_f$, and even the identity of the licensor, f, are unobserved. Therefore, μ_j is unobserved. In such cases, μ_j is typically assumed away, which results in the following commonly used back-out equation for marginal cost:

$$\tilde{c}_j = p_j + \left(\frac{\partial Q_j}{\partial p_j}\right)^{-1} \left(Q_j + \sum_{i \in \mathscr{J}_f \setminus \{j\}} (p_i - \tilde{c}_i) \frac{\partial Q_i}{\partial p_j}\right) \tag{3.3}$$

[4] We think that the existence and uniqueness of the above game does not trivially follow the same argument for Nash-Bertrand pricing game without patent licensing relationships (Caplin and Nalebuff (1991), Vives et al. (2001), Aksoy-Pierson, Allon and Federgruen (2013)). This is mainly because the licensor-manufacturer does not directly choose the price of licensee-manufacturers' products.

[5] We assume that products are substitutes. Complement goods are beyond the scope of this paper.

where the prices are from the data and derivatives are from a correctly specified and unbiased demand model estimation. Comparing Equation (3.3) to Equation (3.2), the resulting bias in marginal cost estimation is driven by the alignment effect μ_j [6]:

$$\Delta c_j := \tilde{c}_j - c_j \approx \mu_j.$$

Because μ_j is always positive, the unobserved alignment effects are likely to result in overestimated marginal costs. In fact, we note that, for a single-product licensor-manufacturer, $\Delta c_j = \mu_j > 0$. Positive estimation bias in marginal costs is intuitive: the licensing relationships create incentives for firms to charge higher prices, which will be explained by higher marginal costs without licensing data. We note that $\mu_j = 0$ for licensees' products. In fact, comparing Equation (3.3) to Equation (3.2) for licensee-manufacturers' products whose $r_f = 0$, the two equations are exactly the same. This implies that licensee's marginal costs are estimated unbiasedly, even when the model ignores the existing patent licensing relationships. Therefore, we focus on estimation bias in the marginal costs of the licensor.

3.2.2 Bias in Counterfactual Predictions

While the bias in estimated marginal costs of single-product firms has a clear sign, the bias in counterfactual predictions does not. We start with counterfactual prices of licensor-manufacturers. There two offsetting effects. First, overestimated marginal costs lead to higher predicted prices. Second, not accounting for the alignment effects lead to lower predicted prices. The net effect on price prediction is determined by the relative importance of the two channels. Since the bias in predicted counterfactual prices is ambiguous, the prediction errors of counterfactual quantities, consumer welfare and profits are also ambiguous.

To formulate the idea, we define the price of a product owned by a licensor-manufacturer that is predicted without licensing data, $\tilde{p}'_j$, as solution to a mis-specified model: Equation (3.2) with $r_f = 0$ for all firms, status-quo biased marginal costs $\tilde{c}_j$ and counterfactual market structures. And we define the price of the same product that is predicted with licensing data, p'_j, as solution to a true model: Equation (3.2) with observed r_f and $\mathscr{E}_f$, unbiased marginal costs c_j and the same counterfactual market structures. Now, we consider the difference between $\tilde{p}'_j$ and p'_j, which is

[6] We note that Equation (3.2) is one equation among a system of equations to back out the vector of marginal costs of firm f. Therefore, the estimation bias is not exactly μ_j.

characterized by the following equation:

$$\Delta p'_j := \tilde{p}'_j - p'_j = \underbrace{\Delta c_j}_{\approx \mu_j} + (\tilde{\nu}'_j - \nu'_j) + \underbrace{r'_f \sum_{l \in \mathscr{E}'_f} \frac{\partial Q'_l / \partial p'_j}{\partial Q'_j / \partial p'_j}}_{-\mu'_j} \tag{3.4}$$

where $\tilde{\nu}'_j := -\left(\frac{\partial \tilde{Q}'_j}{\partial \tilde{p}'_j}\right)^{-1} \left(\tilde{Q}'_j + \sum_{i \in \mathscr{I}'_f \setminus \{j\}} (\tilde{p}'_i - \tilde{c}_i) \frac{\partial \tilde{Q}'_i}{\partial \tilde{p}'_j}\right)$, and $\nu'_j = -\left(\frac{\partial Q'_j}{\partial p'_j}\right)^{-1} \left(Q'_j + \sum_{i \in \mathscr{I}'_f \setminus \{j\}} (p'_i - c_i) \frac{\partial Q'_i}{\partial p'_j}\right)$. In this paper, we don't consider counterfactual scenarios where the licensor-manufacturer's marginal costs are changed. We note that $\Delta p'_j$ is determined by three terms: i) estimation bias Δc_j that is likely to be positive; ii) difference between the predicted "mark-ups" if there were no patent licensing relationships ($\tilde{\nu}'_j - \nu'_j$); iii) negative of counterfactual alignment effect, which is negative. While the second term does not have a clear sign, the first and the third terms are offsetting each other. Therefore, $\Delta p'_j$ has an unclear sign.

In general, The alignment effect μ_j itself will change in counterfactual scenarios. This is mainly because diversion ratios($-\frac{\partial Q_l / \partial p_j}{\partial Q_j / \partial p_j}$) are sensitive to changes of prices and quantities, which are typically changed in the counterfactuals. We focus on mergers among various counterfactual scenarios, because merger is an important studied object in the literature of industrial organization. We consider two types of mergers: licensor-licensee merger and licensee-licensee merger.

The mis-specified model described in Equation (3.3) and the true model described in Equation (3.2) are different in three ways when predicting the effects of horizontal mergers. Firstly, they have different estimated marginal costs as illustrated in Section 2.1. Secondly, the mis-specified model ignores a channel of cost saving after licensor-licensee mergers: the merged licensee can save the royalty payment to the licensor after a merger with the licensor. Thirdly, the mis-specified model ignores the royalty revenues as a part of the licensor's pricing incentive, both before and after the merger. This structural difference directly causes difference between the predicted prices from the mis-specified model and the true model. Moreover, the mis-specified model has different inferred pre-merger profits than the true model. In particular, the mis-specified model infer firms' profits as if $r_f = 0$ for all firms in Equation (3.1). We incorporate all these differences during simulations.

Given the same counterfactual, different market structures can lead to different changes of the alignment effect and thus different prediction biases. We note that the magnitude of the alignment effect, μ_j, is determined by two variables: i) the royalty rates (r_f), ii) the sum of diversion ratios between the licensor and licensees ($-\sum_{l \in \mathscr{E}_f} \frac{\partial Q_l / \partial p_j}{\partial Q_j / \partial p_j}$). We will examine how the biases in marginal cost estimations and merger effect predictions change with respect to the two market characteristics.

3.3 Simulations Methods

3.3.1 The Model and Specification

We simulate a model of three single-product firms, A, B and C. A is a licensor-manufacturer, B and C are licensee-manufacturers.

Demand. We follow the demand model in Mazzeo, Seim and Varela (2018) to allow flexible substitution patterns between "horizontally" equidistant products. A consumer n receives the following utility from choosing product j

$$u_{jn} = \theta_{jn} - \alpha p_j + \varepsilon_{jn} \tag{3.5}$$

and utility $u_{0n} = \varepsilon_{0n}$ from not purchasing. Here, α is the consumer's price coefficient and $(\theta_{jn}, \varepsilon_{jn})$ are two idiosyncratic taste shocks. We assume ε_{jn} follows i.i.d. Type 1 Extreme Value (T1EV) distribution with scale parameter σ_ε. θ_{jn} is draw from a multivariate normal distribution that allows for correlated shocks across products and non-zero means: $\theta_n \equiv (\theta_{1n}, \theta_{2n}, \cdots, \theta_{Jn})^T \sim \mathscr{N}(\theta|\delta, \Sigma)$[7]. The variance-covariance matrix Σ captures horizontally equidistant products with equal correlation (ρ) between θ_j and θ_l for each pair of products j and l:

$$\Sigma = \begin{bmatrix} \sigma^2 & \rho & \rho \\ & \sigma^2 & \rho \\ & & \sigma^2 \end{bmatrix}. \tag{3.6}$$

We follow Mazzeo, Seim and Varela (2018) to call ρ as the *travel parameter*, which captures the distance between products in the preference space, with preferences for close products being highly correlated. We note that ρ is one of the model primitives that determine the diversion ratios.

Let M be the total number of consumers. The model-predicted unit sales of product j is given by

$$Q_j(p) = M \int \frac{e^{\frac{1}{\sigma_\varepsilon}(\theta_{jn} - \alpha p_j)}}{1 + \sum_{l \in \mathscr{J}} e^{\frac{1}{\sigma_\varepsilon}(\theta_{ln} - \alpha p_l)}} d\Phi(\theta_n|\delta, \Sigma) \tag{3.7}$$

where $\mathscr{J} = \{A, B, C\}$ is the set of products. We normalize market size to 1. This normalization is without loss of generality because market size is cancelled out everywhere in the first-order condition Equation (3.2).

Pricing Game. We maintain the Nash-Bertrand Pricing competition assumption. Denote r

[7] As noted by Mazzeo, Seim and Varela (2018), θ_{jn} is comparable to a linear function of observed product characteristics and random taste shocks: $x_j\beta_n + \xi_{jn}$ where (β_n, ξ_{jn}) are random variables distributed according to some parameterized distribution. We also note that, when products are symmetric, such specification can nests consumer-invariant quality shocks, ξ_j, with $\rho = 1$. When we simulate asymmetric markets, we set $\sigma = 0$ and $\rho = 0$.

as the royalty rate charged by firm A. We allow the licensees to have the same marginal costs as the licensor, as long as the licensees' marginal costs are no smaller than the royalty rate: $c_j \geq r, \forall j \in \{B,C\}$. Applying the above set-up of three single-product firms and patent licensing relationships to the first-order condition, Equation (3.2), the equilibrium prices are characterized by the following equations:

$$\begin{aligned} p_A &= c_A - \left(\frac{\partial Q_A}{\partial p_A}\right)^{-1} Q_A - r \sum_{l \in \{B,C\}} \frac{\partial Q_l / \partial p_A}{\partial Q_A / \partial p_A}, \\ p_j &= c_j - \left(\frac{\partial Q_j}{\partial p_j}\right)^{-1} Q_j, \qquad c_j \geq r, \forall j \in \{B,C\}. \end{aligned} \tag{3.8}$$

We apply the measurement of the equivalent variation in McFadden et al. (1973) to quantify consumer surplus as blow

$$CS(p) = \int M\sigma_\varepsilon \ln\left[1 + \sum_{j \in \mathscr{J}} e^{\frac{1}{\sigma}(\theta_{jn} - \alpha p_j)}\right] d\Phi(\theta_n | \delta, \Sigma). \tag{3.9}$$

Merger Effects. We consider two types of merger: i) merger between the licensor, A, and one of the licensees; ii) merger between the two licensees, B and C. We measure merger effects on a market outcome y in percentage change before and after the merger: $\%\Delta y := (y' - y)/y$. We consider market outcomes (y) including prices, quantities, consumer welfare and producer surplus. Lastly, we measure prediction bias of merger effects with difference between the merger effects ($\%\Delta\tilde{y}$) predicted from the mis-specified model that does not account for patent licensing relationships and biased marginal cost estimates and the merger effects ($\%\Delta y$) predicted from the true model that accounts for patent licensing relationships and true marginal costs: denoted as $\Delta\Delta y := \%\Delta\tilde{y} - \%\Delta y$.

3.3.2 The Simulation Algorithm

We have two goals of simulation: i) examine the theoretical ambiguity between model primitives and the biases in estimation and prediction; ii) provide guidance on correlations between typically observed market structures and the biases in estimation and prediction. For these two different goals, we design two different simulation algorithms. For the first goal of theoretical relationships, the designed simulation algorithm changes one model primitive over a continuum of values, holding all the other primitives fixed at a benchmark level. We call this set of simulations as theoretical simulation. For the second goal of guidance, the designed simulation algorithm randomly draws a rich set of various markets, backs out model primitives from the drawn market outcomes, then simulates the biases in estimation and predictions in each market. We call this set of simulations as guidance simulation. We explain the details of each simulation algorithm as below.

Theoretical Simulation. In the theoretical simulation, we specify a benchmark market, and then change one single model primitive in each simulation, holding all other model primitives fixed at the benchmark value. The benchmark market is summarized in Table 3.1. In the benchmark market, we shut down all random coefficients by setting σ and ρ to be zero. Then, the demand model in Equation (3.7) is reduced to a logit-demand model. Next, we set the scale parameter of the logit error, σ_ε, to be 0.1, so that changes in demand parameters have more bites on quantities. Also, we shut down patent licensing relationship in the benchmark market, by setting royalty rate, r, to be zero. We normalize market size, M, to be 1. Lastly, we set the benchmark market as a symmetric one: all products have the same marginal cost value, c, equal to 1; market share of each product is 25%; own price elasticity of demand of each product is -2.5. Based on such a set of model primitives and market outcomes, we back out the value of the common mean utilities (δ) of the three symmetric products; the value of the price coefficient (α); and calculate the price that are consistent with the market shares and price elasticities.

In theoretical simulations, we set $\sigma = 1, \sigma_\varepsilon = 0.1$.[8] We simulate biases in estimation and predictions with respect to different values of royalty rate (r) within $[0,1]$ and the travel parameter (ρ) within $[0,1]$. The simulation step is 0.01 for each parameter. To be aligned with the literature, we report the bias with respect to the resulting sum of diversion ratios between licensor and licensees from ρ,i.e. $-\left(\frac{\partial Q_B/\partial p_A}{\partial Q_A/\partial p_A} + \frac{\partial Q_C/\partial p_A}{\partial Q_A/\partial p_A}\right)$.[9] Holding all the other model primitives fixed at the benchmark value in Table 3.1, for each pair of (r,ρ), we take the following steps to simulate the biases:

1. Simulate the pricing game described in Equation (3.8), which accounts for existing patent licensing relationships.

2. Back out the mis-specified marginal costs, $\tilde{c}_j$, as described in Equation (3.3). Calculate the estimation bias, $\Delta c_j := \tilde{c}_j - c_j$.

3. Simulate post-merger equilibrium with biased marginal costs, $\tilde{c}_j$, and the mis-specified model where $r = 0$ in Equation (3.8) with $r = 0$. Predict biased merger effects. In licensor-licensee mergers, we merge firm A and firm B.[10] We note that, in the mis-specified model, product B's total marginal cost is not changed after the merger, since r is unobserved and assumed away. In licensee-licensee mergers, we merger firm B and firm C.

[8] We follow Mazzeo, Seim and Varela (2018) to set $\sigma_\varepsilon = 0.1$ so that variation in horizontal differentiation, i.e., in the travel parameter ρ, may have a large impact on market outcomes.

[9] In the appendix figure G.12 we show that the travel parameter ρ determines the sum of diversion ratios between licensor and licensees.

[10] Because the two licensees are symmetric during the simulation, mergers between firm A and firm B are the equivalent to mergers between the licensor, A, and licensee C.

Table 3.1: The Benchmark Market in Theoretical Simulations

(a) Product Specific Values	A	B	C	(b) Common Values	
Mean Utility (δ)	1/3	1/3	1/3	Price Coefficient (α)	0.2
Marginal Cost (c)	1	1	1	T1EV Scale (σ_ε)	0.1
Royalty Rate (r)	0	0	0	Market Size(M)	1.0
Market Share (%)*	25%	25%	25%	Taste Heterogeneity (σ)	0
Price	5/3	5/3	5/3	Travel Parameter (ρ)	0
Elasticity*	-2.5	-2.5	-2.5		

* Equilibrium market outcomes that are used to back out the benchmark model primitives (δ,α), given $(c,r,\sigma_\varepsilon,\sigma,\rho)$.

4. Merger simulation with true marginal costs, c_j, and the true model as described in Equation (3.8). Predict true merger effects. We note that, in the true model, after the licensor-licensee merger, product B's total marginal cost is decreased by r.

5. Compute the prediction bias in merger effects: $\Delta\Delta y = \%\Delta\tilde{y} - \%\Delta y$, i.e. the difference between predicted percentage changes in the market outcome y after the merger from the mis-specified model and the true model.

Guidance Simulation. In guidance simulation, we set $\sigma = 0, \sigma_\varepsilon = 1$. We generate and analyze a given theoretical market by the procedure outlined below. To construct our sample of markets, we repeat this process sufficiently as to generate 1,000 markets. For each market, we normalize initial prices, $p_j^0 = 1$ for $j = 1,...,|J|$, where $|J|$ is the number of single-product firms in the market.

1. Obtain market shares by first drawing random variables $\check{s}_j \sim U[0,1]$ for $j = A,B,C$. Calculate firm j's market share by, $s_j = (1-s_0)\frac{\check{s}_j}{\sum_{j'=A,B,C}\check{s}_{j'}}$, where $s_0 = 0.25$ represents the initial share of the outside good. Back out mean utilities, $\delta_j = \log(s_j) - \log(s_0)$.

2. Draw the licensor's product margins as, $m_A \sim U[0.25, 0.75]$.

3. Draw the royalty rate uniformly as, $r \sim U[0, m_A]$.[11]

4. From the licensor's first order condition taking into account the licensing relationship, back out the price sensitivity parameter as,

$$\alpha = \frac{-1}{(1-s_A)m_A p_A - r(\sum_{j=B,C} s_j)} \tag{3.10}$$

[11] Setting the upper bound of the support on r to m_1 ensures the model is well-defined, specifically, that $\alpha < 0$.

5. From the licensees' first order conditions, back out remaining margins as,

$$m_j = \frac{1}{p_j}\left(\frac{-1}{\alpha(1-s_j)} + r\right) \text{ for } j = B,C \tag{3.11}$$

6. Determine implied "true" costs for licensor and licensees as,[12]

$$c_j = p_j(1-m_j) \text{ for } j = A,B,C \tag{3.12}$$

7. Given prices, p, using first order conditions that do not account for the licensing relationships back out the mis-specified marginal costs, $\tilde{c}_j$, as in Equation (3.3).

8. Conduct counterfactual simulation with the mis-specified model and $\tilde{c}_j$.

9. Conduct counterfactual simulation with the true model and true c_j.

10. Compute the prediction bias in merger effects, $\Delta\Delta y$.

11. Repeat Steps 1–10 until a total of 1,000 sample markets are obtained.

Guidance Simulation Sample. In order to understand the scope of markets comprising the sample generated from the above outlined procedure, we detail the key market characteristics in the sample. Table 3.2 documents the empirical distributions of these market characteristics for the generated sample. While royalty rates, r, and the licensor's margin, m_A, are drawn uniformly (see Guidance Simulation Section), we drop markets for which model primitives imply negative costs, inducing a non-uniform observed distribution in the sample over these primitives. Under Nash-Bertrand competition with a logit demand system, the licensees' costs are decreasing in both royalty rates and the licensor's margin. Thus, each respective empirical distribution is shifted left relative to parameter's drawn upon uniform distribution. For example, royalty rates are drawn uniformly from zero to one, however the empirical distribution of markets with non-negative implied costs is centered around 0.225 with a right tail. Similarly, the empirical distribution of the licensor margin is slightly left-shifted relative to the uniform distribution over [0.25, 0.75], from which this parameter is initially drawn.

The resulting sample exhibits variation in concentration, as quantified by market shares and HHI, margins, costs, and market elasticity. Looking ahead to the counterfactuals that we will evaluate under this framework – a merger between licensor and licensee (i.e., AB) and a merger

[12]Markets with any negative costs (i.e., $c_j < 0$ for any j) are discarded. This can occur when r is large.

Table 3.2: Sample Statistics

Percentile	5%	25%	50%	75%	95%
Royalty Rate (r)	0.019	0.11	0.225	0.353	0.538
Shares:					
s_A	0.046	0.172	0.267	0.342	0.483
s_B	0.035	0.145	0.246	0.33	0.469
s_C	0.034	0.15	0.246	0.333	0.444
Total Inside Share	0.75	0.75	0.75	0.75	0.75
HHI:					
Pre-Merger	1895	2025	2256	2617	3274
Post-AB Merger	2636	2799	3026	3553	4721
Δ from AB Merger	145	505	760	1167	2117
Post-BC Merger	2558	2633	2737	2924	3534
Δ from BC Merger	72	356	527	615	692
Margins (Pre-Merger):					
m_A	0.275	0.366	0.498	0.606	0.708
m_B	0.302	0.414	0.567	0.714	0.891
m_C	0.293	0.429	0.57	0.712	0.898
Share-Weighted Avg.	0.286	0.401	0.539	0.673	0.862
Costs:					
c_A	0.292	0.394	0.502	0.634	0.725
c_B	0.333	0.577	0.701	0.79	0.864
c_C	0.361	0.578	0.704	0.791	0.869
Share-Weighted Avg.	0.326	0.486	0.602	0.701	0.789
Market Elasticity	-7.187	-4.625	-3.369	-2.494	-1.707

Notes: Market elasticity under logit demand is given by $\alpha\bar{p}(1-s_0)$, where $\bar{p}$ represents the share-weighed average price

between the two licensors (i.e., BC) – the resulting changes in market concentration vary substantially.[13]

[13]The 2010 DOJ/FTC Horizontal Merger Guidelines indicate scrutiny for horizontal mergers resulting in moderately or highly concentrated markets (post-merger HHI greater than 1500) with changes of HHI greater than 100 points.

3.4 Simulation Results

3.4.1 The Simulated Bias in Marginal Cost Estimation

This subsection presents the simulation results for estimation bias in marginal costs.

Theoretical Simulation Results. Figure 3.1 shows the theoretical simulation results on marginal cost estimation bias in a heat-map. The horizontal axis is the sum of diversion ratios between the licensor A and licensees B and C, i.e. $-\left(\frac{\partial Q_B/\partial p_A}{\partial Q_A/\partial p_A}+\frac{\partial Q_C/\partial p_A}{\partial Q_A/\partial p_A}\right)$. The vertical axis is the royalty rate r charged by the licensor, and also the fraction of r over product-level total marginal cost, since total marginal costs are normalized to 1. The color shows the exact value of the estimation bias in marginal costs of the licensor, $\Delta c_A := (\tilde{c}_A - c_A)/c_A$. It is not surprising that the features of Δc_A are consistent with the features of the alignment effect, because in the pre-merger market, the licensor is a single-product firm, the estimation bias is equal to the alignment effect. In particular, consistent with the fact that the alignment effect is always non-negative, the estimation bias in the licensor's marginal costs is always positive. This implies that ignoring existing patent licensing relationships will lead to over-estimated marginal costs of the licensor. Moreover, consistent with the increasing monotonicity of the alignment effect with respect to royalty rates and the sum of diversion ratios between the licensor and the licensees, the estimation bias is also increasing with respect to these two model features. The average bias in estimated marginal costs from the mis-specified model is 25.28% of the true marginal costs. [14]

Guidance Simulation Results. As described in Step 8 of the Guidance Simulation Procedure, costs are backed out for the mis-specified model for each firm. We compare these costs, $\tilde{c}$, to the true costs implied by a model correctly accounting for the licensing relationships between firms. Given licensee first order conditions are identical between true and mis-specified models, the implied licensee costs are the same between the two models, i.e., $c_j = \tilde{c}_j$ for $j = B, C$. However, for the licensor, the alignment effect term, μ_A, is not accounted for in the mis-specified model, and thus implied licensor costs diverge between the two models. Table 3.3 and Figure 3.2 show the distribution of the bias in estimation of licensor marginal costs from the simulation sample. In all simulated markets, licensor marginal costs are overestimated in the mis-specified model relative to those consistent with the true model.

As detailed in Table 3.3, estimates of the licensor's costs under a mis-specified model can lead to significant differences from the true costs. At the median, estimates of the licensor's costs under the mis-specified model differ by 28% from those implied by the true model. Even at the 5th percentile, costs estimates differ by a non-nelgible amount (i.e., 2.5%).

[14] We report estimation bias of licensee's marginal costs in Appendix D, Figure G.15. Consistent with the theoretical framework, the bias are extremely small, and there is no pattern with respect to royalty rates and diversion ratios. Thus, we interpret these "estimation bias" as computation errors.

Figure 3.1: Theoretical Simulation: marginal cost estimation bias and royalty rates and the sum of diversion ratios between the licensor and the two licensees.

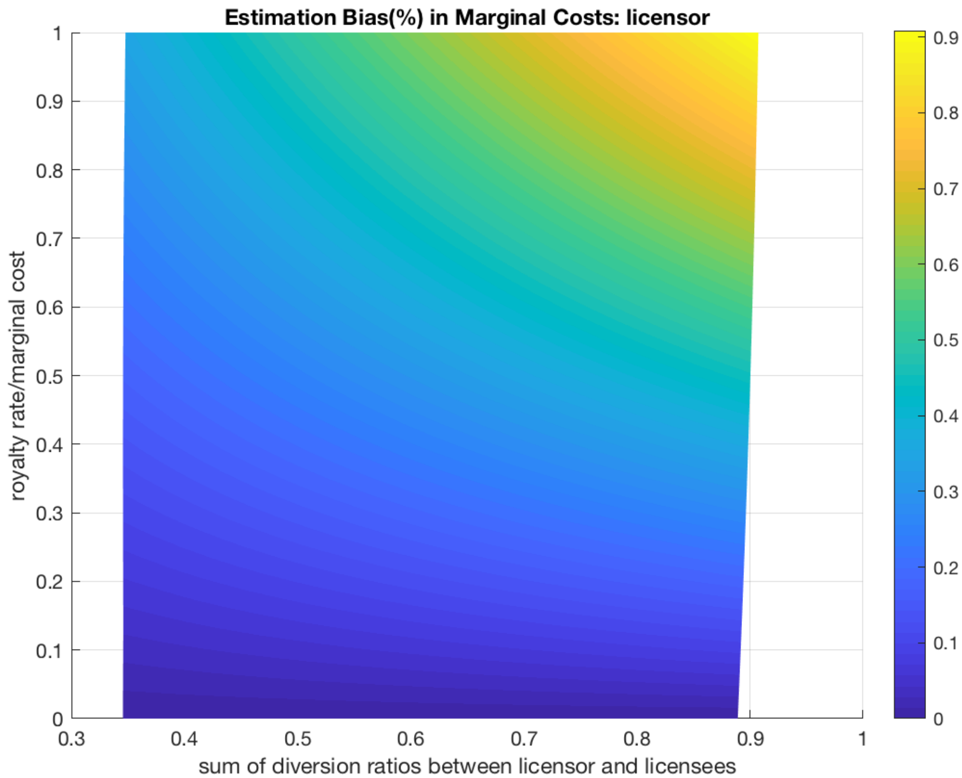


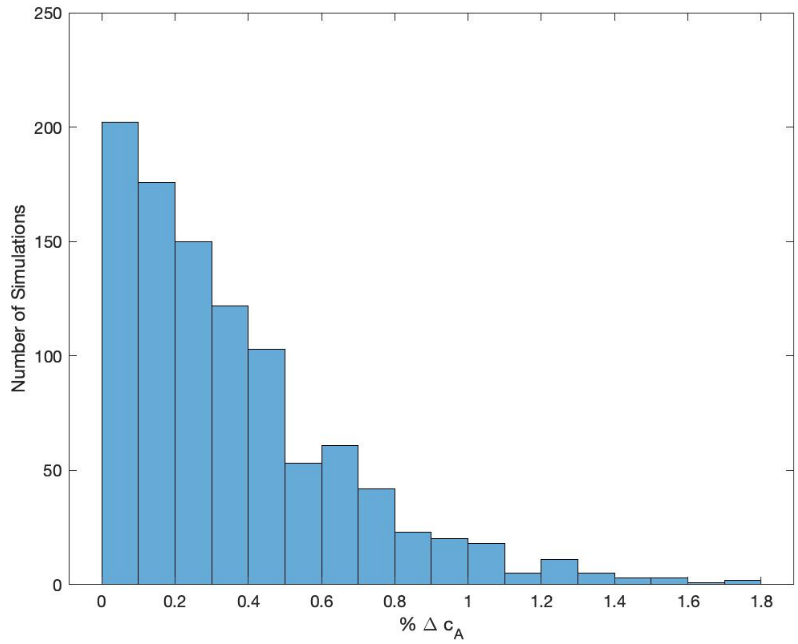
Table 3.3: Bias in Cost Estimation

Percentile	5%	25%	50%	75%	95%
$\%\Delta c_A$	0.025	0.13	0.28	0.493	0.987

Notes: Bias in cost estimation is defined as the percentage difference between costs implied by the mis-specified model and the true costs, i.e., $\%\Delta c_j = (\tilde{c}_j - c_j)/c_j$. Given licensees' first order conditions are identical between the mis-specified and true models, $\%\Delta c_B = \%\Delta c_C = 0$.

In order to understand when we would expect this bias in cost estimation to be larger in real world markets, we can assess how the bias varies by underlying model primitives – specifically, the licensing royalty rate, r, and the sum of the diversion ratios between the licensor and each respective licensee. Given the assumed exogeneity of the royalty rate, r, we would expect size of the cost estimation bias to scale with the royalty rate. Indeed, in Figure 3.3, we see that, for a given sum of diversion ratios between the licensor and each respective licensee, the greater the royalty

Figure 3.2: Guidance Simulations: estimation bias in costs between mis-specified and true models for the licensor (firm A)



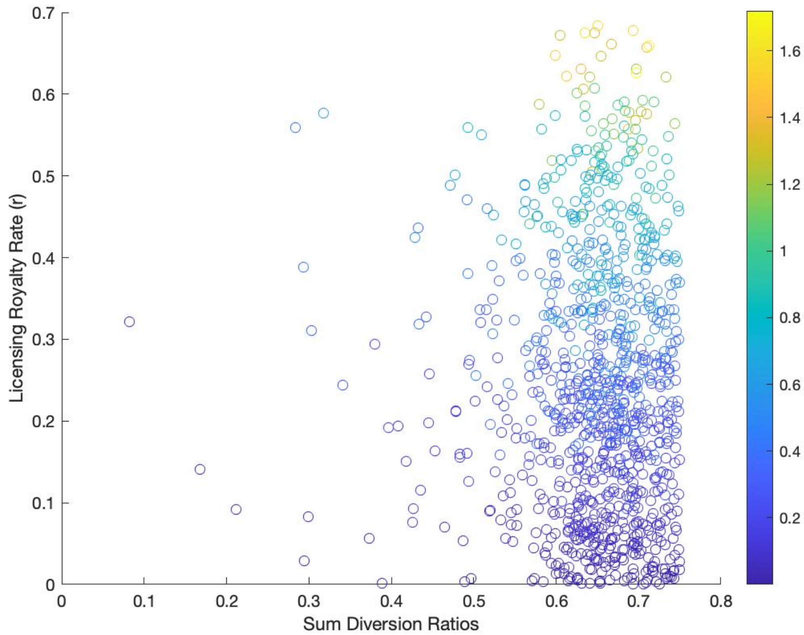
rate, the larger the bias in estimation of licensor costs.[15]

Given this set of guidance simulations is aimed at understanding in what real world markets would we expect larger differences between the estimates of a mis-specified model and the true model, we analyze these cost estimates relative to a common measure of market concentration – HHI. In Figure 3.4, we see that in less concentrated markets (i.e., smaller HHI), there is larger variance in the percentage difference in estimated licensor marginal costs between mis-specified and true models. While in more concentrated markets, there is smaller variance in cost estimate differences, the magnitude is still relatively large. For example, on the right-hand side of Figure 3.4, one relatively concentrated simulated market is observed with a pre-merger HHI of 4523 while exhibiting a relatively large estimated licensor cost difference of 50.62%.

Understanding that, in these simulated markets, implied licensor costs can be significantly overestimated in a model that does not account for licensing, in the next sections, we look to understand the implications of this overestimation on counterfactual equilibrium objects of interest to researchers, policy makers, and antitrust authorities.

[15]This correlation between royalty rate levels and the bias in licensor cost estimation is additionally demonstrated in Appendix D, Figure G.16 where cost biases are plotted against royalty rates.

Figure 3.3: Guidance Simulations: estimation bias in costs between mis-specified and true models for firm A (licensor) by model parameters



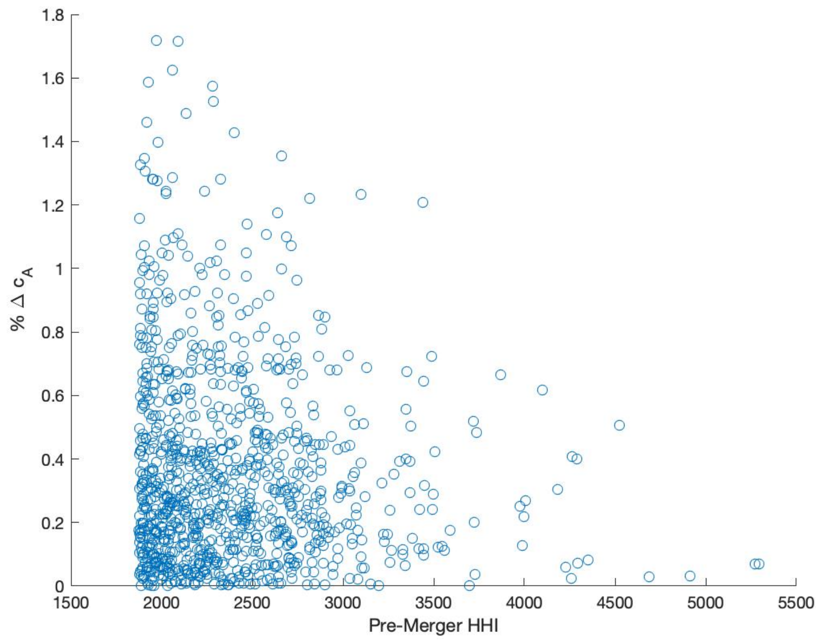
3.4.2 Prediction Bias in Licensor-Licensee Mergers

This subsection presents the simulation results for prediction biases in effects on prices, market shares, consumer welfare, and producer surplus after a licensor-licensee merger.

Theoretical Simulation Results. Figure 3.5 shows the results on the prediction biases in the effects of mergers between the licensor, A, and licensee B. The horizontal axis is the sum of the diversion ratios between the licensor A and the two licensees. The vertical axis is the royalty rate r. The colors show the prediction biases. Panel(a) shows the prediction biases in consumer welfare effects of the mergers. Panel (b) shows the prediction biases in the merger effects on sales-weighted average price. Panel(c) shows the prediction biases in the merger effects on total quantities. Panel (d) shows the prediction biases in the merger effects on producer surplus.

Overall, the results in Figure 3.5 show that i) the higher the royalty rates, the larger the magnitude of the prediction bias, and ii) the relationship between diversion ratios and the prediction bias is non-monotonic. Given a level of the sum of diversion ratios, panel (a) shows that the magnitude of the prediction bias in the consumer welfare effects of the merger increases with royalty rate. Similar patterns exist from panel (b) to (d), including merger effects on sales-weighted average price, total quantities and producer surplus. Given a royalty rate, panel(d) shows that the prediction bias on the merger effects on producer surplus increases with the sum of diversion ratios.

Figure 3.4: Guidance Simulations: estimation bias in costs between mis-specified and true models for firm A (licensor) by HHI



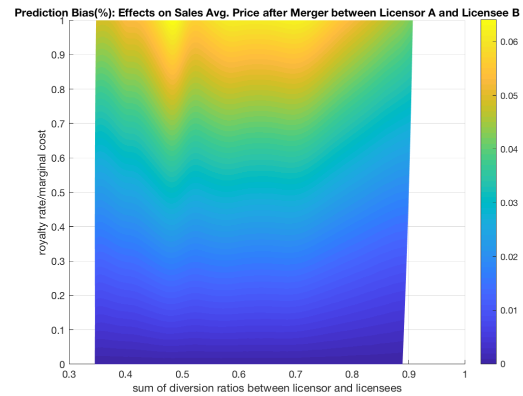
However, panel (a) shows that, given a low level of royalty rate, the magnitude of the prediction bias in the consumer welfare effects typically decrease as the sum of diversion ratios increases. But at high levels of royalty rates, such as 80% of the licensees' marginal costs, the prediction bias might reach the largest magnitude when the sum of diversion ratios is at middle range such as 0.5. Similar patterns show up in the panel (c). Panel (b) shows an even more obvious non-monotonic relationship between the prediction bias and the sum of diversion ratios, when it comes to the prediction of merger effects on sales-weighted average price. In particular, given a level of royalty rate, the prediction bias in panel (b) firstly increases then decreases as the sum of diversion ratios increases.

The above results have two implications. First, when predicting merger effects on consumer surplus, sales-weighted average price and total quantities, it's better to be careful if the royalty rates are high and diversion ratios are in the middle range. Second, estimation bias in marginal costs is not the deterministic factor for prediction biases. As estimation bias in marginal costs always increase with the sum of diversion ratios as shown in Figure 3.1, we see non-monotonic relationship between the prediction biases and the sum of diversion ratios. This is consistent with the theoretical analysis of $\Delta p'_j$ in Equation 3.4.

Table 3.4 provides summary statistics of the theoretical simulation results on the prediction bias, conditional on positive royalty rates. Each row is a market outcome of general interests. The first

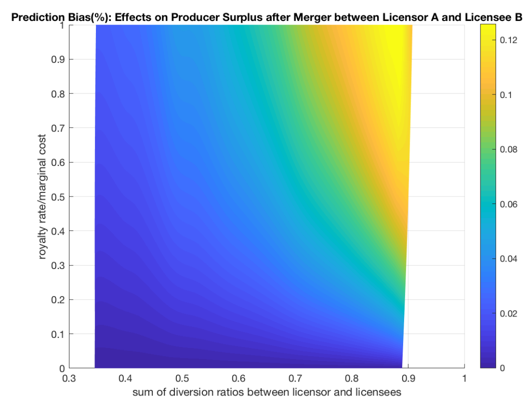
Figure 3.5: Theoretical Simulation: prediction bias in the effects of the merger between licensor and licensee.

(a) Bias in Consumer Welfare Effect



(b) Bias in the Merger Effect on Sales-weighted Average Price

(c) Bias in the Merger Effect on Total Quantities



(d) Bias in the Merger Effect on Producer Surplus

column reports the average prediction bias. The second column reports the average true merger effects, measured in the percentage change of a market outcome relative to its pre-merger levels. The third column reports the percentage of cases that generate over-predicted merger effects in magnitudes during the simulation. Recall that each case is a pair of royalty rates and the travel parameter, i.e. (r, ρ). The last column reports the percentage of cases where the mis-specified model predicts opposite direction of merger effects during the simulation. The results are conditional on positive royalty rates, so that ignoring patent licensing relationship or not is non-trivial.

Column (2) of Table 3.4 shows that the simulated merger effects are economically reasonable. On average, the true model with patent licensing relationships predicts decrease in consumer surplus, increase in all prices, decrease in the quantities of the merged products and increase in the

Table 3.4: Theoretical Simulation: Predict the Effects of Licensor-Licensee Merger

	(1) avg. prediction bias (%)	(2) avg. true effect (%)	(3) over-prediction (%)	(4) opposite prediction (%)
Consumer Surplus	−2.414	−10.668	100	0
Sales-weighted Avg. Price	2.655	8.283	100	0.03
Price: A	0.65	16.334	98.812	0
Price: B	6.237	8.302	100	4.307
Price: C	0.887	2.612	100	0
Total Quantities	−1.761	−5.85	100	0
Quantity: A	1.218	−18.916	0	0
Quantity: B	−7.664	−4.987	99.594	25.228
Quantity: C	1.249	5.577	100	0
Producer Surplus	2.773	4.715	100	0
Profit: A+B	2.206	2.516	100	0
Profit: C	2.635	9.65	100	0

Notes: results are conditional on positive royalty rates. All simulated mergers are profitable: the smallest true merger effect on the joint profit of firms A and B is 0.83 percent.

quantity of the non-merging product, and increase in all profits after the licensor-licensee merger.

At market level, Column (1) of Table 3.4 shows that, on average, the mis-specified model predicts more loss in consumer welfare by 2.4%, more increase in sales-weighted average price by 2.7%, more decrease in total quantities by 1.8% and more increase in producer surplus by 2.8% after the licensor-licensee merger. At product level, Column(1) of Table 3.4 shows that, the market-level overly predicted post-merger price increase is driven by the overly predicted post-merger price increase of the merged licensee's product, B. As prices are strategic complements, it's not surprising to see that post-merger prices of the licensor's good, A, and the non-merging licensee's good, C, are also overly predicted by the mis-specified model. The largely over-predicted post-merger price B also leads to largely over-predicted post-merger decrease in quantity B, which leads to the overall over-predicted post-merger decrease in quantities at the market level.[16] At product-level, we see over-predicted post-merger profit increase as what we see at market-level.

[16]As for the other merged product, A, while its post-merger price are over predicted (i.e. higher post-merger price A), it's quantity decrease is not over predicted and practically always under-predicted. This might be due to the relative over predictions in prices of A, B and C. Note that price of B is the most over-predicted price. Therefore, good A will be predicted more attractive to consumers than good B in the mis-specified model, which leads to the less decrease in quantity A. Similar reasons explain why the non-merging product, C, also see over-predicted post-merger price increase, and higher post-merger downloads predicted from the mis-specified model.

These results are consistent with column (3) of Table 3.4, which shows that in almost all of the cases, merger effects are over-predicted in the mis-specified model.[17]

The last column of Table 3.4 shows the existence of opposite prediction when the prediction is based on mis-specified model ignoring existing patent licensing relationship. This is alarming to researchers. It shows 3% of simulation cases see opposite predictions on merger effect on sales-weighted average price. It further shows that the opposite prediction in market-level average price roots in opposite prediction in price B; and the opposite prediction in price B also leads to opposite prediction in Quantity B.

We provide more details on the opposition predictions in Table 3.5. Column (1) shows the average royalty rates where the opposition prediction on each market outcome happens. Column (2) shows the corresponding average sum of diversion ratios. Column (3) shows the corresponding average travel parameter. Column (4) shows the corresponding average true merger effects. Column (5) shows the corresponding average mis-specified merger effects. Column (6) shows the corresponding average prediction bias.

To answer where the opposite predictions happen, Column (1) of Table 3.5 shows that, the opposite prediction cases mainly happen in relatively extreme cases: royalty rates accounts for more than 85% of total marginal costs. This is good news. Column(2) and (3) show that, to have opposite predictions on market-level sales-weighted average price, we also need extremely high substitutability between products. However, only middle substitutability between products are needed to generate opposite predictions on product-level price and quantity, conditional on extremely high royalty rates.

To answer how the opposite prediction happens, column(4) and (5) of Table 3.5 shows that the reason is that the mis-specified model ignores cost-saving of product B after its merger with the licensor. In particular, column (4) shows that the opposite prediction of the merger effect on sales-weighted average price happens because the true merger effects actually lead to decrease in the sales-weighted average price by 0.16% on average. Such lower price after merger happens because the merged licensee, B, does not need to pay royalty rates to the licensor, A, after the merger. With the cost saving, product B can enjoy a 0.78% lower price on average. However, column (5) shows that, the mis-specified model cannot account for such cost reduction channels, and thus, predicts that the price of product B will increase by 10.8% after the merger, on average. Such discrepancy in predicted post-merger price of B also leads to opposite predictions on post-

[17] We additionally note that, given no opposite prediction on the merger effects on price of A, p_A, over-prediction is the same as $\Delta p'_A > 0$. Therefore, the 98.8% of over-prediction cases imply that in most of the cases, the ignored alignment effect in Equation 3.4 is not driving the prediction biases in post-merger price of the licensor-manufacturer. However, given the averagely negative true merger effect and positive prediction bias on quantity of A, $\tilde{Q}'_A > Q'_A$, and thus $\tilde{v}'_A > v'_A$. Therefore, both the first and the second terms in Equation 3.4 are positive, which disenables us to conclude that marginal cost estimation bias is driving the prediction bias.

Table 3.5: Theoretical Simulation: Mis-specified Model Predicts Opposite Licensor-Licensee Merger Effects

	(1) royalty rates	(2) diversion ratios	(3) travel parameters	(4) true effects	(5) mis-specified effects	(6) prediction biases
	$\bar{r}$	$\bar{d}$	$\bar{\rho}$	$\overline{\%\Delta y}$	$\overline{\%\Delta \tilde{y}}$	$\overline{\Delta\Delta y}$
Sales-weighted Avg. Price	0.99	0.907	1	−0.159	3.931	4.090
consumer surplus				−1.067	−2.819	−1.752
total quantities				−0.712	−1.898	−1.186
producer surplus				2.363	12.970	10.606
Price: B	0.947	0.402	0.313	−0.778	10.835	11.613
consumer surplus				−6.100	−11.101	−5.002
sales-weighted avg. price				4.671	9.596	4.925
total quantities				−4.272	−7.863	−3.591
producer surplus				1.970	4.538	2.568
Quantity: B	0.859	0.464	0.422	2.733	−10.336	−13.069
consumer surplus				−6.491	−10.797	−4.307
sales-weighted avg. price				5.493	10.051	4.558
total quantities				−4.557	−7.666	−3.109
producer surplus				2.629	6.235	3.606

Note: results are conditional on opposite predictions.

merger quantities of B between the true model and the mis-specified model: on average, the true model predicts 2.7% increase of the quantity of product B, while the mis-specified model predicts 10.3% decrease of the quantity of product B. Now, the close relationship between the opposite prediction and high royalty rates are intuitive: a higher royalty rate implies larger cost reduction un-captured by the mis-specified model.

To answer how important are those opposite-prediction cases, we compare column (4) to (6) in Table 3.5 to column (1) and (2) in Table 3.4. Comparing column (1) of Table 3.4 to column (6) of Table 3.5, it shows that in the cases of opposite predictions, the prediction bias are larger in magnitudes. Comparing column (2) of Table 3.4 to column (4) of Table 3.5, it shows that in the cases of opposite predictions, the true merger effects are smaller in magnitudes. More importantly, we note that the average prediction bias on price of B is 11.61% conditional on opposite predictions, while the unconditional average prediction bias is 6.23% as reported in Table 3.4. Since the average prediction bias of merger effects on price B is driving the market-level prediction biases on prices, we conclude that the ignored royalty payment saving due to licensor-licensee merger is driving the prediction bias in licensor-licensee merger.

Guidance Simulation Results. We can analyze the bias in counterfactual predictions between the true and mis-specified models under a horizontal merger between the licensor ($j = A$) and one of the licensees (without loss, $j = B$). From these simulations, we find that, in general, due to the overestimation of licensor costs from not accounting for the licensing relationship, as detailed above in Section 3.4.1, the mis-specified model predicts inflated counterfactual welfare measures. Particularly, relative to the true model, the mis-specified model predicts larger welfare losses for consumers and greater welfare gains for producers following a merger between the licensor and a licensee.

To assess for what characteristics of real world markets we would expect to find larger biases in counterfactual prediction, we compare the biases in counterfactual measures across a rich set of simulated markets that span the space of reasonable underlying model primitives. This allows examine where the bias in counterfactual predictions is largest, and thus cause for concern for market analysts, across generated markets with different distributions of market shares, margins, and royalty rates.

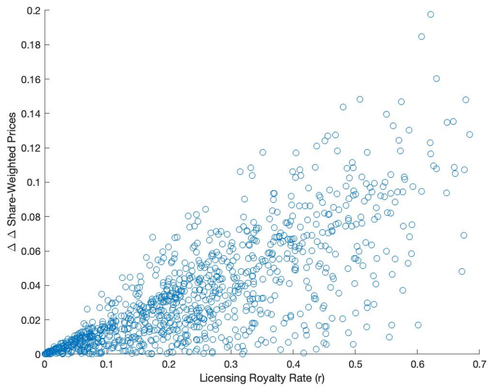
To explore when prediction biases are largest and provide guidance to analysts evaluating markets with potentially unobserved licensing behavior, we plot the bias in a given counterfactual measure derived for each market in our sample against the distribution of a underlying model parameter.

First, we show how the bias in key counterfactual measures vary with the licensing royalty rate, r. Figure 3.6a plots the relationship between a market's royalty rate, r, and the bias in reported counterfactual-factual change in share-weighted average prices between the mis-specified and true model. Each circle represents a generated market with its associated values plotted in r-$\bar{p}$ space. It can be seen from the strictly positive biases (i.e., only positive values on the y-axis) that the mis-specified model exhaustively overestimates the percent change in share-weighted average prices, relative to the true model (see also the top panel of Table 3.11).[18] Further, moving from left to right within Figure 3.6a, we see that markets that exhibit a larger royalty rate are correlated with a larger bias in predicted change in share-weighted average price, with heterogeneity in the prediction bias at larger values of r. Therefore, in real world markets, we would expect markets with larger royalty rates to have the potential to exhibit larger errors in reported prices from a licensor-licensee merger under a mis-specified model and thus more reason for concern for market analysts.

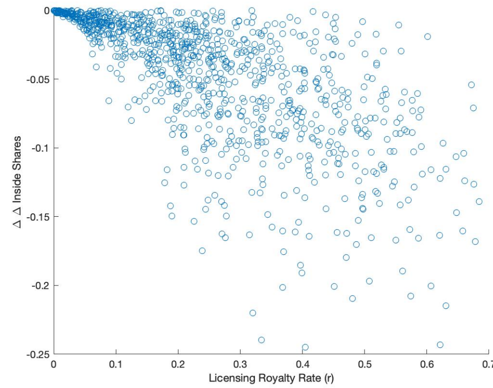
Given that mis-specified model exhaustively predicts a larger change in share-weighted prices than the true model and that larger magnitudes of this overprediction (particularly the maximum bias) are associated with royalty rates, we would expected the mis-specified model to predict larger

[18]In all simulated markets, the mis-specified model predicts an increase in share-weighted average prices due to the licensor-licensee merger, while the true model does so for only 63.7% of markets.

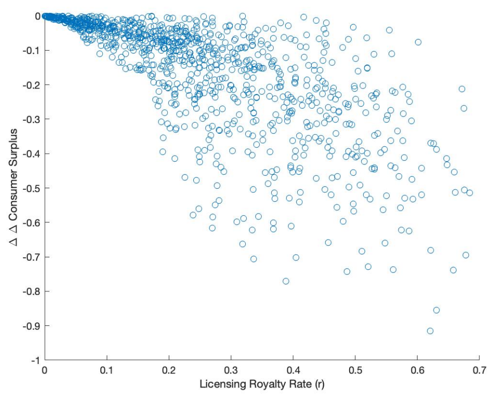
Figure 3.6: Guidance Simulations: prediction bias in the effects of the merger between licensor and licensee by licensing royalty rate, r.



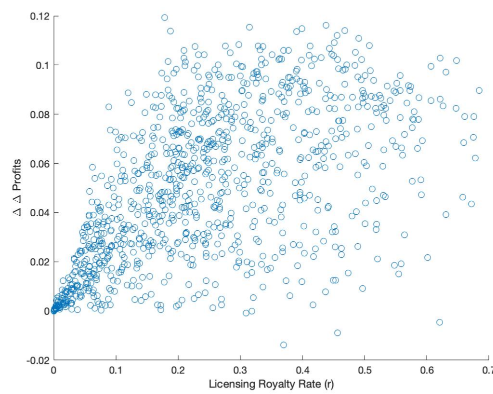
(a) Share-Weighted Prices



(b) Inside Shares



(c) Consumer Surplus

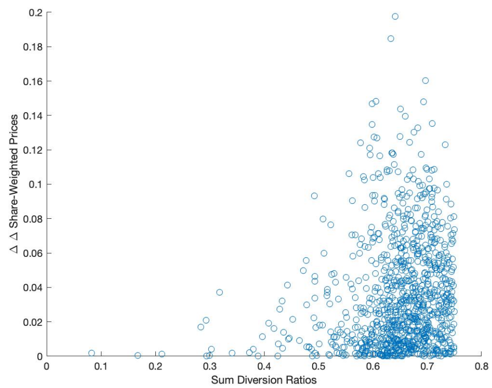


(d) Producer Surplus

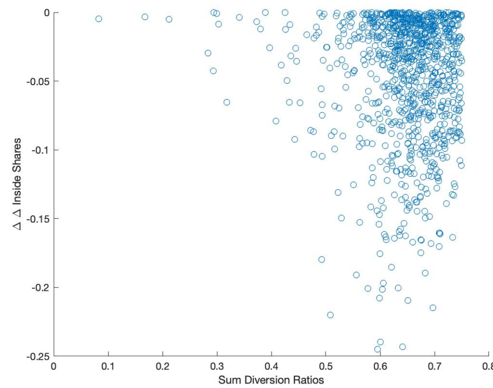
changes in the shares of the inside goods (i.e., $\sum_{j=A,B,C} s_j$) and that the magnitude of this overprediction increases in r. Indeed, this relationship is exhibited in Figure 3.6b – when royalty rates are larger, we see that the mis-specified model generally predicts a larger decrease in total inside shares relative to the true model.[19] Given the overprediction of price increases and the associated overprediction of inside share decreases, we would generally expect an overprediction of consumer welfare losses as the royalty rate increases. In our sample, this relationship holds, as shown in Figure 3.6c, which plots the bias in predicted changes in consumer surplus between the mis-specified and true models against a market's royalty rate, r. Finally, predictions for change in producer profits due to the licensor-licensee merger seem to follow a less clear trend, in terms of the correlation

[19] The mis-specified model underpredicts post-merger shares for Firm B (i.e., the merging licensee), while it overpredicts shares for Firm A (i.e., the merging licensor) and Firm C (i.e., the non-merging licensee). See Table 3.11

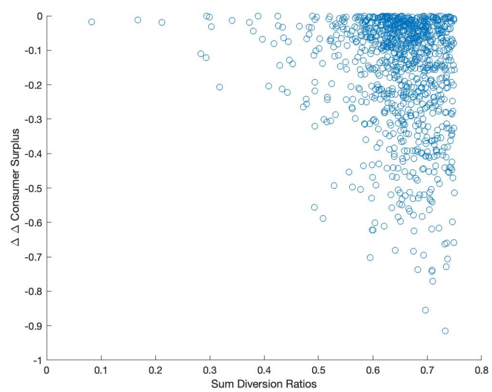
Figure 3.7: Guidance Simulations: prediction bias in the effects of the merger between licensor and licensee by licensing sum of diversion ratios between licensor and licensees.



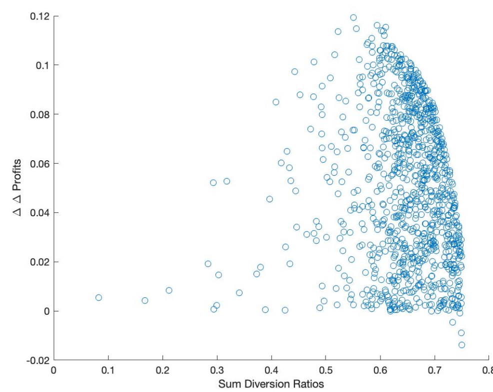
(a) Share-Weighted Prices



(b) Inside Shares



(c) Consumer Surplus



(d) Producer Surplus

with royalty rates, as exhibited in Figure 3.6d, however are largely overpredicted. Overall, these results with respect to the market's royalty rate suggest that for markets with a larger royalty rate, the analysis of a merger between the licensor and a licensee under a model that does not account for licensing behavior can critically overpredict pre- to post-merger changes in key welfare measures.

We can additionally assess the mis-specified model's counterfactual predictions in the simulated markets based on the sum of the diversion ratios between the licensor and each licensee. Figure 3.7 shows the relationship between the same biases in predicted changes in welfare measures as Figure 3.6 – share-weighted average prices, total inside market shares, consumer surplus, and producer surplus – now against the model's implied sum of diversion ratios between the licensor and the two respective licensees. The trend in prediction bias is less clear, however we can observe that maximum (i.e., frontier) over- or underprediction does exhibit an association with the sum of

Table 3.6: Bias in Licensor-Licensee Merger Prediction

	5%	25%	50%	75%	95%
Prices:					
p_A	−0.009	0	0.002	0.005	0.013
p_B	0.007	0.041	0.08	0.126	0.2
p_C	0	0.002	0.005	0.009	0.018
Share-Weighted Avg.	0.001	0.009	0.026	0.053	0.098
Shares:					
s_A	0.001	0.013	0.044	0.12	0.254
s_B	−1.037	−0.486	−0.228	−0.092	−0.011
s_C	0.002	0.018	0.058	0.152	0.357
Inside Shares	−0.139	−0.072	−0.032	−0.011	−0.001
Profits:					
π_A	0.016	0.131	0.289	0.454	0.633
π_B	−4.415	−1.972	−0.813	−0.307	−0.035
π_C	0.003	0.027	0.083	0.197	0.403
Producer Surplus	0.003	0.024	0.048	0.075	0.1
Consumer Surplus	−0.509	−0.235	−0.093	−0.033	−0.004

Notes: Post-merger licensing revenues allocated to π_A.

diversion ratios measure.[20] For example, Figure 3.7a shows that, for larger sum of diversion ratios, the maximum overprediction of share-weighted average prices is generally increasing. Similarly, the maximum overprediction in pre- to post-merger decreases in inside shares and the maximum overprediction in pre- to post-merger decreases in consumer surplus are generally increasing in the sum of the royalty rate. The relationship between the maximum prediction error in changes in producer surplus is non-monotonic in the sum of diversion ratios. Specifically, at smaller values of the sum of diversion ratios, the maximum prediction bias in producer surplus is increasing; however, at larger values, this frontier is decreasing. For market analysts, this non-monotonicity suggests concern for larger overpredictions in producer surplus lie in the interior of feasible sum of diversion ratios, while larger prediction biases for share-weighted average prices, total inside shares, and consumer surplus occur at high values of the sum of diversion ratio term.

Not only are we interested where in the model primitive space do the magnitude of predictions diverge between the mis-specified and true models, but also when they predict changes in welfare measures of the opposite sign. If, for example, a mis-specified model is used to evaluate a merger

[20]Given the simulated outside market shares are calibrated at $s_0 = 0.25$, the maximum possible sum of diversion ratios is $1 - s_0 = 0.75$.

Table 3.7: Average Bias and Opposite Predictions in Licensor-Licensee Mergers

	(1) Avg. True Effect	(2) Avg. Prediction Bias	(3) Opposite Prediction
Prices:			
p_A	0.099	0.002	0
p_B	0.014	0.089	0.499
p_C	0.003	0.006	0.326
Share-Weighted Avg.	0.02	0.035	0.363
Shares:			
s_A	−0.269	0.077	0.011
s_B	0.126	−0.342	0.609
s_C	0.049	0.103	0.326
Inside Shares	−0.02	−0.047	0.327
Profits:			
π_A	−0.272	0.3	0.702
π_B	1.384	−1.367	0.245
π_C	0.057	0.13	0.326
Producer Surplus	0.029	0.05	0.091
Consumer Surplus	−0.05	−0.157	0.327

Notes: Average true effect is given by percent change. Average prediction bias is given by the difference in percent changes between the mis-specified model. Opposite prediction represents a different sign in counterfactual prediction of a given measure between the true and mis-specified model. Only 0.5 percent of simulated licensor-licensee mergers are non-profitable. Thus, the results conditional on profitable licensor-licensee mergers are quite similar, as shown in Table F.13.

that (incorrectly) predicts losses in consumer surplus, whereas a correctly-specified model would predict consumer surplus gains, an analyst may critically arrive at an opposite policy recommendation against allowing the merger occur. In Table 3.7, we summarize the predictions in welfare measures between the true and mis-specified models across the generated sample of markets. In the top panel, first column, we see that, on average, share-weighted prices increase by 2% following the merger between licensor (Firm A) and licensee (Firm B). This increase in share-weighted average prices is larger driven by increases in Firm A's product, which is 9.9% higher than pre-merger levels, on average. The mis-specified model overpredicts share-weighted average prices by 3.5% on average, largely due the overprediction of post-merger price of Firm B (i.e., the merging licensee). This follows from the intuition that Firm B, by merging with the licensor, no longer needs to pay the royalty rate, which reflects a cost savings that induces downward pressure on Firm B's post-merger price. Given the mis-specified model does not account for the pre-merger licensing relationship, no cost savings are exhibited and, thus, the mis-specified model overpredicts Firm B's price increase, on average by 8.9 percentage points.

In the generated sample, the true model predicts increase in share-weighted prices due to the licensor-licensee merger 63.7% of the time, while the mis-specified model always predicts higher post-merger prices. In the remaining 36.3% of sample markets, the true model predicts a decrease in share-weighted average prices, with the mis-specified model predicting the opposite effect in these markets, as shown in the third column of Table 3.7.

We see corresponding trends in the average effects, biases, and opposite predictions in terms of market shares. The true model predicts, on average, a decrease in inside shares, driven by large decreases Firm A's market share. On average, the mis-specified model overpredicts this decrease in total inside market shares by 4.7% and predicts an opposite change in the inside total share in 32.7% of markets.

The true model that accounts for licensing predicts, on average, a 5.0% decrease in consumer surplus due to the licensor-licensee merger. Largely driven by the inability to account for the cost savings from the merging licensee, and the associated prediction of a higher post-merger price for Firm B, the mis-specified model overpredicts the loss in consumer surplus by 15 percentage points. Critically, the true model predicts increases in consumer surplus in 32.7% of markets, while the mis-specified model exclusively predicts consumer surplus losses. In other words, the mis-specified model predicts an opposite effect on consumer for approximately one third of markets in our simulated sample. Given modern merger enforcement's focus on consumer welfare to drive policy recommendations, in these markets with opposite consumer surplus predictions, a mis-specified model that does not take into account licensing behavior would lead to an errant recommendation.

What are the characteristics of the markets in which the true and mis-specified model predict opposite effects? In Table 3.8, we detail the conditional distribution of market characteristics for markets that exhibit opposite predictions between the true and mis-specified model for at least one welfare measure (i.e., prices, shares, profits, or consumer surplus). Relative to the full distribution of the sample given in Table 3.2, markets that exhibit an opposite prediction in at least one welfare measures by the mis-specified model generally exhibit: larger royalty rates, larger licensee shares and margins, higher concentration both pre- and post-merger, larger costs, particularly for licensees, and more elastic demand.

3.4.3 Prediction Bias in Licensee-Licensee Mergers

This subsection presents the results for simulated biases in predicted effects on prices, market shares, consumer welfare, and producer surplus after a licensee-licensee merger.

Table 3.8: Opposite Prediction Market Characteristics

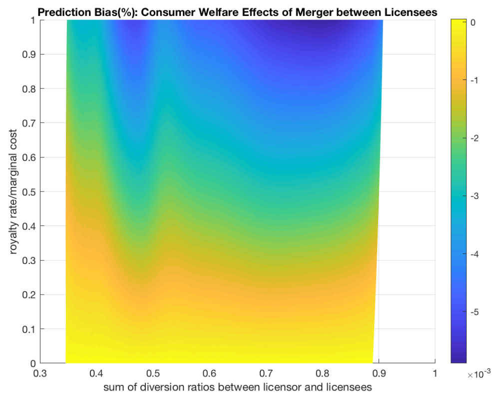
Percentile	5%	25%	50%	75%	95%
Royalty Rate (r)	0.168	0.27	0.372	0.486	0.603
Shares:					
s_A	0.017	0.098	0.204	0.297	0.429
s_B	0.041	0.163	0.271	0.354	0.509
s_C	0.063	0.188	0.278	0.354	0.518
Total Inside Share	0.75	0.75	0.75	0.75	0.75
HHI:					
Pre-Merger	1894	2029	2324	2700	3360
Post Merger	2858	3058	3395	3956	5263
Δ from Merger	133	603	1118	1651	2536
Margins (Pre-Merger):					
m_A	0.267	0.362	0.483	0.576	0.68
m_B	0.351	0.479	0.627	0.784	0.92
m_C	0.359	0.481	0.645	0.771	0.919
Share-Weighted Avg.	0.339	0.472	0.618	0.749	0.87
Costs:					
c_A	0.32	0.424	0.517	0.638	0.733
c_B	0.516	0.689	0.772	0.835	0.889
c_C	0.457	0.694	0.775	0.826	0.892
Share-Weighted Avg.	0.447	0.582	0.664	0.75	0.823
Market Elasticity	−8.546	−5.797	−4.531	−3.5	−2.474

Notes: Market elasticity under logit demand is given by $\alpha\bar{p}(1-s_0)$, where $\bar{p}$ represents the share-weighed average price

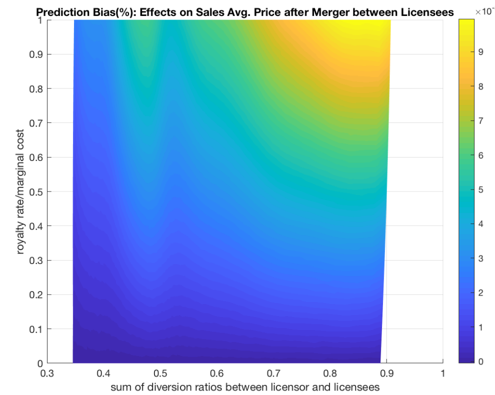
Theoretical Simulation Results. Figure 3.8 shows the theoretical simulation results on the prediction biases in the effect of merger between the two licensees, B and C. Firstly, panel 3.8a, panel 3.8b and panel 3.8c show that i) the prediction biases in the merger's effect on consumer welfare, sales-weighted average prices and total quantities increase with royalty rate; and ii) the prediction biases firstly increase then decrease with the sum of diversion ratios in two rounds. Secondly, panel 3.8d shows that the prediction biases in the merger effects producer surplus increase with both royalty rate and the sum of diversion ratios.

We interpret the above results with three implications similar to those from Figure 3.5. First, when a researcher predicts post-merger quantities and consumer surplus without data on existing patent licensing relationships, the researcher is better to be careful if royalty rates are large and the diversion ratios are at middle range. Second, when a researcher predicts post-merger prices and

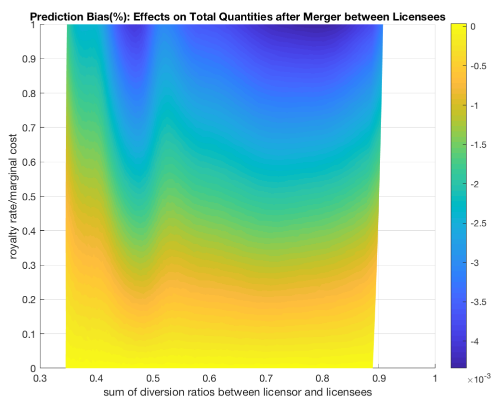
Figure 3.8: Theoretical Simulation: prediction bias in the effects on the merger between licensees.



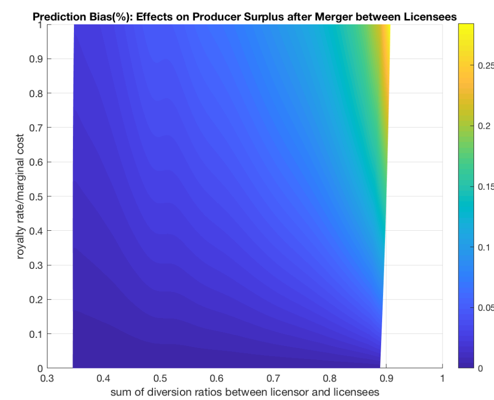
(a) Bias in Consumer Welfare Effect



(b) Bias in the Merger Effect on Sales-weighted Average Price



(c) Bias in the Merger Effect on Total Quantities



(d) Bias in the Merger Effect on Producer Surplus

profits without data on existing patent licensing relationships, the research is better to be careful if royalty rates and the sum of diversion ratios are large [21]. Third, estimation bias in marginal costs is not the deterministic factor for prediction biases: As alignment effect always increase with the sum of diversion ratios, we see up and downs in the prediction bias with respect to the sum of diversion ratios.

We summarize the prediction bias in Table 3.9, in the same format as Table 3.4. In particular, each row is a market outcome of interests. Column (1) reports the average prediction bias. Col-

[21] We note that non-monotonicity with respect to the sum of diversion ratios is more obvious in prediction bias on post-merger quantities and consumer welfare than in prediction bias on post-merger prices. We think the reason is that, apart from indirectly affecting the demand non-linearly through affecting prices with alignment effects, the travel parameter (determinant of diversion ratios) directly affect demand through consumer preferences.

umn (2) reports the average true merger effects. Column (3) reports the percentage of cases that generate over-predicted merger effects in magnitudes during the simulation. Column (4) reports the percentage of cases where the mis-specified model predicts opposite direction of merger effects during the simulation. The results are conditional on positive royalty rates, so that ignoring patent licensing relationship or not is non-trivial. Compared to the results in Table 3.4 for licensor-licensee merger, Table 3.9 shows that, for licensee-licensee merger, prediction bias on market-level outcomes have the same signs and smaller magnitudes; while prediction bias on product-level outcomes are quite different even in signs. We firstly discuss each group of outcomes, then focus on the opposite prediction cases.

Table 3.9: Theoretical Simulation: Predict the Effects of Licensee-Licensee Merger

	(1) avg. prediction bias (%)	(2) avg. true effect (%)	(3) over-prediction (%)	(4) opposite prediction (%)
Consumer Surplus	−0.203	−10.888	99.921	0
Sales-weighted Avg. Price	0.253	11.27	99.95	0
Price:A	0.446	2.221	97.911	4.446
Price:B	0.103	16.819	97.97	0
Price:C	0.093	16.722	97.089	0
Total Quantities	−0.152	−7.089	99.921	0
Quantity: A	−0.687	10.599	0.01	0
Quantity: B	0.079	−15.631	0.386	0
Quantity: C	0.099	−15.048	0.198	0
Producer Surplus	2.959	4.901	100	0
Profit:A	9.571	5.581	100	4.119
Profit:B+C	0.237	4.678	99.98	0

Notes: results are conditional on positive royalty rates. All simulated mergers are profitable: the smallest true merger effect on the joint profit of firms A and B is 1.28 percent.

With respect to licensee-licensee merger's effect on consumer surplus, the first row of Table 3.9 shows that, the average prediction bias is -0.2% with an average true merger effects of -10.9%. This implies that the mis-specified model typically over-predict the decrease in consumer surplus due to the licensee-licensee merger. This is also seen in the first row of Table 3.4. The last two columns in the first-row shows that, in 99.9% cases during simulation, the mis-specified model predicts larger consumer welfare effect than the true model, and in no case does the mis-specified model predict opposite consumer welfare effect compared to the true model.

With respect to licensee-licensee merger's effect on prices, the rows two to five of Table 3.9 show that, the average prediction biases and the average true merger effect on sales-weighted average price and product-level prices are all positive. This implies that the mis-specified model typically over-predicts the increase in prices due to the licensee-licensee merger, which also happens in licensor-licensee merger as shown in Table 3.4. The high ratios of over-prediction cases are consistent with this implication. Lastly, there are 4.4% cases where the mis-specified model predicts opposite licensee-licensee merger effect on price of the licensor's product, A. We discuss the underlying reasoning later.

With respect to licensee-licensee merger's effect on quantities, the rows six to nine of Table 3.9 show that, the average prediction bias and the average true merger effect on total quantities are negative. This implies that the mis-specified model typically over-predicts the decrease in quantities due to the licensee-licensee merger, which also happens in licensor-licensee merger as shown in Table 3.4. This implication is consistent with the 99.9% cases that see over-predictions. At product-level, we see different merger effects and prediction biases compared to Table 3.4. This is reasonable since the merged firms and different. In particular, with the merged firms being the licensees, the licensor's product, A, sees increase in quantities due to the merger. Then, given the average negative prediction bias on merger effect on quantity A (-0.69%), the result implies that the mis-specified model typically under-predicts the increase in the non-merging firm's quantity due to the licensee-licensee's merger. Notice that this is qualitatively different from the result in Table 3.4, where the averagely positive effect on the non-merging firm's quantity, C, is typically over-predicted. We will explain this result together with the opposite predictions on merger effect on price A later. As for merged firms' quantities, Table 3.9 shows that, the licensee-licensee merger averagely leads to decrease in merged firms' quantities (by 15.0% to 15.6%), and the mis-specified model averagely weakly under-predicts the decrease their quantities. This result is consistent with the close-to-zero ratio of over-prediction[22]. There is no opposite-prediction case with respect to quantity effects of the licensee-licensee merger.

With respect to licensee-licensee merger's effect on profits, the last three rows of Table 3.9 shows that, the mis-specified model averagely over-predicts the increase in firms' profits due to the licensee-licensee merger. This is consistent with the high ratio of over-prediction cases. Lastly, given the opposite prediction on price effects on product A, there also exists opposite prediction on profit effects on product A.

We summarize the opposite-prediction cases in Table 3.10, in the similar format as Table 3.5. In particular, Column (1) shows the average royalty rates where the opposite prediction on each

[22]For clarification, in Table 3.9, we note that the product-level over-prediction ratios for quantity effects are small while the market-level over-prediction ratio is large. This is because the over-prediction at market-level quantity effect is due to the under-prediction of quantity effect on product A: the increase in quantity of A is under-predicted, thus the decrease in market-level quantities is over-predicted.

market outcome happens. Column (2) shows the corresponding average sum of diversion ratios. Column (3) shows the corresponding average travel parameter. Column(4) shows the corresponding average true merger effects. Column (5) shows the corresponding average mis-specified merger effects. Column (6) shows the corresponding average prediction bias.

Table 3.10: Theoretical Simulation: Mis-specified Model Predicts Opposite Licensee-Licensee Merger Effects

	(1) royalty rates	(2) diversion ratios	(3) travel parameters	(4) true effects	(5) mis-specified effects	(6) prediction biases
	$\overline{r}$	$\overline{d}$	$\overline{\rho}$	$\overline{\%\Delta y}$	$\overline{\%\Delta\tilde{y}}$	$\overline{\Delta\Delta y}$
Price: A	0.801	0.348	0.085	−0.191	0.199	0.391
consumer surplus				−9.757	−9.967	−0.210
sales-weighted avg. price				7.500	7.691	0.190
total quantities				−6.018	−6.167	−0.149
producer surplus				1.046	2.958	1.912
Profit: A	0.953	0.370	0.177	−0.225	6.458	6.683
consumer surplus				−9.838	−10.138	−0.300
sales-weighted avg. price				7.955	8.238	0.283
total quantities				−6.278	−6.496	−0.219
producer surplus				0.909	3.345	2.435

Notes: results are conditional on opposite predictions.

Column(4) of Table 3.10 show that the opposite predictions of merger effects on price A happen with negative true merger effects on price A. Why would the true model predict lower price of A, the non-merging licensor, after licensee-licensee merger? This is due to decrease in alignment effect after the merger. In particular, after the licensee-licensee merger, the prices of the licensees' products increase, and their quantities decrease, which decreases how much the licensor cares about the licensees' sales. In fact, the alignment effects, in the case of opposite prediction of price of the licensor, are always smaller after the merger, and on average, decrease by 8.73%. While prices are strategic complements, the higher prices of B and C lead to smaller positive effect on the price A than the negative effect due to lower alignment effect. Such decrease in price of A after the licensee-licensee merger, also leads to decrease of profit of A after the merger, by about 0.23%.

To answer when the opposite predictions happen. Column (1) of Table 3.10 shows that such opposite predictions typically happen at relatively extreme cases: royalty rates account for 80% to 95% of total marginal costs on average. This is intuitive, because the larger the royalty rates, the more important is the alignment effect. Column (2) and (3) of Table 3.10 shows that such opposite predictions happen with relatively low values of diversion ratios and small travel parameter.

To answer how important are these opposite-prediction cases, we compare column (4) to (6) in Table 3.10 to column (1) and (2) in Table 3.9. Comparing column (1) of Table 3.9 to column (6) of Table 3.10, it shows that in the cases of opposite predictions, the prediction biases are not quite different in magnitudes, if not smaller. Comparing column (2) of Table 3.9 to column (4) of Table 3.10, it shows that in the cases of opposite predictions, the true merger effects are smaller in magnitudes. More importantly, we note that the average prediction bias in merger effects on price A conditional on opposite prediction is 0.39%, while the unconditional average prediction bias is 0.44% as reported in column (1) of Table 3.9. The smaller conditional average prediction bias implies that while the channel of ignored decrease in alignment effect after merger is driving the opposite predictions, there are other larger prediction biases happening when the true merger effects on price A are positive.[23] However, the conditional average prediction bias in merger effects on price A is larger than unconditional average prediction biases in merger effects on price B and C as reported in column (1) of Table 3.9. Therefore, we conclude that, ignored decrease in alignment effect due to licensee-licensee merger is a driving force of prediction biases.

As a summary of main findings from theoretical simulations on licensor-licensee merger and licensee-licensee merger, we find an alarming results to researchers: unobserved patent licensing relationships can lead to opposite prediction of merger effects. And the opposite predictions are fundamentally due to ignored mechanisms: i) royalty payment saving due to licensor-licensee merger; ii) smaller alignment effect due to licensee-licensee merger. The good news is that, in the theoretical simulations, which is based on a symmetric benchmark market, opposite predictions happen in relatively extreme cases. In particular, opposite predictions happen when royalty rates account for about 80% of total marginal costs.

Theoretical simulations also illustrate on the patterns between prediction bias and two model primitives: i) royalty rates; ii) the sum of diversion ratios between the licensor and licensees (controlled by the *travel parameter*). We find that larger royalty rates, larger bias in marginal cost estimation and merger evaluations. We also find that with larger sum of diversion ratios (i.e. substitutability), the bias in marginal cost estimation increases, but the prediction biases in merger effects typically firstly increase then decrease.

Lastly, theoretical simulations find that, on average, merger effects are over-predicted. In particular, for both licensor-licensee and licensee-licensee mergers, by ignoring patent-licensing re-

[23] Based on Equation 3.4 and the results in Table 3.9 and Table 3.10, we argue that estimation bias in marginal cost of the licensor-manufacturer, Δc_A, is also a driving force for overall prediction biases. In particular, given no opposite predictions on Q_A, the over-prediction ratio for merger effects on Q_A implies that, in almost all cases, $\tilde{s}'_A < s'_A$. Since firm A is a single-product firm, $v_A = -1/(\alpha(1-s_A))$. Therefore, $\tilde{v}'_A < v'_A$ in almost all cases. Then, both the second and the third terms in Equation 3.4 are negative, the first term – marginal cost estimation bias – is the only positive term. Moreover, Table 3.9 shows that on average, the prediction bias on merger effects on price A is 0.446%, which is positive. Therefore, on average, the estimation bias on marginal costs is driving the prediction bias on post-merger prices of the licensor-manufacturer, p'_A.

lationships, consumer surplus decrease is over-predicted(2.4% for licensor-licensee merger, 0.2% for licensee-licensee merger), increase of sales-weighted average price is over-predicted(2.65% for licensor-licensor merger, 0.25% for licensee-licensee merger), decrease in total quantities is over-predicted (1.76% for licensor-licensee merger, 0.15% for licensee-licensee merger), increase in producer-surplus is over-predicted (2.7% for licensor-licensee merger, 2.95% for licensee-licensee merger). Moreover, on average, the prediction bias in licensor-licensee merger is larger than that in licensee-licensee merger.

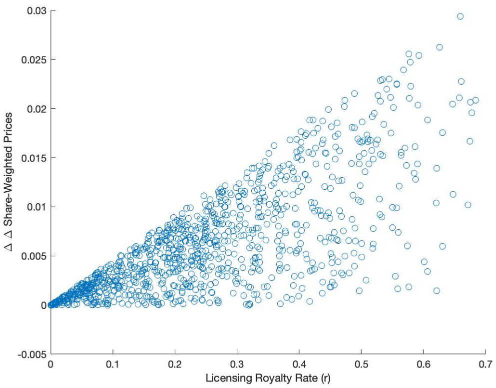
Guidance Simulation Results. Across the generated sample of simulated markets, we analyze a horizontal merger between the two licensees, i.e., Firm B and Firm C, and assess the predictions of counterfactual measures from the mis-specified model relative to those from the model accounting for the licensing in the market. We similarly evaluate how these predictions change with respect to underlying model parameters which comprise the alignment effect term, namely the royalty rate and the sum of diversion ratios.

In Figure 3.9, which plots simulated markets' biases in predicted welfare measures form a licensee-licensee merger against its royalty rate, we see similar trends to predictions in the guidance simulations of the licensor-licensee merger case. The mis-specified overpredicts increases in share-weighted prices (panel (a)), overpredicts decreases in inside shares (panel (b)), and overpredicts decreases in consumer surplus (panel (c)), relative to the true model, and the magnitudes of each of these overpredictions generally increases in the market's royalty rate, r. Additionally, for licensee-licensee mergers, a trend is present for the prediction bias is producer surplus with respect to the royalty rate – the mis-specified model overpredicts changes in producer surplus and the magnitude of this overprediction is correlated with r.

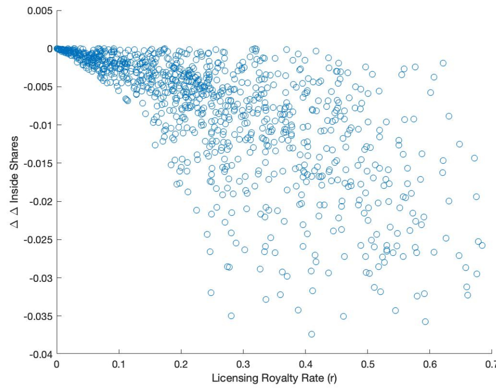
We also analyze how predictions differ based on the other component of the alignment effect – the sum of the diversion ratios. In Figure 3.7a, we see that the maximum overprediction in share-weighted average prices is first increasing in the sum of diversion ratios and then sharply drops at extreme values of the sum of diversion ratios. A similar pattern in overprediction of decreases in total inside shares is shown in Figure 3.7b. Correspondingly, the overprediction in decrease in consumer surplus is largest at interior, but relatively large values of the diversion ratios (Figure 3.7c). Finally, the maximum bias in producer surplus due to a licensee-licensee merger in increasing in the sum of the diversion ratios, as is shown in Figure 3.7d.

For licensee-licensee mergers, we want to understand when a mis-specified model predicts the opposite sign of a key welfare measure, relatively to the effect under a true model. In Table 3.12, we document, across the generated sample of markets, the average true effect, average prediction bias, and the frequency in which an opposite prediction occurs for key counterfactual objects – prices, shares, profits, and consumer surplus. We see, on average, due to the licensee-licensee merger, the true model predicts an increase in share-weighted average prices (by 4.8%) and a decrease in inside

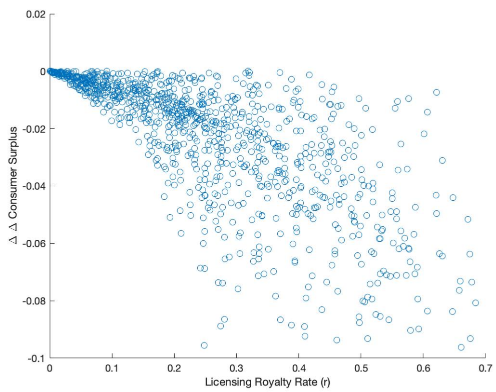
Figure 3.9: Guidance Simulations: prediction bias in the effects of the merger between licensees by licensing royalty rate, *r*.



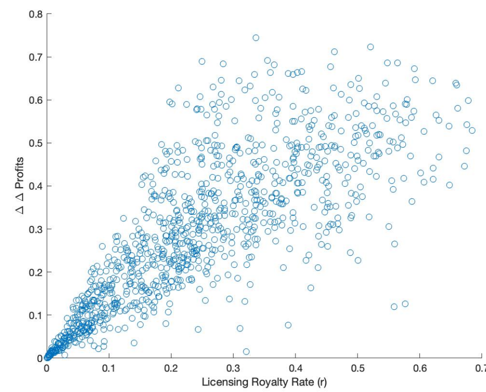
(a) Share-Weighted Prices



(b) Inside Shares



(c) Consumer Surplus

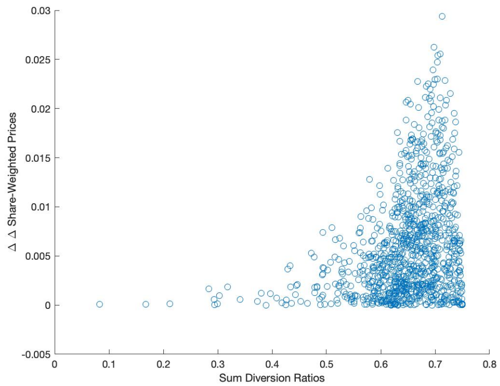


(d) Producer Surplus

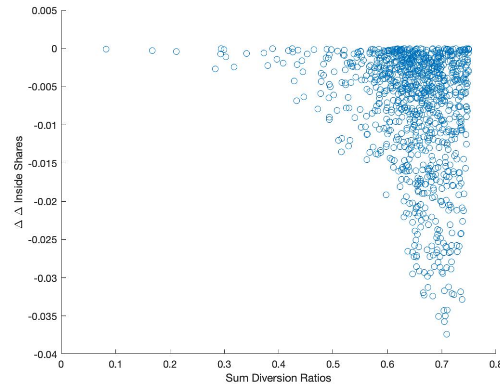
shares (by 5.4%), which is associated with an average decrease in consumer surplus (by 17.2%). In these three measures, prices, shares, and consumer surplus, the mis-specified model overpredicts the respective effects, however to a relatively small magnitude. For example, the mis-specified model overpredicts the percentage decrease in consumer surplus by 2.2 percentage points. Further, we see that in these three measures, the mis-specified model predicts the correct sign of the change in welfare measure in all markets (column (3)). However, in terms of producer surplus, the mis-specified model predicts an opposite effect in an overwhelming majority of markets in the sample (87.6%).

What are the characteristics of the markets in which the true and mis-specified model predict opposite effects? In Table 3.13, we detail the conditional distribution of market characteristics for markets that exhibit opposite predictions between the true and mis-specified model for at least

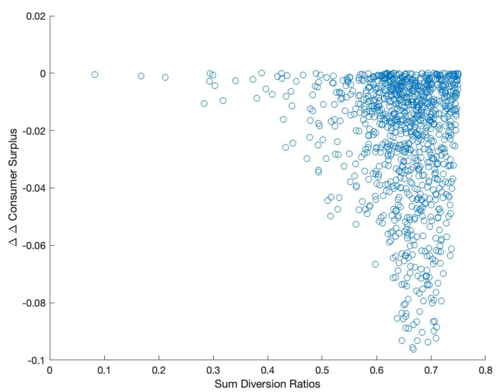
Figure 3.10: Guidance Simulations: prediction bias in the effects of the merger between licensees by licensing sum of diversion ratios between licensor and licensees.



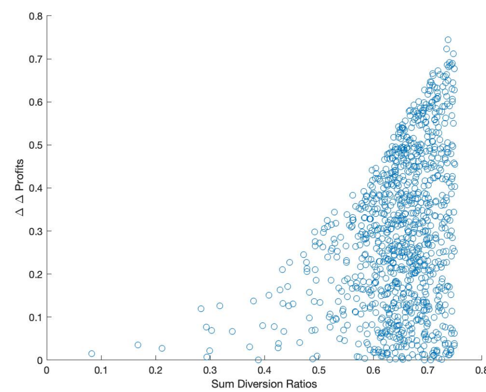
(a) Share-Weighted Prices



(b) Inside Shares



(c) Consumer Surplus



(d) Producer Surplus

one welfare measure. We see that relative to the distributions in the full sample detailed in Table 3.2, the markets with opposite predictions have larger royalty rates. However, in many other key measures, the conditional distributions of these market characteristics are relatively close to those of the full sample.

Summary on Guidance Simulations. In the guidance simulations, we assess differences in cost estimates and counterfactual merger predictions between a mis-specified model that does not take into account licensing behavior and a true model that does. In terms of costs, the bias in estimation of a licensor's cost from mis-specified model can be significant. Further, we show that this estimation bias increases in the licensor's royalty rate and exhibited larger potential magnitudes in less concentrated markets.

This difference in estimated costs plays through to model predictions of counterfactual scenar-

Table 3.11: Bias in Licensee-Licensee Merger Prediction

	5%	25%	50%	75%	95%
Prices:					
p_A	0	0.003	0.009	0.018	0.038
p_B	0	0.001	0.002	0.003	0.006
p_C	0	0.001	0.002	0.003	0.006
Share-Weighted Avg.	0	0.002	0.005	0.009	0.017
Shares:					
s_A	-0.237	-0.07	-0.026	-0.008	-0.001
s_B	0	0.001	0.004	0.01	0.022
s_C	0	0.001	0.004	0.01	0.022
Inside Shares	-0.026	-0.011	-0.005	-0.002	0
Profits:					
π_A	0.004	0.044	0.112	0.238	0.564
π_B	0.011	0.135	0.549	1.803	12.361
π_C	0.012	0.13	0.523	1.848	13.756
Producer Surplus	0.024	0.14	0.282	0.423	0.591
Consumer Surplus	-0.07	-0.034	-0.014	-0.005	-0.001

ios. Using this sample of generated markets, we evaluate two mergers – one between the licensor and a licensee, and one between the two licensees. Under the licensor-licensee merger, we show the mis-specified model over-predicts decreases in consumer surplus (from over-predicting increases share-weighted average prices and thus over-predicting decreases inside shares) and, on average, over-predicts increases in producer surplus. The magnitude of the over-predictions is shown to positive correlated in the licensor's royalty rate. Further, we show that the mis-specified model, under a licensor-licensee merger, can lead to policy recommendation opposite of that suggested by the true model. For example, the mis-specified model always predicts a loss of consumer surplus due to the merger, while in approximately one third of the simulated markets the true model predicts increases in consumer surplus.

Additionally, we use the generated sample to evaluate a licensee-licensee merger. Similarly, under this type of merger, the mis-specified model overpredicts changes in prices, shares, consumer and producer surplus, and these overpredictions increasing in underlying royalty rate. However, opposite implied policy recommendations under the mis-specified model are less frequent.

Table 3.12: Average Bias and Opposite Predictions in Licensee-Licensee Mergers

	(1) Avg. True Effect	(2) Avg. Prediction Bias	(3) Opposite Prediction
Prices:			
p_A	−0.003	0.012	0.512
p_B	0.098	0.002	0
p_C	0.099	0.002	0
Share-Weighted Avg.	0.048	0.006	0
Shares:			
s_A	0.2	−0.061	0
s_B	−0.203	0.007	0.006
s_C	−0.205	0.007	0.002
Inside Shares	−0.054	−0.008	0
Profits:			
π_A	0.002	0.171	0.513
π_B	−17.953	17.976	0.623
π_C	−4.885	4.907	0.624
Producer Surplus	−0.214	0.29	0.876
Consumer Surplus	−0.172	−0.022	0

Notes. Results here use all simulated markets. There are 92.5 percent of simulated markets where the licensee-licensee merger is not profitable. For results conditional on profitability of merger, please see Table F.14, which shows much smaller prediction bias and ratio of opposite predictions. For example, conditional on merger profitability, the average prediction biases on share-weighted average price, producer surplus, and consumer surplus are 0.1 percent, 3 percent, and negative 0.1 percent, respectively.

3.5 Conclusion

We use simulations to examine estimation bias in marginal costs and prediction bias in merger evaluations due to not accounting for existing patent licensing relationships between manufacturers. We examines two types of mergers: licensor-licensee merger, licensee-licensee merger. We use two sets of simulations for two complementary purposes: theoretical simulations for examining theoretical relationships between model primitives and the biases; guidance simulations for providing guidance for likely magnitude of the biases and likely correlations between observed market features and the biases. We find that estimation bias in marginal costs are increasing with royalty rates and the sum of diversion ratios between licensor and licensees. The guidance simulation shows that such bias is typically not ignorable: the median estimation bias is 28% of true

Table 3.13: Opposite Prediction Market Characteristics

Percentile	5%	25%	50%	75%	95%
Royalty Rate (r)	0.069	0.16	0.249	0.378	0.551
Shares:					
s_A	0.05	0.175	0.269	0.345	0.491
s_B	0.03	0.139	0.242	0.326	0.473
s_C	0.029	0.145	0.244	0.333	0.452
Total Inside Share	0.75	0.75	0.75	0.75	0.75
HHI:					
Pre-Merger	1895	2029	2269	2642	3345
Post Merger	2564	2641	2744	2926	3538
Δ from Merger	68	339	522	616	696
Margins (Pre-Merger):					
m_A	0.276	0.365	0.498	0.606	0.707
m_B	0.31	0.419	0.576	0.715	0.891
m_C	0.298	0.438	0.575	0.721	0.903
Share-Weighted Avg.	0.315	0.439	0.577	0.709	0.846
Costs:					
c_A	0.293	0.394	0.502	0.635	0.724
c_B	0.437	0.619	0.719	0.801	0.867
c_C	0.411	0.611	0.726	0.8	0.873
Share-Weighted Avg.	0.371	0.513	0.617	0.709	0.791
Market Elasticity	−7.382	−4.805	−3.63	−2.675	−1.923

Notes: Market elasticity under logit demand is given by $\alpha\bar{p}(1-s_0)$, where $\bar{p}$ represents the share-weighed average price

marginal cost values.

We show an alarming finding to researchers: not accounting for existing patent licensing relationships can lead to opposite prediction of merger effects. Such opposite predictions show up in both theoretical simulations and guidance simulations for both licensor-licensee mergers and licensee-licensee mergers. Because the guidance simulations cover a richer set of markets (including asymmetric market) compared to the theoretical simulations that are based on a symmetric benchmark market, opposite predictions show up more frequently in guidance simulations. In particular, in the theoretical simulations, we find 4.3% to 4.4% of cases where the merger effects on prices predicted from a mis-specified model not accounting for existing patent licensing relationships are opposite to the true merger effects predicted from a true model accounting for existing patent licensing relationships. In the guidance simulations, such fraction of opposite prediction cases can be as high as 51.2%. Moreover, both sets of simulations find that, opposite predictions happen when royalty rates are large. In particular, theoretical simulation finds that opposite predictions happen when royalty rates account for about 80% of total marginal costs; while guidance simulation finds that opposite predictions happen at high values of royalty rates in the random sample of markets. We also find the economic driving forces for opposite predictions in each type of mergers. In licensor-licensee mergers, we find that ignored saving of royalty payment due to licensor-licensee merger leads to decrease of prices and increase of consumer welfare after merger, which causes opposite predictions if one assumes away the patent licensing relationships. In licensee-licensee mergers, we find that ignored decrease in alignment effect due to licensee-licensee merger leads to decrease of licensor's price, which causes opposite predictions if one assumes away the patent licensing relationships. We argue that these two ignored channels when assuming away existing patent licensing relationships are also driving forces for overall prediction biases that are unconditional on opposite predictions.

In both sets of simulations and both types of mergers, we find that market-level merger effects are over-predicted when one assumes away existing patent licensing relationships. We further find that, the prediction biases are smaller in licensee-licensee merger than those in licensor-licensee merger. In particular, the guidance simulation finds that, in licensor-licensee mergers, assuming away existing patent licensing relationships lead to over-predicted increase of share-weighted average price by 2.6% at median, and over-predicted decrease of consumer welfare by 9.3% at median ; in licensee-licensee mergers, the two median predictions biases are 0.5% and 1.4% respectively.

We also examine the relationships between prediction biases, royalty rates and sum of diversion ratios between licensor and licensees. Theoretical simulation shows that prediction biases increase with royalty rates; while guidance simulation also shows positive correlations between the two. On the other hand, theoretical simulation shows non-monotonic relationship between prediction biases and the sum of diversion ratios; while guidance simulation finds that the potentials for prediction biases (i.e. maximum prediction biases) are positively correlated with the sum of diversion ratios. We leave examining the extent of prediction bias in real-life empirical context for future research.

APPENDIX A

Details on Data

A.1 Sample Selection: Criteria and Process

Here I describe the details in sample selection. A category is in the sample if it has benchmark conversion rates data from AppTweak. The sample selection processes for apps and keywords are listed below.

App Selection Process:
1. Find apps that have ever-ranked top50 in the top-grossing charts in the selected categories during Apr.2019 to Sep.2019.
2. Conditional on selection into step 1, find the 50 mostly downloaded apps in 2019 in each category.
3. Repeat the above steps for top-paid charts.
4. Drop an app-month observations if i) that app has zero download in that month; or ii) that app has unobserved file size or ratings in that month.

Keyword Selection Process:
1. Find keywords that have ever entered the list of recently used keywords of a selected app, based on AppTweak's keyword suggestions. The "recency" on AppTweak is last 3 month, which corresponds to March 2020 to June 2020. The suggested keywords are keywords that either i) the app has ranked in top100 in the keyword recently; ii) the app's title, or subtitle, or descriptions contains the keyword.
2. For each app-keyword, track the app's historical search ranking in the keyword on each day during 2019. For each category, find the 60 keywords that have the most apps showing up in top500 search results in the category.

A.2 Variable Construction

Update Frequency is weighted number of updates(i.e.,released versions) in a month. An update is weighted by the length of release notes. I calculate the three quartiles of release note length for each category. If an update of an app in a category has a release note shorter than the first quartile of release note length for the category, the weight on this update is 0.25. If an update of an app in a category has a release note longer than the first quartile but shorter than the second quartile of release note length for the category, the weight on this update is 0.50. If an update of an app in a category has a release note longer than the second quartile but shorter than the third quartile of release note length for the category, the weight on this update is 0.75. If an update of an app in a category has a release note larger than the third quartile of release note length for the category, the weight on this update is 1. Then, the weighted number of updates of an app in a month is the sum of the weighted updates of the app in the month. **Update level** is $\log(1+n_{jt})$, where n_{jt} is the update frequency of app j in month t.

Search Ranking of an app in a category in a month is average positions of the app in the search results of the popular keywords in the category in the month, conditional on the position is in top50. If an app has never reached top50 in any keyword on any day in a category in a month, that app does not have a search ranking in that category in that month. Such situation is tracked with an indicator.

Number of Star Ratings of an app/category/month combination is the average number of ratings the given star level across days in the month. Star levels are 1 to 5 in integers.

Average Rating is the weighted average of the app/category/month's number of star ratings, with the weight equating the level of the star.

Title (subtitle) Match is the weighted average number of keywords that contains any word in the title (subtitle) of the app in a given month, with the weight equating the search volume of the keywords.

Ratio of Adopted Keywords for an app in an category on a day is the number of popular keywords that are adopted by the app on that day over the number of keywords selected for that category. Monthly level measurement is average of that ratio across days. An app's adoption of a keyword on a day is approximated by the appearance of that app in the keyword's top 500 search results on that day.

APPENDIX B

Details on Estimation

B.1 Descriptive Evidence on Product Characteristics Known by Consumers Before Search

Here I provide descriptive test results on the assumption that consumers know all the observable app characteristics before search. I test the assumption with search volumes of keywords, an integer between 5 and 100 indicating the number of consumers searching for the keyword. Different keywords have different sets and orderings of apps in the search results. If consumers know app characteristics before search, they should search for keywords that have more high-value apps more and search for keywords that have less high-value apps less. In other words, the characteristics of apps in the search results of an keyword should affect the search volume of the keyword. The above theoretical implication motivates the following regression equation:

$$\begin{aligned} SearchVolume_{kgt} = &\beta_1^V AllPreinstall_{kgt} + AllPreinstall_{kgt} \times \{\bar{x}_{kgt}^V \cdot \beta_2^V + \beta_3^V \bar{p}_{kgt}\} \\ &+ \beta_4^V \overline{Apple}_{kgt} + \beta_5^V \overline{Preinstall}_{kgt} + \beta_5^V brand_k + \lambda_{gt}^V + \varepsilon_{kgt}^V \end{aligned} \tag{B.1}$$

where $AppPreinstall_{kgt}$ indicates whether all the top50 search results in keyword k month t contain no apps in category g other than pre-installed apps. Within each category/month pair gt, in the case of there are non-pre-installed apps showing up in the top50 search results for a keyword k, I calculate the average prices across these apps, denoted by $\bar{p}_{kgt}$. Similarly, $\bar{x}_{kgt}$ denotes the vector of average levels of app characteristics, including update levels, average rating, age, file size, number of screenshots, description length, offer in-app-purchase or not, and paid installation or not. To capture the idea that higher-ranked products are more considered by consumers, these characteristics are weighted by $1/log(1+ranking_{jkt})$, where $ranking_{jkt}$ is the average ranking of app j in keyword k in month t. $\overline{Apple}_{kgt}$ denotes the ratio of observed positions that are taken by Apple's apps in search results of keyword k. Similarly, $\overline{Preinstall}_{kgt}$ denotes the ratio of observed positions that are taken by pre-installed apps in search results of keyword k. $brand_k$ indicates whether the keyword is a brand-name keyword.λ_{gt} denotes category-month fixed effects, capturing unobservables that are keyword-invariant and change across markets. In the robustness check, I use keyword-category fixed effects and month-fixed effects, omitting the brand-name keyword indicator. The keyword-category fixed effects capture unobservables that are time-invariant and change across keyword/category pairs. The month-fixed effects capture unobservables that are keyword/category-invariant and change over time.

Table F.4 reports the summary statistics of the data for estimating the above equation. The left panel reports the data used for the main specification: search volume and average app characteristics across observed apps in top50 search results. The right panel reports the data used for robustness check: search volume and app characteristic of the observed top1 app. It shows that, for an average keyword/category/month pair, there are 2% of observed top50 positions taken by Apple's apps. It also shows that, about 5% of observed top1 positions are taken by Apple's apps. It also presents that, compared to an average observed top50 app, an average top1 app has more experiences, larger file size, more screenshots, longer descriptions and higher installation prices. In the left panel, it shows that, among keyword/category/month combinations that have observed top50 search results, the average search volume is 48.29 with a standard deviation of 13.88, and 34% of them are generated with brand-name keywords.

Table F.5 presents the estimation results of Equation B.1. The first column reports the main specification results. It shows that consumers are significantly more likely to search for keywords whose top50 search results contain apps that on average update more, have more experience, smaller in file size, have more screenshots and shorter description, offer in-app-purchase, have lower or even zero installation price. It indicates that consumers at least know these app characteristics to some extent such that they can predict what apps they will see in the search results and thus choose what keywords to search for. Although average rating does not significantly affect search volume in the main specification, its coefficient is insignificantly positive across all specifications, and becomes significantly positive when only looking at top1 search results and controlling for keyword-category fixed effects and month-fixed effects. Overall, the descriptive evidence indicates the that assumption that consumers know the observable app characteristics before search is not too crazy.

B.2 Details in Estimation of Search Ranking Model

Here I list the control variables in the vector z^s_{jm} in Equation 1.12.

- $1\{j \textit{ is pre-installed}\} * 1\{t(m) = \tau\}$
- price($p_{jt(m)}$)
- paid($z^s_{1jt(m)}$)

- title-match($z^s_{2jt(m)}$)
- subtitle-match($z^s_{3jt(m)}$)
- last-period number of 5-star ratings($z^s_{4jt(m)}$)
- last-period number of 3-star ratings($z^s_{5jt(m)}$)
- last-period number of 1-star ratings($z^s_{6jt(m)}$)
- ever show up in top 500 search results in any selected keyword for the category ($z^s_{7jt(m)}$).
- brand ratio: the ratio of brand keywords among the keywords where the app has an observed ranking, interacted with z_{7ht}. ($z^s_{8jt(m)}$)
- $z_{jt} * 1\{g = h\}$ for each category $h \neq$ *Business* and $z_{jt} \in \{p_{jt(m)}, z^s_{1jt(m)}, z^s_{2jt(m)}, z^s_{3jt(m)}, z^s_{4jt(m)}, z^s_{5jt(m)}, z^s_{6jt(m)}, z^s_{7jt(m)}, z^s_{8jt(m)}\}$. Here, to save computation time, all game apps are indexed into one category: Game.
- $z_{jt} * 1\{t = \tau\}$ for each $t = 1, 2, \cdots, 23$ and $z_{jt} \in \{p_{jt(m)}, z^s_{1jt(m)}, z^s_{2jt(m)}, z^s_{3jt(m)}, z^s_{4jt(m)}, z^s_{5jt(m)}, z^s_{6jt(m)}, z^s_{7jt(m)}, z^s_{8jt(m)}\}$.

B.3 Truncated Set of Possible Orders

For markets with strictly more than 5 products, I use truncated set of possible orders $\mathscr{B}_a$ to construct firms' beliefs on search rankings given update levels a. Now I explain how $\mathscr{B}_a$ is constructed. First recall that in Equation 1.15, the probability for an order y, denoted by $\mathbb{P}[y]$, is given by

$$\mathbb{P}[y] = p_{y(1)} \cdot \frac{p_{y(2)}}{1 - p_{y(1)}} \cdot \frac{p_{y(3)}}{1 - p_{y(1)} - p_{y(2)}} \cdot \cdots \cdot \frac{p_{y(J-1)}}{p_{y(J-1)} + p_{y(J)}} \cdot \frac{p_{y(J)}}{p_{y(J)}}$$

$\mathscr{B}_a$ are constructed based on the most-likely order, y^*, predicted from the estimated rank-ordered logistic model, given a vector of updates a. Notice that the numerator of $\mathbb{P}[y]$ is unchanged with y. Inspired by that observation, $\mathscr{B}_a$ includes the following types of permutations of y^*, found by enlarging the denominator of $\mathbb{P}[y]$[1]. For all the examples below, I use 12345 to denote the ordering of 5 products in the most-likely ordering in a market with at least 5 products.[2]

1. first-layer enlargement-1: alter the positions of two nearby products. For example, 12345 $\rightarrow$ 21345. $\forall j \in [1, J^{[1]}_m - 1]$.
2. first-layer enlargement-2: alter the positions of two products that only have one other product located between them. For example, 12345 $\rightarrow$ 32145. $\forall j \in [1, J^{[1]}_m - 2]$.

[1] Another type of permutation that also marginally enlarges the denominator is to alter the positions of two products with closest mean ranking scores. However, this will cause the truncation set to be sensitive to marginal changes in update levels. Therefore, it's not considered here.

[2] In rank-ordered logistic regression model, the most-likely ordering is ordering of products by the mean score, which maximizes the $\mathbb{P}[y]$.

3. second-layer enlargement-1: alter the positions of two nearby products; then for the positioned higher product among the two after the shift, alter its position with the product that's right above it. For example, 12345 → 13245 = 13245 → 31245. $\forall j \in [2, J_m^{[1]} - 1]$.
4. second-layer enlargement-2: alter the positions of two nearby products; then for the positioned lower product among the two after the shift, alter its position with the product that's right below than it. For example, 12345 → 21345 = 21345 → 23145. $\forall j \in [1, J_m^{[1]} - 2]$.
5. second-layer enlargement-3: alter the positions of two nearby products; then alter the positions of two nearby products that are right below them. For example, 12345 → 21345 → 21435. $\forall j \in [1, J_m^{[1]} - 3]$.

Conduct the above permutations until position 30, whenever it applies. The resulting size of $\mathscr{B}_a$ is

$$5 \times \min\{J, 30\} - 9 \leq 141$$

B.4 First-Stage Results of Instruments in the Demand Model and Supply Model

Table F.6 presents the F-statistics of first-stage IV regressions on the demand side (column 1-5) and supply side (column 6). All F-statistics for excluded instruments are larger than 40. Table F.7 reports the coefficients and standard errors in the first-stage estimation.

B.5 Constraints on Supply Model Estimation

The following constraints are applied on estimation of Equation 2.1 to guarantee conditional profit maximization in Nash Equilibrium.

1. first-order increasingness: $\tau_1 > 0$, $\tau_1 + \tau_2 > 0$
2. concavity: $\tau_3 < 0$, $\tau_3 + \tau_4 < 0$

The following constraints are applied on estimation of Equation 2.7 to guarantee conditional profit maximization in Nash Equilibrium.

1. Necessary Second-order Condition (negative Hessian Diagonal) for update level of app j, $a_j > 0$:
$$\frac{\partial^2 \pi_f^{II}(a_j, a_{-j}, \omega_j; \phi, \psi)}{\partial a_j^2} < 0$$
2. Non-negative marginal update benefits: $g'(a_{jgt}, \omega_{jgt}; \phi) \geq 0$.
3. non-negative marginal in-app-advertising profit wrt downloads: $F'(Q_{jgt}; \psi) \geq 0$.

4. Constraints on signs of parameters:
    (a) higher update, higher costs: $\phi_1 > 0$.
    (b) increasing and concave in-app-advertising profit wrt downloads: $\psi_1 > 0$, $\psi_2 < 0$

B.6 Algorithm to Find Equilibrium Positive Update Levels

Given a draw of update cost slope shifters, ω, and an update portfolio D, find equilibrium update levels (including zeros), $a^*(\omega;D)$ in the following steps:

1. Starting update levels: $a_0 = (0, a_{-j})$ if $D_j = 0$; $a_0 = (a_j, a_{-j})$ if $D_j = 1$.
    - If $D_j = 1$ in the data, then a_j takes the value in the data.
    - Otherwise, $a_j = \log(1.25)$.
2. Use the truncation method to construct the set of likely orders $\mathscr{B}_0$ following these steps:
    (a) predict probabilities to be ranked first, p_0 based on the rank-ordered logit model, given a_0 .
    (b) use the probabilities, p_0, to find the set of likely rankings, $\mathscr{B}_0 := \mathscr{B}_{a_0}$.
3. Solve for the equilibrium update levels, a^*, based on the ranking set, $\mathscr{B}_{a_0}$.
4. Check if $\mathscr{B}_{a_0} = \mathscr{B}_{a^*}$ or $||a^* - a_0||_\infty < 0.001$.
    - If true, equilibrium found.
    - If false, set $\mathscr{B}_{a_0} = \mathscr{B}_{a^*}$, $a_0 = a^*$, repeat steps 3 and 4.

APPENDIX C

Details on Simulation

C.1 Compare with Difference-in-Differences Estimates

Here I compare the structural and difference-in-differences(DiD) estimates of the average treatment effect(ATE) of the search algorithm change in July 2019. To compute the structural estimates of the ATE, I simulate the post-change market outcomes if there were not the algorithm change. Because only categories with Apple's apps, i.e., the treatment group, will have different market outcomes, the simulation is conducted in categories with Apple's apps during the post-change periods from August to November in 2019. Because only bounds on fixed-costs of updates are identified, I follow the specification in **??** to draw fixed costs and report average effects across the draws. Moreover, because there is unobserved randomness in search rankings from the unobserved ranking score shocks, which feeds into randomness of installations, when comparing no-algorithm-change search rankings(installations) to with-algorithm-change search rankings(installations), I compare the expected values of search rankings(installations). Specifically, I simulate no-algorithm-change *expected* search rankings(installations) based on simulated no-algorithm-change update levels, and with-algorithm-change *expected* search rankings(installations) based on observed post-change update levels, where the expectation is over the truncated sets of possible search rankings. The structural estimates of average treatment effect on search rankings(installations) is difference between expected search rankings(installations) without the search algorithm change and expected search rankings(installations) with the search algorithm change.

Table F.8 reports the simulation results. It shows that the structural estimate of the average treatment effect on update levels is 1.7% while the difference-in-differences estimate is 2.1%. They are reasonably close. The smaller structural estimate also makes sense, because only the top5 firms are allowed to change update levels in the simulations, which potentially restricts the magnitudes of the effect. The structural estimate of the average treatment effect on search rankings is 1.5% while the difference-in-differences estimate is 3.6%. The discrepancy can be explained by the imperfect fitness of the search ranking model and the fact that the expectation is over the truncated set of possible search rankings instead of the complete set. The structural estimate of the average treatment effect on installation is 3.4% while the difference-in-differences estimate is 22.1%. They have the same signs but different magnitudes. The reasons for the discrepancy are two-fold: i) the discrepancy in effects on updates and search rankings feeds into the discrepancy in effects on downloads; ii) in the structural estimation, apart from update levels, all the other product characteristics and market features are fixed; while in the difference-in-differences, when comparing pre-change outcomes to post-change outcomes across treatment and control groups, there might be changes in other product characteristics (for example, file size and average ratings) and market features (for example, number of products). These changes lead to additional changes in installations that are not captured by the structural estimates. Overall, the structural estimates of ATE have the same signs as the difference-in-differences estimates, and the discrepancy have reasonable explanations.

C.2 Fix Pre-installed Apps' Search Rankings

Here I explain how I fix the search rankings of pre-installed apps and present its effects on market equilibrium. Following the idea that, search rankings are not perfectly known by developers in the stage of update choices, I fix the beliefs on search rankings of pre-installed apps (instead of observed search rankings of pre-installed apps). Specifically, following the estimated search ranking model and the construction of truncated set of possible search rankings as developers' beliefs on search rankings, I calculate the marginal distribution of search rankings of pre-installed apps in the status-quo. Then, I fix this marginal distribution through all simulations, with or without changes in preferential treatment on platform-owned products. Therefore, the welfare and product competition effects presented in Section 1.6 are immune from the effects of fixing pre-installed apps' search rankings. To complete the picture, I explain and present the effects of fixing the marginal distribution of pre-installed apps' search rankings on market outcomes below.

Depending on the extent of update adjustment that's allowed in the counterfactual simulation, the market outcomes with fixed marginal distributions of the search rankings of pre-installed apps may be different. Specifically, I calculate the market outcomes in three cases: i) no-game simulation: app developers are not allowed to change update levels; ii) partial-game simulation: app developers are only allowed to change positive update levels; iii) full-game simulation: only top5 app developers in each category are allowed to change update levels, including update or not and positive update levels. It is relatively easy to understand why the latter two cases see different market outcomes when fixing the marginal distribution of pre-installed apps' search rankings: equilibrium update levels could be different due to different beliefs of developers. When pre-installed apps' search rankings are not fixed, developers may believe that they can replace pre-installed apps in the positions that are occupied by pre-installed apps. When pre-installed apps' search rankings are fixed, they won't believe so.

In the first case (no-game simulation), the market outcome changes with fixing pre-installed apps' search rankings because of truncation of the set of possible search rankings. To illustrate the idea, let's consider an example where there are three products, 1-2-3, with the first product as the pre-installed app. As a benchmark, let's firstly consider the case of no truncation of the set of possible search rankings. The goal is to compare the probability to observe product-2 ranked above product-3 with and without fixing pre-installed apps' search rankings. The complete set of possible rankings of the three products is $\{123, 132, 213, 231, 312, 321\}$. Then, based on Equation 1.14 and 1.15, the probability to observe product-2 ranked above product-3 is given by $\mathbb{P}[23] = \mathbb{P}[123] + \mathbb{P}[213] + \mathbb{P}[231] = \exp(\overline{score}_2)/[\exp(\overline{score}_2) + \exp(\overline{score}_3)]$. After fixing the rankings of the pre-installed product-1, there are only two products that will be ranked in the search ranking model: product-2 and product-3. Therefore, the probability to observe product-2 ranked above product-3 after fixing the pre-installed product's ranking is also $\exp(\overline{score}_2)/[\exp(\overline{score}_2) + \exp(\overline{score}_3)]$. Therefore, without truncation of the set of possible search rankings, market outcome may not change with fixing pre-installed apps' search rankings.

Now, let's consider the case of truncation of the set of possible search rankings. As described in Appendix B.3, one type of permutation included in the truncated set is alternating the positions of two near-by products in the most-likely ordering of products. For simplicity, we only consider this type of permutation here.[1] Then, let us suppose the most-likely ordering of products is $\{123\}$, which gives the following truncation set of orderings: $\{123, 132, 213\}$. Notice that the order $\{231\}$ is not captured by the truncation set. Then, the probability of observing product-2 ranked above product-3 within the truncation set is $\tilde{\mathbb{P}}[23] = \exp(\overline{score}_2)/[\exp(\overline{score}_2) + \exp(\overline{score}_3)(\exp(\overline{score}_1) + \exp(\overline{score}_3))]$. When fixing the pre-installed product-1's position, the truncation set becomes $\{23, 32\}$, which is also the complete set for the two products. Then, the probability of observing product-2 ranked about product-3 within the truncation set is $\exp(\overline{score}_2)/[\exp(\overline{score}_2) + \exp(\overline{score}_3)]$, which is different from the probability without fixing pre-installed apps' search rankings. Therefore, with truncation of the set of possible search rankings, market outcomes will change with fixing pre-installed apps' search rankings.

To show the effects of fixing pre-installed apps' search rankings on market outcomes, I compare the simulated market outcomes with fixed marginal distribution of pre-installed apps' search rankings(denoted by $y_{jgt}^{[fix]}$) to market outcomes without fixed marginal distribution of pre-installed apps' search rankings(denoted by $y_{jgt}^{[0]}$). The difference captures the effect of fixing marginal dis-

[1] Considering all five types of permutation listed in Appendix B.3 will cause the truncation set equal to the complete set for just three products.

tribution of pre-installed apps' search rankings. To measure the difference, I use relative L1-Norm in percentage : $100 \times [\sum_{jgt}(|y_{jgt}^{[fix]} - y_{jgt}^{[0]}|)] / [\sum_{jgt} |y_{jgt}^{[0]}|]\%$.

Table F.9 reports the effect of fixing marginal distribution of pre-installed apps' search rankings. All of the changes are smaller than 0.1%. Although the changes are small, they are taken into account during counterfactual simulations, i.e., all reported effects in Section 1.6 are comparing the market outcomes without preferential treatment effects on platform-owned products in search algorithm and market outcomes with preferential treatment effects on platform-owned products in search algorithm, while the marginal distribution of pre-installed apps' search rankings are fixed in both cases.

C.3 Calculation of Consumer Surplus

This appendix gives details on computing the expected consumer welfare in Equation 1.16. To compute the consumer welfare, I take 10,000 draws of the idiosyncratic unobserved match values and consumer search costs. The match values are drawn from the Type-I Extreme Value distribution. The search costs are drawn from the app/category/month-specific distribution given in Equation 1.6.[2] On top of these random shocks, there are i) 4 sparse-grid nodes for the one-dimension random coefficient on update levels generated from the sparse-integration method in Heiss and Winschel (2008) with accuracy level of 4; and ii) the truncated set of possible search rankings. Given each draw of the random coefficients, each vector of possible search rankings in the truncated set and each draw of match values and search costs, I simulate the optimal sequential search problem in each market(a category/month pair) in the following steps based on the three rules in Weitzman (1979):

1. Implement the *searching rule*. Sort all apps in the markets by reservation values in descending order. This is the order of search by the consumer. The consumer-app specific reservation values, r_{ij}, are computed based on Lemma 1 in Moraga-González, Sándor and Wildenbeest (2018), i.e., $r_{ij} = \delta_{ij} + H_0^{-1}(c_{ij})$, where δ_{ij} is the known utility before search, equating $u_{ij\cdot} - \varepsilon_{ij\cdot}$ in Equation 1.3; c_{ij} is the consumer-app specific search costs; and the function $H^0(\cdot)$ is given in Equation 1.6.

2. Check the *stopping rule*. Along the order of search, compare the highest realized utility to the reservation value of the next app to be searched. The consumer stops search when the highest realized utility is higher than the reservation value of the next app to be searched.

[2] A useful detail in computing the consumer-product specific search costs is that I save the search costs across consumers for each evaluated pair of product-position. This significantly saves the computational time.

The *search costs* incurred by the consumer is the sum of search costs for all the apps that have been searched.

3. Check the *purchasing rule*. The consumer downloads the app with the highest realized utility among the apps that have been searched. This is also the realized utility of this consumer. The welfare of this consumer is (realized utility - search costs) in the unit of *util*. Divide the consumer welfare by the estimated price coefficient in Table 1.5 gives the consumer welfare in dollars.

Then I take average of simulated consumer welfare and multiply it with market size to get market-level consumer surplus. Notice that, by simulating the optimal sequential search problem, I have computational market shares derived from the discrete choices of simulated consumers. Therefore, there might be distance between the computational market shares and analytical market shares derived from the closed-form choice probability. This can be used to measure the accuracy of the consumer surplus measurement. Table F.12 presents the computational error during computation of consumer surplus. It shows that all computational errors are smaller than 0.4%.

C.4 Explain Heterogeneous Effects on Update Frequency

The understand why some independent apps increase their update frequency while others decrease, I summarize the patterns between the percentage change of update frequency and some explanatory variables with the following linear regression. It is only a correlation results, instead of causal relationship. But it is helpful for summary.

$$\begin{array}{lllll} \%\Delta update_{jm} = & -0.65\; + & 0.66\;\%\Delta Q_{jm} - & 0.27\; nop_{jm} - & 0.17\; June_m + \rho_{jm} \\ & (0.96) & (0.36) & (0.14) & (1.19) \end{array}$$

where $\%\Delta update_{jm}$ is the percentage change of update frequency of app j in market m after eliminating the identified self-preferencing, $\%\Delta Q_{jm}$ is the percentage change of downloads of app j in market m after the eliminating without update adjustment, nop_{jm} is the number of apps owned by app j's developer in market m, and $June_m$ indicates whether the market m is in June 2019 (the month with stronger identified self-preferencing).

The regression results indicate that reduction of update frequency after the elimination is related to reduction of downloads without update adjustment, stronger cannibalization concern (captured by larger nop_{jm}), and aggressive reduction in self-preferencing (captured by $June_m$). In particular, looking into the simulation data, I find that apps whose downloads decrease after the elimination without update adjustment are typically apps who are always above or below Apple's apps with or without self-preferencing. Then, their reduction in downloads are due to business stealing from boosted-up independent apps who replace Apple's apps. It implies that the boosted-up independent apps are stronger competitors and more attractive for consumers than Apple's apps.

C.5 Counterfactual Simulation Results without Update Portfolio Adjustment

Here I report the simulation results with changes in positive update level but without update portfolio adjustments. I call such simulation as partial-game simulation. Because all independent apps with positive update levels are allowed to make a change, there more simulation data points than those in the case of full update adjustment by top5 developers.[3] It motivates me to use these results to discuss patterns with respect to heterogeneous effects of self-preferencing on update levels. I firstly show the quantified effects of self-preferencing on update levels, search rankings, installations and welfare. Then I discuss the heterogeneity in the effects on update levels.

Table F.10 presents the effect of self-preferencing on positive update levels from the partial-game simulation. The effects have the same direction as the main results reported in Table 2.2, with smaller magnitudes. While we don't see much heterogeneous effects at market-level, the heterogeneity at product-level remains strong. The standard deviation of the percentage change of product-level positive update levels is 4.2 times of the mean. The range of the percentage change of product-level positive update levels is from -0.2% to 4.4%.

Table F.11 presents the effects of self-preferencing on search rankings, installations and welfare in the partial-game simulation. The effects are quite similar to the main results reported in Table 2.3. The smaller increase in producer surplus compared to that in the main result can be explained by restricted adjustment: the developers are not allowed to adjust update portfolios in the partial game.

C.6 Ranking Imperfection due to Self-preferencing

Here I explain how the ranking imperfection due to self-preferencing is calculated. Ideally, the rankings should be based on average scores equal to consumer values on the apps (relative mean-utility). However, this is not true neither with or without the self-preferencing for Apple's apps. Therefore, I firstly calculate i) within-market rankings that are based on app values, ii) the within-market rankings that are based on the mean scores with self-preferencing, and iii) the within-market rankings that are based on the mean scores without self-preferencing. Then I calculate

[3] There are 337 app/category/month combinations whose update levels are subject to changes in counterfactual simulations with endogenous positive update levels. That number is 79 app/category/month combinations in counterfactual simulations with endogenous update levels of top5 developers in each category.

the gap between the first and the second within-market rankings, and the gap between the first and the third within-market rankings. The gaps are the ranking imperfections across products. To aggregate the product-level ranking imperfections to market-level, I firstly construct a weight. Because higher positions are more important for welfare, I use $1/\log(1+\mathit{ranking})$ as the weight on the gaps across products. Then, I use the weighted average gaps in a market to measure the ranking imperfection in the market. Lastly, I calculate the percentage change in the market-level ranking imperfection after eliminating self-preferencing to measure the ranking imperfection due to self-preferencing for Apple's apps.

APPENDIX D

Ambiguous Supply-side Effect of Self-preferencing

Here I explain the ambiguity of supply-side responses to change in self-preferencing: it is theoretically ambiguous that whether developers will update more or less with less self-preferencing of Apple's apps. The ambiguity roots in two offsetting economic forces: i) strategic complementarity of the virtual mean relative utility (which roots in concavity of revenue curves); ii) larger marginal quantity with respect to update with higher search rankings. I firstly illustrate the intuitions in a two-product market, then discuss the case in more complex markets.

Imagine a market with only two apps: i) app-1, an Apple's app; ii) app-2, an independent app. Then, there are two possible orders of the apps: 12 or 21. Then, the marginal benefit from update for app-2 is given by

$$MB_2 = \sum_{y\in\{12,21\}} \left(\frac{\partial \pi_2^I(y)}{\partial a_2} \tilde{\mathbb{P}}[y] + \pi_2^I(y) \frac{\partial \tilde{\mathbb{P}}[y]}{\partial a_2} \right)$$

A decrease in self-preferencing of Apple's apps is a decrease in θ_{Apple}, which only changes the function $\tilde{\mathbb{P}}[y]$, but not the profit function output π_2^I. In particular, denote the new probability function with lower θ_{Apple} as $\tilde{\mathbb{P}}'[y]$. Then, the post-change marginal benefit from updates for app-2 is given by

$$MB_2' = \sum_{y\in\{12,21\}} \left(\frac{\partial \pi_2^I(y)}{\partial a_2} \tilde{\mathbb{P}}'[y] + \pi_2^I(y) \frac{\partial \tilde{\mathbb{P}}'[y]}{\partial a_2} \right)$$

Denote the difference between the post-change values and pre-change values with $\Delta x := x' - x$. Notice that $\tilde{\mathbb{P}}[12] + \tilde{\mathbb{P}}[21] \equiv 1$. Therefore, $\Delta\tilde{\mathbb{P}}[12] + \Delta\tilde{\mathbb{P}}[21] = 0$, and $\frac{\partial \tilde{\mathbb{P}}[12]}{\partial a_2} = -\frac{\partial \tilde{\mathbb{P}}[21]}{\partial a_2}$. It is also useful to notice that i) a decrease in θ_{Apple} will cause $\Delta\tilde{\mathbb{P}}[21] > 0$; ii) $\pi_2^I(21) > \pi_2^I(12)$ since only rankings are different, and it's identified and estimated that $\lambda_2 > 0$ and thus $Q_2(21) > Q_2(12)$, and marginal profits from quantities are estimated as positive[1]. I use these observations to simplify

[1]Marginal "revenue" from quantities is given by $0.7p_j + 0.7(\tau_1 + 2\tau_2 Q_j) + (\phi_1 + 2\phi_2 Q_j)$ for each product j.It's estimated that $\tau_1 > 0$, $\tau_2 < 0$, $\phi_1 > 0$, $\phi_2 < 0$. Therefore, the "revenue" curve is increasing and concave with respect to quantities.

ΔMB_2 and get the following expression:

$$\Delta MB_2 = \underbrace{\Delta\tilde{\mathbb{P}}[21]}_{>0}\underbrace{\left(\frac{\partial \pi_2^I(21)}{\partial a_2} - \frac{\partial \pi_2^I(12)}{\partial a_2}\right)}_{I\ (unclear\ sign)} + \underbrace{\Delta\frac{\partial \tilde{\mathbb{P}}[21]}{\partial a_2}}_{II\ (unclear\ sign)}\underbrace{\left(\pi_2^I(21) - \pi_2^I(12)\right)}_{>0} \tag{D.1}$$

I analyze term I first, then term II. The term I is given by:

$$\begin{aligned}\frac{\partial \pi_2^I(21)}{\partial a_2} - \frac{\partial \pi_2^I(12)}{\partial a_2} = & \underbrace{[0.7\underbrace{(\tau_1 + 2\tau_2 Q_2(21))}_{R_2^{(q)}} + \underbrace{\phi_1 + 2\phi_2 Q_2(21)}_{F_2^{(q)}}]}_{small}\underbrace{\frac{\partial Q_2(21)}{\partial a_2}}_{probably\ large} \\ & - \underbrace{[0.7\underbrace{(\tau_1 + 2\tau_2 Q_2(12))}_{R_2^{(q)'}} + \underbrace{\phi_1 + 2\phi_2 Q_2(12)}_{F_2^{(q)'}}]}_{large}\underbrace{\frac{\partial Q_2(12)}{\partial a_2}}_{probably\ small}\end{aligned} \tag{D.2}$$

Because i)$Q_2(21) > Q_2(12)$, ii) it's estimated that $\tau_2 < 0$ and $\phi_2 < 0$, I have lower marginal "revenues" wrt quantities from the order 21 than those from the order 12. This is just saying that when ranked higher, the app-2's downloads increase. Given the concave revenue curve, this decreases the marginal benefit from updates. An alternative explanation is strategic complementarity of the virtual mean relative utility (product value net of search costs): app-2's competitor, app-1, is ranked lower, and thus has a lower virtual mean relative utility; which decreases app-2's marginal benefits from increasing the virtual mean relative utility because their virtual mean relative utilities are strategic complements. The strategic complementarity roots in the increasing and concave revenue curve wrt quantities.

Now I compare $\frac{\partial Q_2(21)}{\partial a_2}$ and $\frac{\partial Q_2(12)}{\partial a_2}$, and argue that the former is likely to be larger in this paper's empirical context. Their difference is given by:

$$\frac{\partial Q_2(21)}{\partial a_2} - \frac{\partial Q_2(12)}{\partial a_2} = M \underbrace{\int \underbrace{[(s_{i2}(21) - s_{i2}^2(21)) - (s_{i2}(12) - s_{i2}^2(12))]}_{>0\ if\ s_{i2}(21) \leq 0.5}(\gamma + \sigma v_i) d\Phi(v_i)}_{unclear\ sign}$$

The function $x(1-x)$ is increasing and concave, reaching maximum when $x = 0.5$. Moreover, $s_{i2}(21) > s_{i2}(12)$ since $\lambda < 0$. Therefore, when $s_{i2}(21) \leq 0.5$, the bracket-term is positive. And this inequality is likely to happen in our empirical context because the largest total market share in the data is smaller than 0.25. On the other hand, v_i may be negative. It turns out that the bracket-term could be increasing or decreasing with ${v_i}^2$. Therefore, ultimately, the difference has

[2] I derived that $\partial s_{i2} / \partial v_i = (s_{i2} - s_{i2}^2)\sigma a_2 > 0$. I further derived that the partial derivative of the bracket-term with

an unclear sign. Intuitively, this difference captures how the marginal downloads with respect to updates change with discoverability. I conclude that, while the model is flexible enough to allow both more and less marginal downloads with more discoverability, our data is likely to fit the case of discoverability increasing marginal downloads with respect to updates.

Next I analyze term II and argue that it has an unclear sign. Note that $\partial\tilde{\mathbb{P}}[21] = p_2$, where p_2 is the probability that app-2 is ranked first. I write out term II as the following

$$\Delta\frac{\partial\tilde{\mathbb{P}}[21]}{\partial a_2} = \underbrace{[(p_2' - p_2^2) - (p_2 - p_2^2)]}_{unclear\ sign}\underbrace{\gamma}_{>0}\underbrace{\theta_{quality}}_{>0}$$

I only know that $p_2' > p_2$. In fact, $p_2 \in (0,1)$ with $p_2 + p_1 = 1$ while $s_2 + s_1 < 1$ since there is out-side option for consumers to choose but all products are ranked. Therefore, I have no clue for the likely sign of the bracket-term[3]. And term II has an unclear sign.

As a summary of the illustrative example. Equation D.1 writes out the effect of the self-preferencing on developers' incentives to update apps. Equation D.2 writes out the most important term in Equation D.1, which illustrates the two offsetting economic forces. On the one hand, concave revenue curve generates strategic complementarity of product values net of search costs. This decreases independent developers' incentive to update apps with less self-preferencing of Apple's apps: my competitors perform worse, why don't I take a good rest. On the other hand, discoverability is likely to increase the marginal downloads with respect to updates. This increases independent developers' incentive to update apps with less self-preferencing of Apple's apps: my efforts are more visible now, why don't I take the chance.

Now I discuss the case in more complex markets. A general version of Equation D.1 is given by

$$\begin{aligned}\Delta MB_j = &\sum_{y' \in \mathscr{B}_a'} \left(\frac{\partial \pi_f^I(y')}{\partial a_j} \tilde{\mathbb{P}}'[y'|a] + \pi_f^I(y') \frac{\partial \tilde{\mathbb{P}}'[y'|a]}{\partial a_j} \right) \\ &- \sum_{y \in \mathscr{B}_a} \left(\frac{\partial \pi_f^I(y)}{\partial a_j} \tilde{\mathbb{P}}[y|a] + \pi_f^I(y) \frac{\partial \tilde{\mathbb{P}}[y|a]}{\partial a_j} \right)\end{aligned}$$

respect to v_i is given by $(s_{i2}(21)(1-s_{i2}(21))(1-2s_{i2}(21)) - s_{i2}(21)(1-s_{i2}(21))(1-2s_{i2}(21)))\sigma a_2$. The value of the partial derivative depends on the behavior of the function $x(1-x)(1-2x)$. This function is symmetric through $(0.5, 0)$, with the points left to $(0.5, 0)$ being positive and the points right to $(0.5, 0)$ being negative. And it is reserve U-shape before reaching (0.5,0) and U-shape after the point (0.5,0). This behavior of the function implies that even when $s_{i2}(21 \leq 0.5$, I could have smaller (less positive) bracket-term with larger v_i. However, to have positive difference for sure when $s_{i2}(21 \leq 0.5$, we need the bracket-term to be larger with larger v_i so as to surely offset negative values of v_i with positive v_is.

[3]To fit the idea, the following table is useful:

bracket-term	$p_2 < 0.5$	$p_2 > 0.5$
$p_2' < 0.5$	+	n/a
$p_2' > 0.5$	unclear	-

.

The difference between the above equation and Equation D.1 are mainly two-fold. First, the permutation set $\mathscr{B}$ is a truncated set of permutations, therefore, $y \neq y'$, or, $\mathscr{B}'_a \neq \mathscr{B}_a$.In particular, independent apps are ranked higher in every $y' \in \mathscr{B}'_a$ compared to $y \in \mathscr{B}_a$, because the most likely ordering has independent apps ranked higher, and it determines the truncation set. However, in the illustrative example, the permutation set is not changed with θ_{Apple}. This difference add complication into the analysis of ΔMB_j because I cannot take out a common term $\Delta\tilde{\mathbb{P}}[y]$ as what I did in Equation D.1. However, because independent apps are ranked higher in every $y' \in \mathscr{B}'_a$ compared to $y \in \mathscr{B}_a$, the term I in the Equation D.1 still exists in the general version. Therefore, it's possible that the intuition still applies. The next job is to check whether we have similar predictions on the sign of term I in the general version.

Analysing the term I in the general version introduces the second difference between Equation D.1 and the general case: firms are allowed to be multiple-product firms in the general version, while the illustrative example just has two single-product firms. The corresponding general version of Equation D.2 is given by:

$$\begin{aligned}\frac{\partial \pi_f^I(y')}{\partial a_j} - \frac{\partial \pi_f^I(y)}{\partial a_j} = \sum_{l \in \mathscr{J}_f} & \underbrace{[0.7\underbrace{(\tau_1 + 2\tau_2 Q_l(y'))}_{R_l^{(q)}} + \underbrace{\phi_1 + 2\phi_2 Q_l(y')}_{F_l^{(q)}}]}_{\textit{unclear relative size}} \underbrace{\frac{\partial Q_l(y')}{\partial a_j}}_{\textit{unclear}} \\ - \sum_{l \in \mathscr{J}_f} & \underbrace{[0.7\underbrace{(\tau_1 + 2\tau_2 Q_l(y))}_{R_l^{(q)'}} + \underbrace{\phi_1 + 2\phi_2 Q_l(y)}_{F_l^{(q)'}}]}_{\textit{unclear relative size}} \underbrace{\frac{\partial Q_l(y)}{\partial a_j}}_{\textit{unclear}}\end{aligned}$$

where $y' \in \mathscr{B}'_a$ and $y \in \mathscr{B}_a$. Now the bracket terms do not have unclear relative sizes, even though independent apps' rankings are ranked higher if not unchanged. This is because apps are competitors, one of the apps that are ranked higher may become so much more attractive to consumers such that it steals business from all the other apps. Similar logic applies to the analysis of relative magnitudes between $\partial Q_l(y')/\partial a_j$ and $\partial Q_l(y)/\partial a_j$: it's unclear whether $s_{il}(y')$ will be larger or smaller than $s_{il}(y)$, even though the ranking of l is in y' is no lower than that in y. Therefore, there are more ambiguity in the general case, and I focus on the illustrative example for illustrating the fundamental intuitions.

Overall, in the general case where multiple products exist in one market, who replace the platform-owned product in the higher positions matters for the competition outcome. If the boosted-up independent products are more competitive than the platform-owned products, then the strengthened product competition will provide additional incentive to work against the strategic complementarity and encourage quality provision. Otherwise, the product competition is further weakened, providing additional incentive to exacerbate the strategic complementarity and discourage quality provision. There are three possible scenarios: i) the platform-owned products have higher before-change net product values than all of the boosted-up independent products; ii) the platform-owned products have higher before-change net product values than some of the boosted-up independent products; iii) the platform-owned products have lower before-change net product values than all of the boosted-up independent products.

In the scenario i), the product competition is weakened, and strategic complementarity discourages quality provisions of all independent apps, boosted-up or not. In the scenario ii), the product competition is firstly strengthened for the independent apps that were ranked above platform-owned products. However, for the boosted-up independent apps that have lower before-change net product values than the platform-owned products, it's unclear whether their product competition might be strengthened or weakened. On the one hand, the average competitiveness of products ranked before them is higher; on the other hand, the number of products ranked before them is smaller. Lastly, for the boosted-up independent apps that have higher before-change net product values than the platform-owned products, their product competition is weakened due to less products ranked before them. On top of the potentially heterogeneous changes in product competition, there is smaller marginal revenue from downloads due to increased downloads from higher rankings. Therefore, in this scenario, it is unclear whether the combination of heterogeneous changes in product competition and concavity in marginal revenue curves will lead to more or less quality provision. In the scenario iii), the product competition is strengthened for independent products that were ranked before platform-owned products; and weakened for the boosted-up independent products. This again leads to unclear overall changes of quality provision. While the scenario i) is unlikely to happen following the idea of self-preferencing, the other two scenarios both lead to ambiguous changes in the incentives for quality provision.

On top of the changes in product competition and smaller marginal revenues from quantities, there is higher marginal quantity from quality that encourages quality provision. The three forces lead to ambiguous supply-side effects of self-preferencing in the general case.

APPENDIX E

Some Examples of Real-world Royalty Rates

Recall that Chapter 3 looks at how the estimation and prediction biases change with the portion of marginal costs accounted by royalty rates. In simulation, we examine the universe of this ratio, from zero to 100 percent. In reality, it might not be true. Therefore, this section reviews some publicly available information on real-world royalty rates.

First, as a general statement, business insiders say that "Typically, royalties are paid as a percentage of the product's gross sales. The typical percentage for royalty payments will be 3-5%." [1] To allow some differentiation, some insiders say that "Royalty rates vary per industry, but a good rule of thumb is between 2-3% on the low end, and 7-10% on the high end." [2] If we assume constant marginal costs and per-unit royalty rates, these quotes imply that the ratio of royalty rates in marginal costs would typically be *larger* than 2-10%. But how much larger? We need information on production costs.

To that end, we find some existing studies that revealed information on royalty payments for specific industries. For smartphones, Armstrong, Mueller and Syrett (2014) estimate that estimate potential patent royalties in excess of \$120 on a hypothetical \$400 smartphone — which is almost equal to the cost of device's components. This implies that the total royalty payments might account for up to 50% of marginal costs. That's said, no one company owns all the patents to produce smartphones, the estimated patent royalties in Armstrong, Mueller and Syrett (2014) include royalties for all components in a smartphone, such as cellular baseband chip, memory, wi-fi, bluetooth, global positioning system (GPS), and so on. When pinned down to one licensor, the corresponding royalty rate might be much smaller. For example, according to Qualcomn's announced royalty rate for their LTE portfolios, the royalty payment for a a \$400 device would be \$13.00, which is approximately 5.4% of the suggested production cost of one smartphone ($2 \times 120 = 240$).

For TV sets, Conlon (2012) shows a cost breakdown example for an average LCD TV sets

[1] Goldstein,Rich. "How to License a Patent for Royalties." Goldstein Patent Law. https://goldsteinpatentlaw.com/how-to-license-patent-for-royalties/

[2] Tharp,Bruce M.. 2012. "Product Licensing 101: So Let's Talk Money." core77, September 11. https://www.core77.com/posts/23366/Product-Licensing-101-So-Lets-Talk-Money

from a new dataset provided by NPD-DisplaySearch. In the table, the total royalties to produce an average television is reported as $10.00, accounting for 0.9 percent of the total reported production cost. Similarly, Philips announced the royalty rates for its licenses under the TV/STB program.[3] The total royalties sum up to 1.61 euro, approximately $1.77. Therefore, the ratio of royalty rates over marginal costs is likely to be even lower than 2% in the TV set industry.

Overall, from the general statement and the industry-specific studies, we think that it is rare to see royalty rates from one licensor accounting for a large portion of marginal cost in production.

[3]Philips. 2022. "TV & Set-Top Boxes: Our extensive TV & STB patent portfolio." https://www.philips.com/a-w/about/innovation/ips/ip-licensing/programs/tv-and-set-top-boxes.html

APPENDIX F

Additional Tables

Table F.1: App Categories in the Sample

	Non-Game App Category	Game App Category
(1)	Book	Games-Action
(2)	Business	Games-Adventure
(3)	Education	Games-Arcade
(4)	Entertainment	Games-Board
(5)	Finance	Games-Card
(6)	Food & Drink	Games-Casino
(7)	Health & Fitness	Games-Family
(8)	Lifestyle	Games-Music
(9)	Medical	Games-Puzzle
(10)	Music	Games-Racing
(11)	Navigation	Games-Role Playing
(12)	News	Games-Simulation
(13)	Newsstand	Games-Sports
(14)	Photo & Video	Games-Strategy
(15)	Productivity	Games-Trivia
(16)	Reference	Games-Word
(17)	Shopping	
(18)	Social Networking	
(19)	Sports	
(20)	Travel	
(21)	Utilities	
(22)	Weather	

Table F.2: List of Pre-installed Apps and Data Availability

App Name	Data Available?	Category	App Name	Data Available?	Category
App Store	0		Measure	1	Utilities
Calculator	1	Utilities	Messages	0	
Calendar	0	Productivity	Music	1	Music
Camera	0		News	1	News
Clock	0		Notes	0	Productivity
Compass	0	Navigation	Numbers	1	Productivity
Contacts	1	Utilities	Pages	1	Productivity
FaceTime	1	Social Networking	Passbook	0	
Files	1	Utilities	Phone	0	
Find My Friends	1	Social Networking	Photos	0	
Find My iPhone	1	Utilities	Podcasts	1	Entertainment
Game Center	0		Reminders	0	Productivity
Health	0		Safari	0	
Home	1	Lifestyle	Settings	0	
iBooks	1	Book	Stocks	1	Finance
iCloud Drive	0		Tips	1	Utilities
iMovie	1	Photo & Video	TV	1	Entertainment
iTunes Store	1	Entertainment	Videos	0	
iTunes U	1	Education	Voice Memos	1	Utilities
Keynote	1	Productivity	Wallet	0	Finance
Mail	0	Productivity	Watch	0	Utilities
Maps	1	Navigation	Weather	1	Weather

Notes: For apps without data availability, there is category information if it shows up on the Apple App Store.

Table F.3: Additional Summary Statistics and Data Sources

Variable	Mean	SD	Min	Max	Source
Panel A. App Data					
#5-star ratings(million)	0.048	0.240	0	10.11	AppTweak
#4star ratings(million)	0.006	0.028	0	1.03	AppTweak
#3-star ratings(million)	0.002	0.009	0	0.34	AppTweak
#2-star ratings(million)	0.001	0.003	0	0.13	AppTweak
#1-star ratings(million)	0.002	0.013	0	0.71	AppTweak
Average Title Match	0.024	0.029	0	0.16	constructed[a]
Average Subtitle Match	0.023	0.028	0	0.17	constructed[a]
Market Size(million)	107.800	4.691	98.55	117.70	Comscore
Pre-installs[c] :					
#Pre-installs	0.643	1.312	0	7	public info[b,c]
min.Ever Top50	2.756	6.985	0	50	SensorTower[c]
{min.Search Ranking}×Ever Top50	0.300	0.458	0	1	SensorTower[c]
Obs(app/category/month)		56,570			
Panel B. Benchmark Conversion Rates Data					
Type(free[0]/paid[1] apps)	0.35	0.48	0	1	AppTweak
Benchmark Conversion Rates	0.052	0.059	0.001	0.45	AppTweak
Obs(type/category/month)		1,337			
Number of Categories		38			
Number of Months		23			
Panel C. Search Volume Data					
Brand Keyword?	0.36	0.48	0	1	AppTweak
Search Volume	45.03	15.15	5	100	AppTweak
Obs(keyword/day)		1,246,000			
Number of Keywords		1,780			

a. Constructed based on data on title and subtitle history from AppTweak.
b. Some pre-installed apps do not have category information, and thus are not counted. For data availability of each pre-installed app, please see Table F.2.
c. Each app/category/month observation is matched with information about pre-installed apps in the category/month.

Table F.4: Summary Statistics: Search Volume and Characteristics of Apps in the Search Results of Keywords

	Top50 Search Results			Top1 Search Results		
Variable	Obs	Mean	SD	Obs	Mean	SD
Search Volume	41,974	48.29	13.88	18,789	48.90	14.07
Brand-name Keywords?	41,974	0.34	0.47	18,789	0.42	0.49
Pre-installed	41,974	0.02	0.08	18,789	0.05	0.22
Apple	41,974	0.02	0.09	18,789	0.06	0.24
Update Level	41,908	0.60	0.33	17,858	0.60	0.50
Average Rating	41,908	4.55	0.22	17,858	4.55	0.34
Age(month)	41,908	50.27	19.32	17,858	53.72	28.03
File Size (GB)	41,908	0.24	0.29	17,858	0.27	0.47
#Screenshots	41,908	5.78	1.29	17,858	5.86	1.94
Description Length(1,000 characters)	41,908	2.44	0.63	17,858	2.46	0.98
Offer In-app-purchase	41,908	0.93	0.19	17,858	0.92	0.27
Paid Installation	41,908	0.11	0.23	17,858	0.13	0.33
Price	41,908	0.53	1.57	17,858	0.73	2.68

Notes. App characteristics are reported as average levels across the observed apps in the top50 or top1 search results for a given keyword/category/month combination.

Table F.5: Estimation Results: Known App Characteristics before Search

	Top50 Search Results		Top1 Search Results	
	Search Volume			
Variable	(1)	(2)	(1)	(2)
All Pre-install?	1.176	-1.997***	5.861**	-3.135***
	(1.974)	(0.741)	(2.292)	(1.037)
Update Level	1.391***	0.464***	1.131***	0.161**
	(0.275)	(0.0758)	(0.217)	(0.0659)
Average Rating	0.328	0.0593	0.241	0.296*
	(0.333)	(0.114)	(0.286)	(0.169)
Age(month)	0.0120***	-0.0138***	0.0179***	-0.0208***
	(0.00438)	(0.00212)	(0.00370)	(0.00442)
File Size(GB)	-0.949***	0.215	-0.344	0.0588
	(0.367)	(0.165)	(0.262)	(0.199)
#Screenshots	0.785***	0.0492**	0.395***	0.00915
	(0.0692)	(0.0230)	(0.0527)	(0.0301)
Description Length(1,000 characters)	-1.112***	-0.0921*	-0.496***	0.135
	(0.143)	(0.0522)	(0.105)	(0.0865)
Offer In-app-purchase	3.141***	-1.748***	3.659***	-3.002***
	(0.625)	(0.347)	(0.514)	(0.658)
Paid Installation	-12.61***	-0.711**	-8.762***	-2.683***
	(0.572)	(0.344)	(0.456)	(0.845)
Price	-0.209***	0.000665	-0.0887*	0.154***
	(0.0774)	(0.0369)	(0.0457)	(0.0430)
Apple	10.99***	-1.611***	12.31***	-2.347***
	(1.521)	(0.508)	(1.081)	(0.838)
Pre-install	-2.582	1.485**	-10.23***	3.212***
	(1.728)	(0.595)	(2.060)	(0.979)
Brand-name Keywords?	5.327***		5.737***	
	(0.127)		(0.181)	
Constant	40.32***	50.05***	40.36***	51.22***
	(1.666)	(0.671)	(1.380)	(0.911)
Observations	41,974	41,964	18,787	18,708
R-squared	0.262	0.963	0.347	0.965
Category-Month FE	YES		YES	
Keyword-Category FE		YES		YES
Month FE		YES		YES
Mean level	48.29		48.90	

Notes. Robust standard errors in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table F.6: First-stage IV Regression Results: F statistics

Variables	Demand (1) Price	(2) Average Rating	(3) Update Level	(4) Search Ranking $\times$Ever Top50	(5) Ever Top50	Supply (6) Update Level
F^a	545.5	86.53	372.1	1759	2702	1645
Excluded F^b	43.73	58.46	113.7	66.52	239.8	40.47

(a). On the demand side, the reported F is F(93,52865). On the supply side, the reported F is F(73,25252)

(b). On the demand side, the reported excluded F is F(57,52865). On the supply side, the reported excluded F is F(50,25252).

Table F.7: First-stage IV Regression Results

Variables	Demand					Supply
	(1)	(2)	(3)	(4)	(5)	(6)
	Price	Average Rating	Update Level	Search Ranking $\times$Ever Top50	Ever Top50	Update Level
Price		0.00280***	0.00159***	0.0358***	-0.000258	0.00176**
		(0.000683)	(0.000492)	(0.0102)	(0.000315)	(0.000899)
Average Rating	0.0990***		0.0514***	-0.313***	0.0473***	
	(0.0261)		(0.00292)	(0.0670)	(0.00203)	
Update Level	0.0908***	0.0830***		-0.756***	0.0334***	
	(0.0302)	(0.00465)		(0.102)	(0.00270)	
Search Ranking $\times$ Ever Top50	0.00399***	-0.000987***	-0.00147***		0.0200***	
	(0.00109)	(0.000210)	(0.000198)		(8.80e-05)	
Ever Top50	-0.0373	0.193***	0.0842***	25.90***		
	(0.0464)	(0.00819)	(0.00684)	(0.119)		
Pre-installs:Search Ranking $\times$ Ever Top50	-0.00234	0.00155*	-5.78e-05	-0.00497	9.25e-05	
	(0.00464)	(0.000928)	(0.000720)	(0.0152)	(0.000419)	
Pre-installs: Ever Top50	0.0633	-0.0499*	0.0176	-0.167	0.0130	
	(0.138)	(0.0269)	(0.0220)	(0.455)	(0.0127)	
Apple	-0.505***	-0.182***	-0.0731**	-5.336***	0.232***	
	(0.137)	(0.0611)	(0.0286)	(0.918)	(0.0264)	
Paid Installation?	4.285***	-0.0290***	-0.228***	-0.0655	-0.188***	-0.0739***
	(0.0542)	(0.00795)	(0.00579)	(0.136)	(0.00387)	(0.00994)
Offer In-app-purchase?	-0.572***	0.0267***	0.0902***	0.367***	0.00522	0.103***
	(0.0557)	(0.00758)	(0.00450)	(0.110)	(0.00344)	(0.0110)
$\log(Age)^a$(month)	0.133***	0.00574	-0.0185***	-0.821***	0.0263***	-9.56e-05
	(0.0197)	(0.00463)	(0.00345)	(0.0745)	(0.00208)	(0.000126)
$\log(FileSize)$(MB)	0.487***	0.0135***	0.0361***	-0.347***	0.0165***	0.0503***
	(0.0175)	(0.00232)	(0.00152)	(0.0358)	(0.00106)	(0.00294)
#Screenshots	-0.0620***	0.0185***	0.0245***	0.0576***	-0.00255***	
	(0.0119)	(0.00122)	(0.00104)	(0.0214)	(0.000587)	
$\log(1+DescriptionLength)$(1,000 characters)	0.290***	0.0308***	0.0112***	-0.0865	0.0175***	
	(0.0266)	(0.00463)	(0.00311)	(0.0574)	(0.00160)	
#Pre-installs	-0.0639***	-0.00305	-0.00528	-0.680***	0.0182***	
	(0.0213)	(0.00425)	(0.00327)	(0.0685)	(0.00194)	
Game Apps?	-1.721***	-0.0182	0.0304	3.448***	-0.168***	
	(0.146)	(0.0343)	(0.0257)	(0.540)	(0.0148)	
$PromotionTextLength$	0.000355*	-2.00e-05	-6.22e-05*	0.000556	-5.25e-05**	
	(0.000195)	(3.89e-05)	(3.31e-05)	(0.000748)	(2.08e-05)	
#Icon Changes	0.0270	0.00475	0.292***	0.803***	-0.00684**	
	(0.0198)	(0.00481)	(0.00570)	(0.131)	(0.00336)	
Title Match	1.556***	-0.310***	-0.175**	-73.53***	3.568***	0.197**
	(0.519)	(0.0886)	(0.0822)	(1.900)	(0.0451)	(0.0930)
Subtitle Match	0.421	1.128***	0.727***	-11.86***	1.289***	0.531***
	(0.474)	(0.0850)	(0.0811)	(1.853)	(0.0459)	(0.0952)
Apple$\times$Post	0.323**	0.0340	0.00912	7.077***	-0.0952***	
	(0.137)	(0.0925)	(0.0354)	(1.012)	(0.0298)	
AppleCompetitor$\times$Post	0.182***	-0.0101	-0.00491	0.134	-0.00692	
	(0.0560)	(0.00964)	(0.00750)	(0.167)	(0.00462)	
#Category	0.0663***	-0.00828***	-0.00361***	-0.0240	-0.000430	
	(0.00513)	(0.00135)	(0.000916)	(0.0221)	(0.000651)	
Multiple-Category Developer?	0.268	-2.070***	-0.0105	8.622***	-0.281***	0.0434
	(0.238)	(0.0850)	(0.0409)	(1.064)	(0.0344)	(0.0658)
Other-Category Same-Developer Average:						
Price	0.187***	0.00943***	-0.00265**	0.0505*	-0.00192**	-0.0141***
	(0.0176)	(0.00131)	(0.00107)	(0.0285)	(0.000895)	(0.00234)
Update Level	0.103***	-0.0332***	0.144***	0.427***	-0.0192***	
	(0.0175)	(0.00361)	(0.00551)	(0.104)	(0.00280)	
Average Rating	0.0919*	0.453***	-0.0486***	-2.196***	0.0714***	-0.0204
	(0.0499)	(0.0181)	(0.00729)	(0.199)	(0.00648)	(0.0145)
Ever Top50	0.0624	0.0101	0.0628***	3.433***	-0.110***	
	(0.177)	(0.0356)	(0.0202)	(0.557)	(0.0173)	
Keyword Un-adoption Ratio	-1.581***	0.115***	0.0334	-4.461***	0.174***	
	(0.0993)	(0.0185)	(0.0228)	(0.491)	(0.0135)	
Days without Search Results Ratio	-0.552***	0.0115	0.0420**	3.240***	-0.124***	
	(0.153)	(0.0310)	(0.0197)	(0.549)	(0.0167)	
Average of Competing Apps' Other-Category Same-Developer Average:						
Price	-0.486***	0.0166	0.0372*	-0.0674	0.0160	0.0493***
	(0.0882)	(0.0203)	(0.0193)	(0.416)	(0.0110)	(0.0190)
Update Level	-1.154***	0.0222	-0.0819*	-0.0213	0.0130	
	(0.226)	(0.0485)	(0.0461)	(1.052)	(0.0288)	
Average Rating	0.976***	0.185***	-0.275***	0.582	-0.0747**	-0.0449***
	(0.372)	(0.0697)	(0.0588)	(1.347)	(0.0367)	(0.00916)
Ever Top50	-2.911*	-0.464	0.514**	-11.07*	0.850***	
	(1.637)	(0.301)	(0.256)	(5.950)	(0.161)	
Ratio.Keyword Un-adoption	1.808***	-0.436***	0.911***	12.17***	-0.708***	
	(0.677)	(0.163)	(0.155)	(3.190)	(0.0834)	
Ratio.Days without Search Results	-2.487*	-0.690***	0.0483	-7.261	0.590***	
	(1.455)	(0.253)	(0.233)	(5.335)	(0.146)	
Ratio.Paid Apps	-0.0392	0.00181	-0.104**	-3.164***	0.0176	0.123*
	(0.425)	(0.0668)	(0.0522)	(1.181)	(0.0327)	(0.0653)
Ratio.Multiple-Category Developers	0.772**	0.101*	-0.0293	0.660	-0.000627	0.144**
	(0.309)	(0.0593)	(0.0521)	(1.139)	(0.0319)	(0.0663)
Constant	-4.360***	3.723***	-0.00568	7.490***	-0.247***	0.437***
	(0.325)	(0.0564)	(0.0427)	(0.880)	(0.0247)	(0.0428)
Observations	52,959	52,959	52,959	52,959	52,959	25,325
Adjusted R-squared	0.372	0.177	0.335	0.644	0.751	0.072

Notes. Robust standard errors in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$. All columns include category-fixed effects and month-fixed effects.
(a). On the supply side, *age* is an included IV, thus I use *age* instead of $\log(age)$.

Table F.8: Compare Structural Estimates with Difference-in-Differences Estimates of Average Treatment Effects of the Search Algorithm Change

ATE	$\log(1+\text{Update Level})$	$\log(\text{Search Ranking})$	$\log(\text{Downloads})$
Structural Estimates	0.0166	-0.0150	0.0342
DiD Estimates	0.0212	-0.0355	0.2210

Notes. Difference-in-Differences (DiD) estimates are taken from Table 1.4. Structural estimates are computed from difference between market outcomes with and without the search algorithm change in categories with Apple's apps during the same post-change period as the DiD specification. The market outcomes are computed without fixing pre-installed apps' search rankings.

Table F.9: Effects of Fixing Pre-installed Apps' Search Rankings

Variable	Update	Search Rankings	Downloads
No-game Simulation(%)		0.0064	0.0210
Partial-game Simulation(%)	0.0002	0.0064	0.0207
Full-game Simulation(%)	0.0606	0.0055	0.0205

Notes. Figures are relative L1-Norm of each column variable y_{jgt} in percentage: $100 \times [\sum_{jgt}(|y_{jgt}^{[fix]} - y_{jgt}^{[0]}|)]/[\sum_{jgt} y_{jgt}^{[0]}]\%$, where $y^{[fix]}$ denotes the market outcome with fixing pre-installed apps' search rankings, $y^{[0]}$ denotes the market outcome with flexible pre-installed apps' search rankings as assumed in supply-side estimation. The sample covers affected markets with pre-installed apps: Entertainment category, Music category and Utilities category in the months of June and July in 2019. Search rankings and downloads are evaluated with expectation over the truncated set of possible search rankings. No-game simulation hold update levels fixed. Partial-game simulation endogenizes positive update levels but hold update portfolios fixed. Full-game simulation endogenize both positive update levels and update portfolios of top5 developers in each category.

Table F.10: Effects of Preferential Search Ranking on Positive Update Levels

	Status-quo	Shut-down	Percentage Change(%)			
Positive Update Level	mean	mean	mean	min	max	std
Market-level Average	0.7951	0.7956	0.06	0.01	0.09	0.03
Product-level	0.8060	0.8064	0.07	−0.19	4.37	0.31

Notes. Update level is $\log(1+n_{jt})$, where n_{jt} is weighted number of updates, and the weights are based on the length of release notes. Update portfolios are holding fixed. Figures reported are among independent apps with positive update levels and valid profit functions in the data. For the status-quo case, there is estimated preferential treatment of Apple's apps in the search ranking algorithm. For the shut-down case, there is no preferential search ranking. Product-level percentage change of update levels are relative to the product's status-quo update level.

Table F.11: Partial-Game Counterfactual Simulation: Effects of Preferential Search Ranking on Search Rankings, Installations and Welfare

	Variable	Status-quo	Shut-down	Mean Δ	Mean $\%\Delta$
(1)	Average Search Rankings	38.92	38.92	0	0
(2)	- Independent apps	39.86	39.23	−0.63	−1.56
(3)	- Apple's apps	14.35	31.85	17.51	142.07
(4)	Total Installations (million)	15.16	15.40	0.24	1.58
(5)	- Independent apps	15.06	15.33	0.27	1.78
(6)	- Apple's apps	0.10	0.08	−0.03	−26.59
(7)	Consumer Surplus (million \$)	321.41	321.96	0.55	0.17
(8)	Total Search Costs (million \$)	15.39	15.35	−0.04	−1.52
(9)	Total Realized Utility (million \$)	336.79	337.31	0.52	0.15
(10)	Producer Surplus (million \$)	76.60	77.01	0.41	0.62

Notes. In the partial game, all independent apps with valid profit functions and positive update levels are allowed to change their positive update levels, but update portfolios are fixed. For the status-quo case, there is estimated preferential treatment of Apple's apps in the search ranking algorithm. For the shut-down case, there is no preferential search ranking. Producer surplus are total revenues net of variable update costs across all independent apps with valid profit functions.

Table F.12: Computational Error in Consumer Surplus

	Max Relative L2 Norm		Max Relative L-infinity Norm	
	Status-quo	Shut down	Status-quo	Shut down
No Game	0.33	0.37	0.26	0.30
Partial Game	0.33	0.36	0.26	0.30
Full Game	0.33	0.37	0.26	0.30

Notes. Figures are percentage of computational errors with respect to analytical market shares. The computational error is the distance between the computational market shares from the simulated optimal sequential search model for computing consumer surplus and analytical market shares. The distance are measured by maximum relative L2 norm in the left panel, and maximum relative L-infinity norm in the right panel, across all simulated observations.

Table F.13: Average Bias and Opposite Predictions in Profitable Licensor-Licensee Mergers

	(1) Avg. True Effect	(2) Avg. Prediction Bias	(3) Opposite Prediction
Prices:			
p_A	0.099	0.002	0
p_B	0.014	0.089	0.496
p_C	0.003	0.006	0.323
Share-Weighted Avg.	0.020	0.035	0.360
Shares:			
s_A	−0.269	0.078	0.011
s_B	0.127	−0.343	0.607
s_C	0.049	0.104	0.323
Inside Shares	−0.020	−0.047	0.324
Profits:			
$\pi_A + \pi_B$	0.024	0.031	0
π_A	−0.271	0.300	0.706
π_B	1.390	−1.373	0.246
π_C	0.057	0.131	0.323
Producer Surplus	0.030	0.050	0.086
Consumer Surplus	−0.050	−0.157	0.324

Notes. This table replicates Table 3.7 for mergers that are profitable: the total profit of firms A and B is larger after merger. 0.5 percent out of the 1,000 simulated markets violate this profitability condition. Thus, the results in this table is quite similar to those in Table 3.7, qualitatively and quantitatively.

Table F.14: Average Bias and Opposite Predictions in Profitable Licensee-Licensee Mergers

	(1) Avg. True Effect	(2) Avg. Prediction Bias	(3) Opposite Prediction
Prices:			
p_A	0.013	0.002	0.053
p_B	0.173	0.000	0
p_C	0.215	0.000	0
Share-Weighted Avg.	0.131	0.001	0
Shares:			
s_A	0.243	−0.006	0
s_B	−0.165	0.000	0
s_C	−0.249	0.000	0
Inside Shares	−0.095	0.000	0
Profits:			
$\pi_B + \pi_C$	0.044	0.030	0
π_A	0.144	0.131	0.053
π_B	0.034	0.035	0.133
π_C	−0.027	0.057	0.187
Producer Surplus	0.069	0.030	0.013
Consumer Surplus	−0.279	−0.001	0

This table replicates Table 3.12 for mergers that are profitable: the total profit of firms B and C is larger after merger. 92.5 percent out of the 1,000 simulated markets violate this profitability condition. Thus, the results in this table is different than those in Table **??**. Conditional on profitability, the average prediction bias and ratio of opposite prediction are much smaller. The driving force for opposite prediction is still over-predicted price of the non-merging licensor.

APPENDIX G

Additional Figures

Figure G.1: From AppTweak: Fitness of Estimated Downloads for Actual Downloads of Apps in All-Categories Top Charts in the US Market

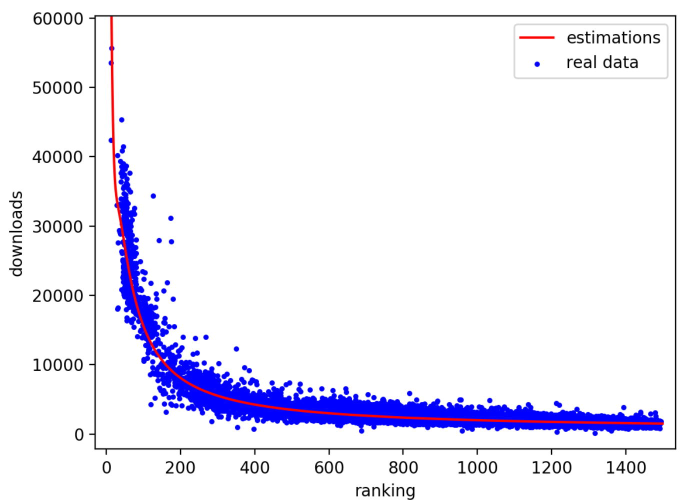
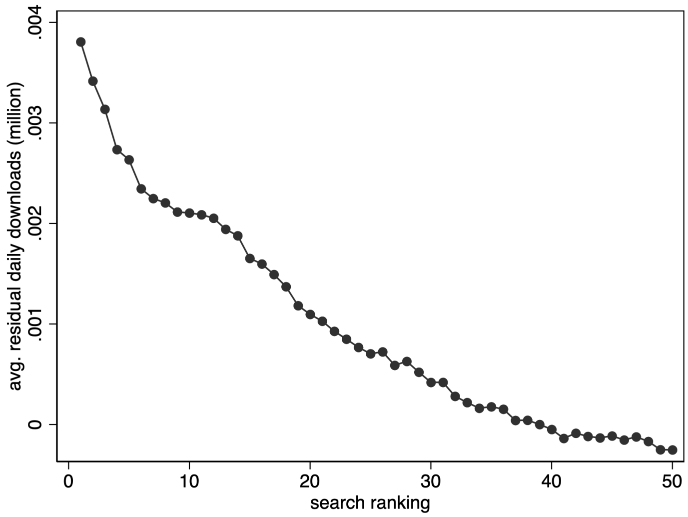


Figure G.2: Weighted Average Residual Downloads across App/Day/Keyword Given the Search Ranking



Notes. The figure plots weighted average residual daily downloads against search ranking. The residuals are from installation price, installation payment type, category-fixed effects, and daily fixed effects. The weights are based on search volumes.

Figure G.3: Average Search Rankings of Non-preinstalled Apple's Apps around the Search Algorithm Change in July 2019.

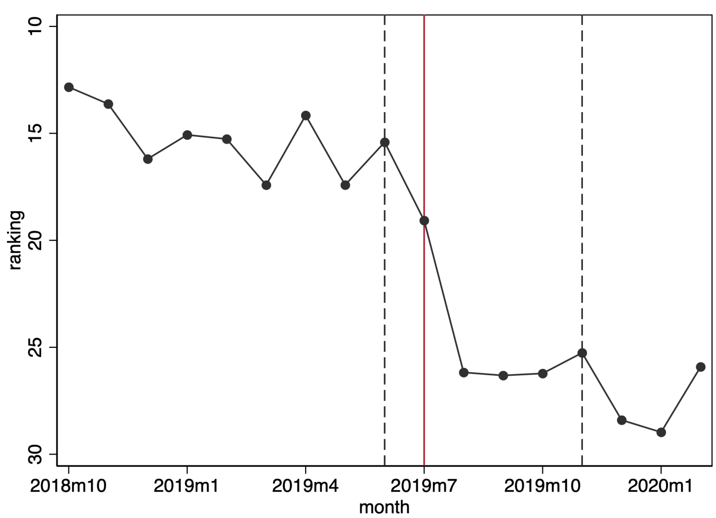
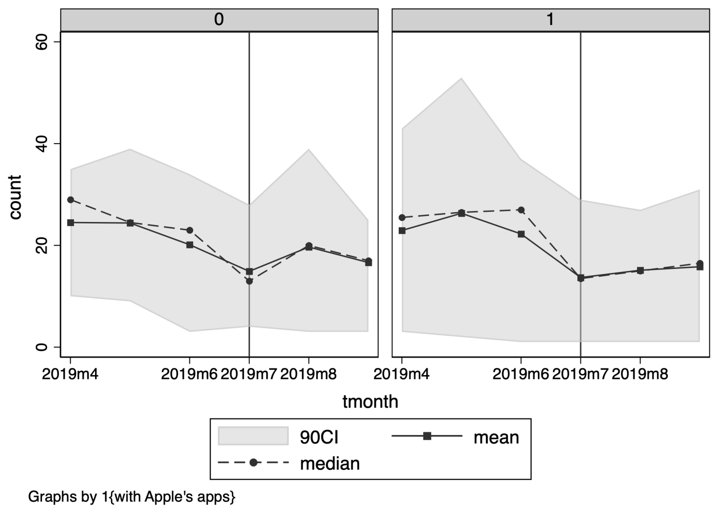
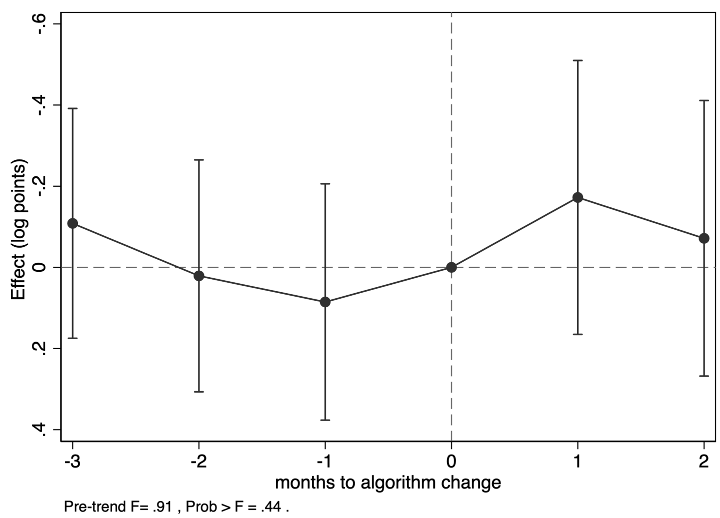


Figure G.4: Entry Around the Search Algorithm Change

Panel A. Number of New Apps



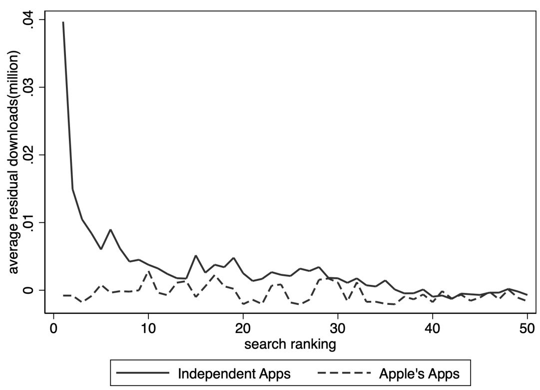
Panel B. DiD Estimates



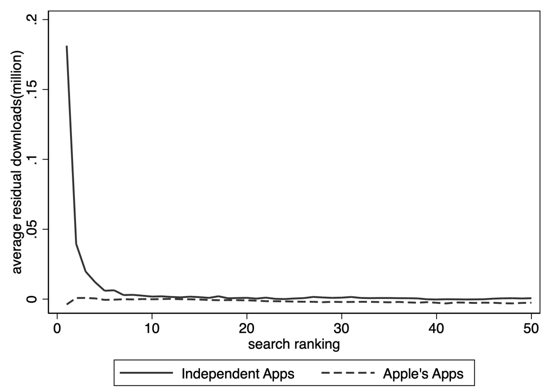
Notes. The sample to draw the figures include independent apps that were ever ranked top50 in category-specific top grossing charts during April - September 2019. Panel A shows the number of new apps in categories with Apple's apps and categories without Apple's apps. To generate Panel B, I regress the logarithm of category-month specific number of new apps as an outcome variable on the interaction terms of monthly indicator and whether the category contains Apple's apps (taking July 2019 as the reference point), as well as category-fixed effects and month-fixed effects. Panel B reports the coefficients on the interaction terms for each month. The results indicate that entry of competing independent apps did not significantly change due to the search algorithm change.

Figure G.5: Independent Apps *v.s.* Apple's Apps: Residual Downloads at Each Search Ranking, Before and After the Search Algorithm Change

Panel A. Before

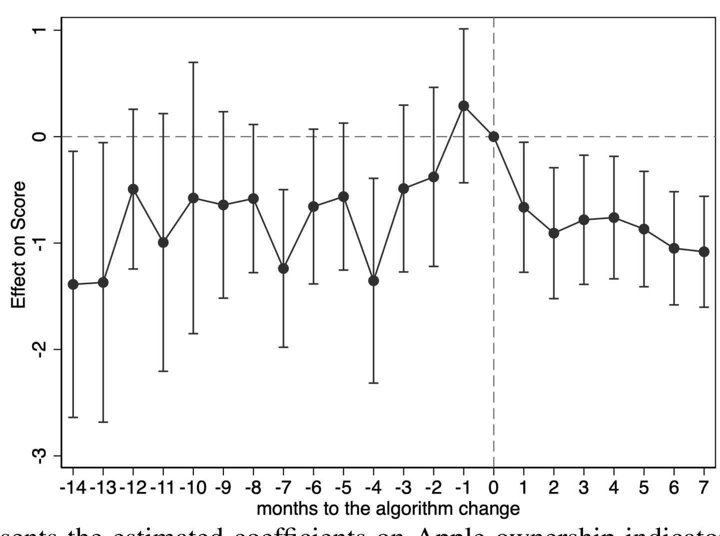


Panel B. After



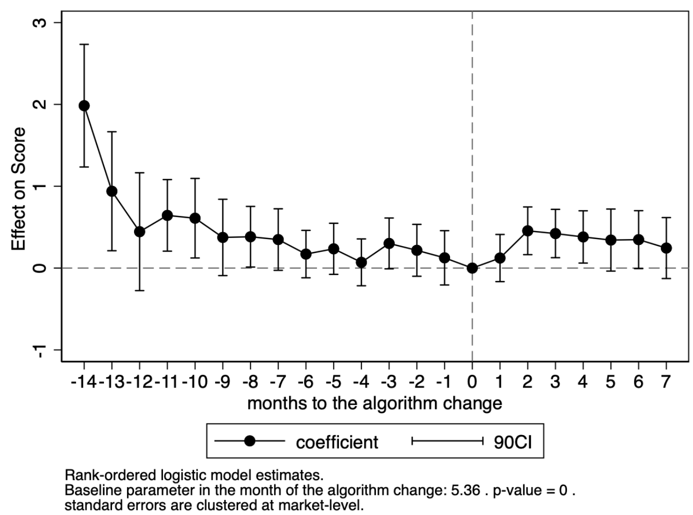
Notes. The figure compares the average residual downloads of independent apps and Apple's apps on the same position in the search results for the same keyword across different days.The residual downloads are residuals from price, installation payment type,category-fixed effects and daily fixed effects. Panel A presents the comparison result before the search algorithm change(July 2019). Panel B presents the comparison result after the search algorithm change(July 2019). While Apple's apps always need lower residual downloads to achieve the same position than independent apps, the gap is smaller after the search algorithm change.

Figure G.6: Dynamic Preferential Treatment Parameters on Apple Relative to July 2019



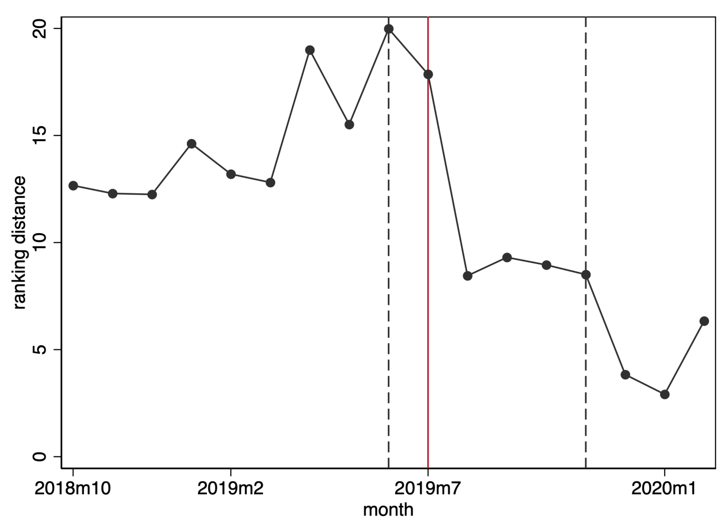
Notes. The figure presents the estimated coefficients on Apple-ownership indicator in each month, while taking the month of the search algorithm change (July 2019) as the reference point. Reported intervals are 90 percent confidence intervals. It shows that the degree of self-preferencing significantly decreased after the search algorithm change.

Figure G.7: Dynamic Search Ranking Parameters on Lagged Number of 5-star Ratings Relative to July 2019



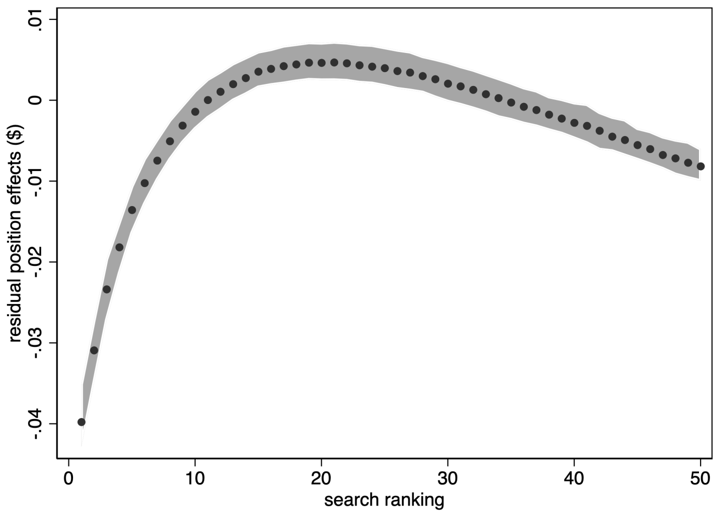
Notes. The figure presents the estimated coefficients on the one-month lagged number of 5-star ratings in each month, while taking the month of the search algorithm change (July 2019) and Business category as the reference point. It shows that the effect of 5-star ratings on search rankings became significantly more positive after the search algorithm change.

Figure G.8: Observed Rankings Relative to Rankings by Residual Downloads of Apple's Apps



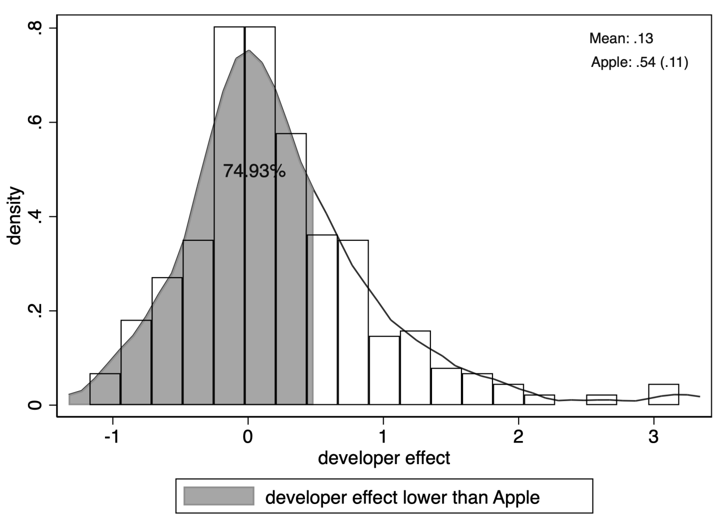
Notes. The figure presents the gap between average observed within-market search rankings of Apple's apps to average within-market rankings of residual downloads of Apple's apps in each month. The residual downloads are residuals from price, installation payment type, category-fixed effects, and daily fixed effects. It shows that an average Apple's apps would be ranked lower according to residual downloads in each of the month. More importantly, it shows that the gap were flat before April 2019, and reached peak during April and July 2019, then significantly dropped after July 2019. Such pattern is consistent with identified self-preferencing across months.

Figure G.9: Residual Position Effects across Search Rankings



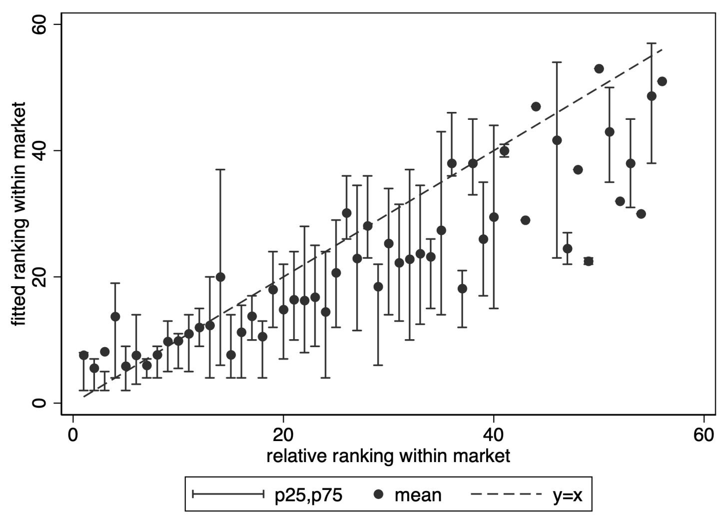
Notes. The points report average residual position effects at each integer search ranking. The residuals are from market-fixed effects. The shaded area covers the 5-th percentile and 95-th percentile of residual position effects. The U-shape curve indicates inelastic demand for apps ranked top in search results and high search costs for apps ranked low in search results.

Figure G.10: Histogram of Static Developer-fixed Effects in Search Ranking



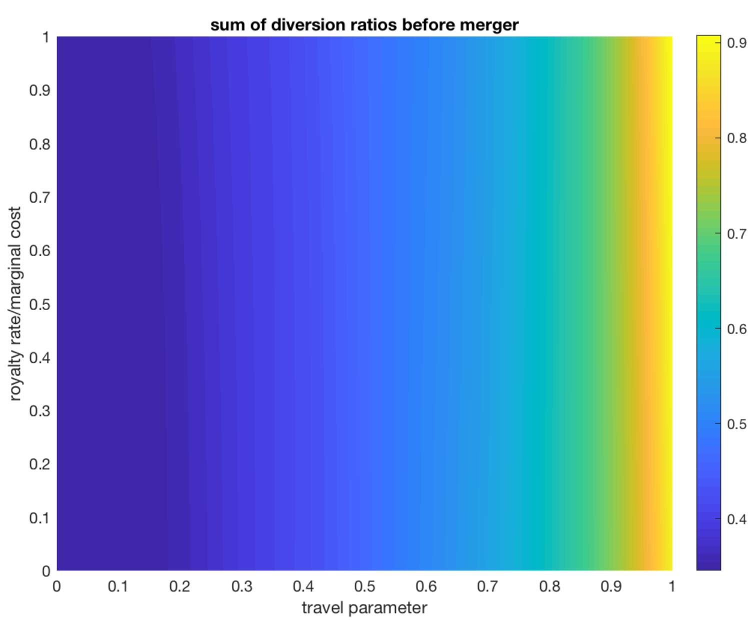
Notes. This figure presents the histogram of static developer-fixed effects across all developers. To estimate the developer-fixed effects, I replace the interaction terms between the Apple-ownership indicator and month indicators in Equation 1.12 with developer-fixed effects, where the single-product developers are normalized as the reference group. It shows that, even when allowing all developers to have their own advantage or disadvantage in the search ranking algorithm, Apple ownership still generates larger advantages than most developers.

Figure G.11: Rank-ordered Logistic Regression Model Fitness for Apple's Apps



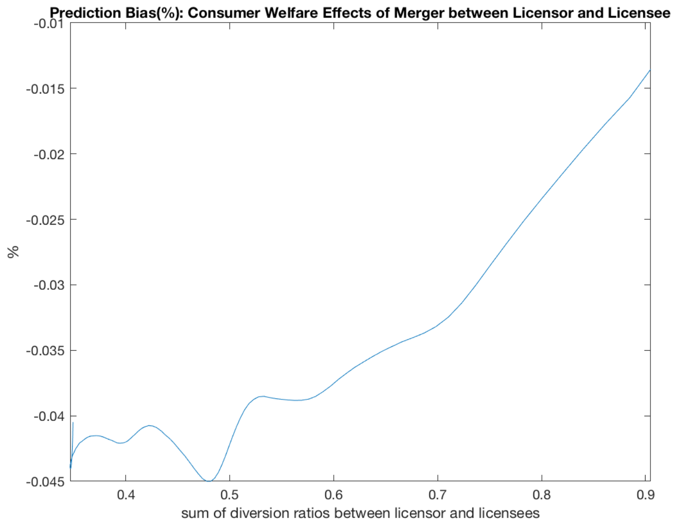
Notes. The figure presents predicted most-likely within-market ranking (y-axis) against observed within-market ranking (x-axis) across markets for Apple's apps. Bars indicate the 25 percentile and 75 percentile of fitted within-market rankings across Apple's apps that have ranked at the given observed within-market ranking.

Figure G.12: Theoretical Simulation: The Sum of Diversion Ratios between the Licensor and the Two Licensees before Merger.



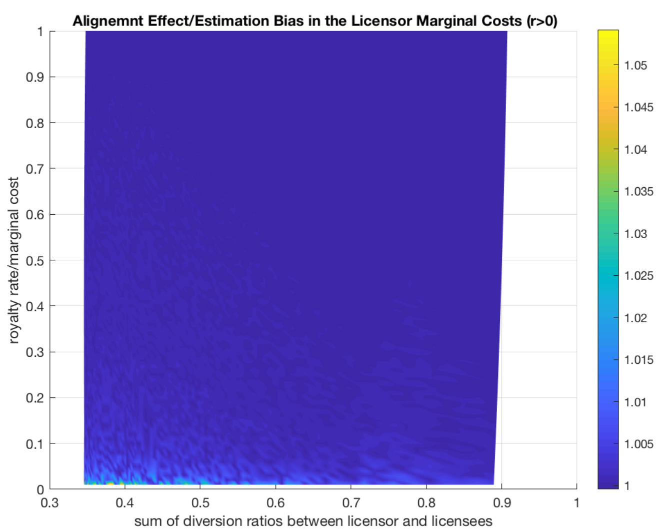
Notes. This figure shows that the equilibrium sum of diversion ratios is determined by and increases with respect to the travel parameter ρ.

Figure G.13: Theoretical Simulation: Prediction Bias on Consumer Welfare Effect of the Merger between the Licensor A and the Licensee B when royalty rate is 80% of the licensee's marginal costs.



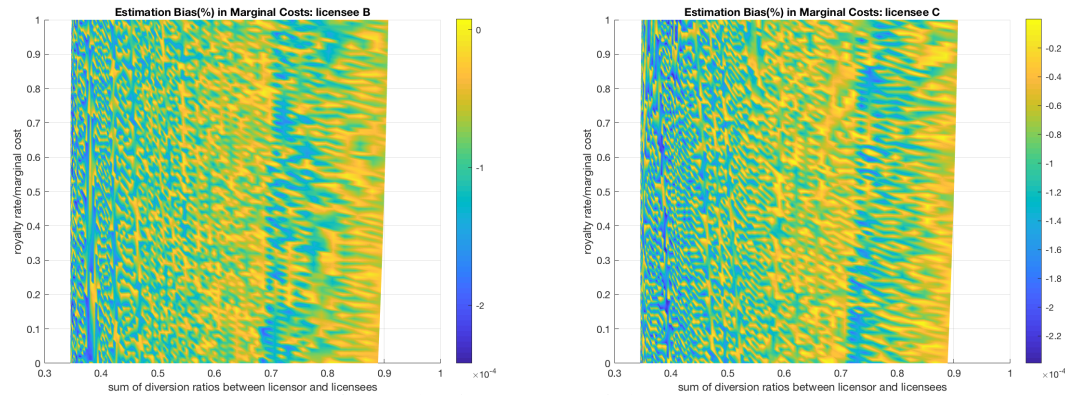
Notes. This figure illustrates the non-monotonicity of the prediction bias with respect to the sum of diversion ratios.

Figure G.14: Theoretical Simulation: Magnitude of the alignment effect divided by the estimation bias in the licensor's marginal costs.



Notes. The average ratio is 1.009, minimum ratio is 0.9996, maximum ratio is 1.0541.

Figure G.15: Theoretical Simulation: Estimation Bias in the Marginal Costs of the Licensees.



Notes. Left panel: Licensee B. Right panel: Licensee C.

Figure G.16: Guidance Simulations: estimation bias in costs between mis-specified and true models for firm A (licensor) against royalty rate, r.

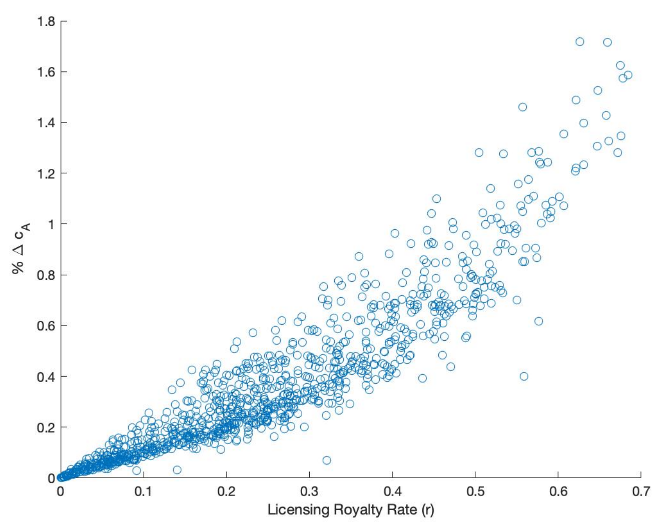
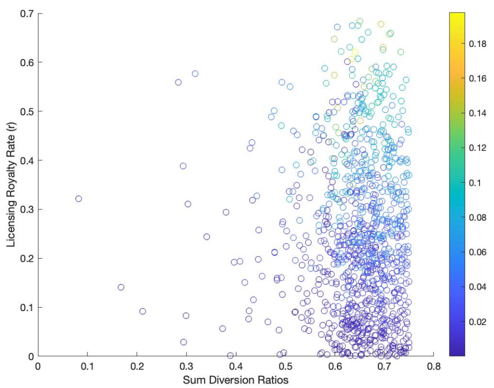
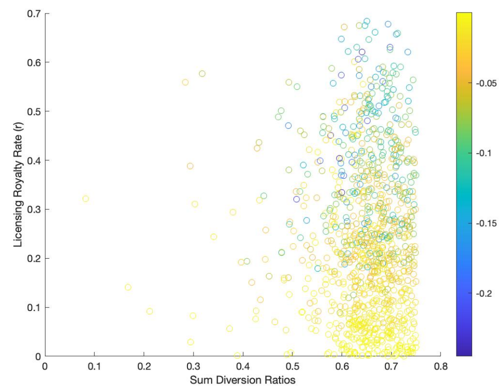


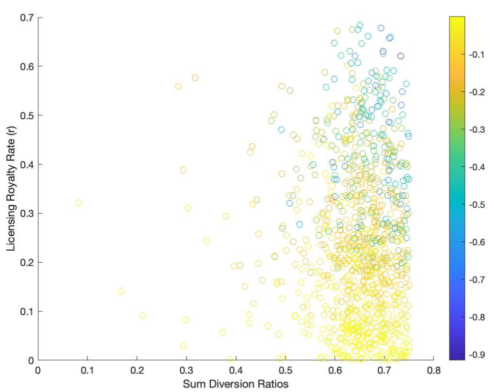
Figure G.17: Guidance Simulations: prediction bias in the effects of the merger between licensor and licensee.



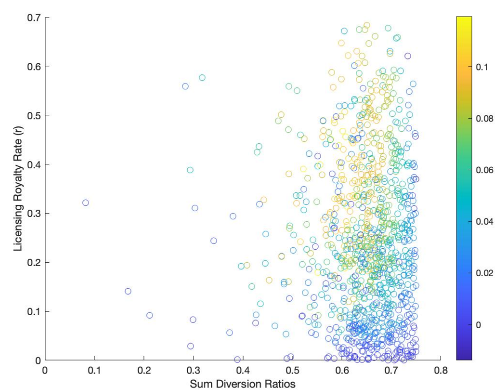
(a) Share-Weighted Prices



(b) Inside Shares



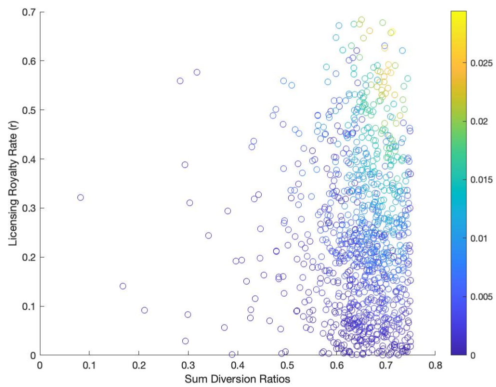
(c) Consumer Surplus



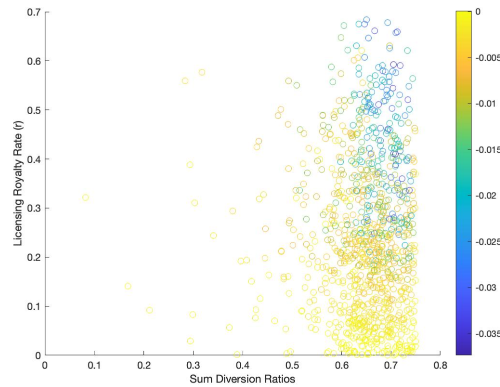
(d) Producer Surplus

Notes. Here, we can observe the relationship between the bias in predicted changes in equilibrium objects and model primitives. In Figure G.17c, we see that, for a given sum of diversion ratios between the licensor and each respective licensee, the overstated decrease in consumer surplus from the licensor-licensee merger is increasing in the royalty rate, r. In terms of producer surplus, for a given sum of diversion ratios, the overstated increase in producer surplus from the merger is increasing in the royalty rate, r.

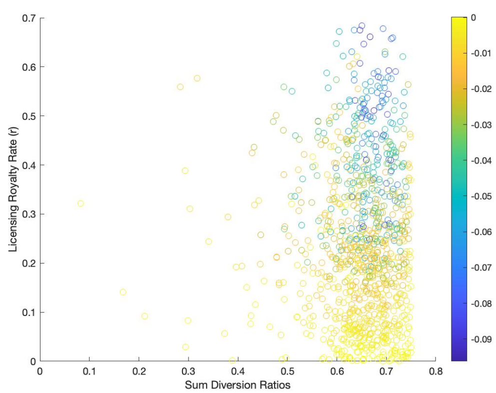
Figure G.18: Guidance Simulations: prediction bias in the effects of the merger between licensees.



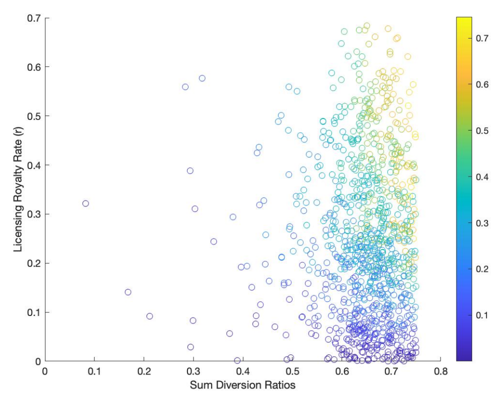
(a) Share-Weighted Prices



(b) Inside Shares



(c) Consumer Surplus



(d) Producer Surplus

BIBLIOGRAPHY

Abaluck, Jason, and Vilsa Curto. 2018. "What do consumers consider before they choose?" ASHECON.

Aksoy-Pierson, Margaret, Gad Allon, and Awi Federgruen. 2013. "Price competition under mixed multinomial logit demand functions." *Management Science*, 59(8): 1817–1835.

Allen, Jason, Robert Clark, and Jean-François Houde. 2019. "Search frictions and market power in negotiated-price markets." *Journal of Political Economy*, 127(4): 1550–1598.

Allon, Gad, Georgios Askalidis, Randall Berry, Nicole Immorlica, Ken Moon, and Amandeep Singh. 2021. "When to be agile: Ratings and version updates in mobile apps." *Management Science*.

Anderson, Simon P, and Régis Renault. 2018. "Firm pricing with consumer search." In *Handbook of Game Theory and Industrial Organization, Volume II*. Edward Elgar Publishing.

Armstrong, Ann, Joseph Mueller, and Tim Syrett. 2014. "The smartphone royalty stack: Surveying royalty demands for the components within modern smartphones." *Available at SSRN 2443848*.

Armstrong, Mark, and Jidong Zhou. 2011. "Paying for prominence." *The Economic Journal*, 121(556): F368–F395.

Arnosti, Nick, Ramesh Johari, and Yash Kanoria. 2014. "Managing congestion in dynamic matching markets." 451–451.

Balan, David J, and Keith Brand. 2018. "Simulating hospital merger simulations." *Available at SSRN 3153109*.

Bar-Isaac, Heski, Guillermo Caruana, and Vicente Cuñat. 2012. "Search, design, and market structure." *American Economic Review*, 102(2): 1140–60.

Baye, Michael R, John Morgan, and Patrick Scholten. 2004. "Price dispersion in the small and in the large: Evidence from an internet price comparison site." *The Journal of Industrial Economics*, 52(4): 463–496.

Beggs, Steven, Scott Cardell, and Jerry Hausman. 1981. "Assessing the potential demand for electric cars." *Journal of econometrics*, 17(1): 1–19.

Berry, Steven, Alon Eizenberg, and Joel Waldfogel. 2016. "Optimal product variety in radio markets." *The RAND Journal of Economics*, 47(3): 463–497.

Berry, Steven, James Levinsohn, and Ariel Pakes. 1995. "Automobile prices in market equilibrium." *Econometrica: Journal of the Econometric Society*, 841–890.

Bresnahan, Timothy, Joe Orsini, and Pai-Ling Yin. 2014. "Platform choice by mobile app developers." *NBER working paper*.

Brown, Zach Y. 2017. "An empirical model of price transparency and markups in health care." *Manuscript*.

Brown, Zach Y. 2019. "Equilibrium effects of health care price information." *Review of Economics and Statistics*, 101(4): 699–712.

Caplin, Andrew, and Barry Nalebuff. 1991. "Aggregation and imperfect competition: On the existence of equilibrium." *Econometrica: Journal of the Econometric Society*, 25–59.

Chen, Yuxin, and Song Yao. 2017. "Sequential search with refinement: Model and application with click-stream data." *Management Science*, 63(12): 4345–4365.

Choi, Hana, and Carl F Mela. 2019. "Monetizing online marketplaces." *Marketing Science*, 38(6): 948–972.

Chu, Chenghuan Sean. 2010. "The effect of satellite entry on cable television prices and product quality." *The RAND Journal of Economics*, 41(4): 730–764.

Coey, Dominic, Bradley J Larsen, and Brennan C Platt. 2020. "Discounts and deadlines in consumer search." *American Economic Review*, 110(12): 3748–85.

Compiani, Giovanni, Gregory Lewis, Sida Peng, and Will Wang. 2021. "Online Search and Product Rankings: A Double Index Approach." *Available at SSRN 3898134*.

Conlon, Christopher T. 2012. "A dynamic model of prices and margins in the LCD TV industry." *unpublished, Columbia University.[818, 835] Daley, Brendan and Brett Green (2012), "Waiting for news in the market for lemons." Econometrica*, 80: 1433–1504.

Crawford, Gregory S. 2012. "Endogenous product choice: A progress report." *International Journal of Industrial Organization*, 30(3): 315–320.

Crawford, Gregory S, and Ali Yurukoglu. 2012. "The welfare effects of bundling in multichannel television markets." *American Economic Review*, 102(2): 643–85.

Crawford, Gregory S, Oleksandr Shcherbakov, and Matthew Shum. 2019. "Quality overprovision in cable television markets." *American Economic Review*, 109(3): 956–95.

Crooke, Philip, Luke Froeb, Steven Tschantz, and Gregory J Werden. 1999. "Effects of assumed demand form on simulated postmerger equilibria." *Review of Industrial Organization*, 15(3): 205–217.

Das Varma, Gopal, and Martino De Stefano. 2020. "Equilibrium analysis of vertical mergers." *The Antitrust Bulletin*, 65(3): 445–458.

De los Santos, Babur, Ali Hortaçsu, and Matthijs R Wildenbeest. 2012. "Testing models of consumer search using data on web browsing and purchasing behavior." *American economic review*, 102(6): 2955–80.

Diamond, Peter A. 1971. "A model of price adjustment." *Journal of economic theory*, 3(2): 156–168.

Dinerstein, Michael, Liran Einav, Jonathan Levin, and Neel Sundaresan. 2018. "Consumer price search and platform design in internet commerce." *American Economic Review*, 108(7): 1820–59.

Domnenko, Gleb B, and David S Sibley. 2020. "Simulating Vertical Mergers and the Vertical GUPPI Approach." *Available at SSRN 3606641*.

Draganska, Michaela, Michael Mazzeo, and Katja Seim. 2009. "Beyond plain vanilla: Modeling joint product assortment and pricing decisions." *QME*, 7(2): 105–146.

Dranove, David, Daniel Kessler, Mark McClellan, and Mark Satterthwaite. 2003. "Is more information better? The effects of "report cards" on health care providers." *Journal of political Economy*, 111(3): 555–588.

Dutra, Jéssica, and Tarun Sabarwal. 2020. "Antitrust analysis with upward pricing pressure and cost efficiencies." *PloS one*, 15(1): e0227418.

Eizenberg, Alon. 2014. "Upstream innovation and product variety in the us home pc market." *Review of Economic Studies*, 81(3): 1003–1045.

Ellison, Glenn, and Sara Fisher Ellison. 2018. "Match quality, search, and the Internet market for used books." National Bureau of Economic Research.

Ershov, Daniel. 2020. "Consumer product discovery costs, entry, quality and congestion in online markets." *Unpublished manuscript*.

Fan, Ying. 2013. "Ownership consolidation and product characteristics: A study of the US daily newspaper market." *American Economic Review*, 103(5): 1598–1628.

Fan, Ying, and Chenyu Yang. 2020*a*. "Competition, product proliferation, and welfare: A study of the US smartphone market." *American Economic Journal: Microeconomics*, 12(2): 99–134.

Fan, Ying, and Chenyu Yang. 2020*b*. "Merger, product variety and firm entry: The retail craft beer market in California."

Fershtman, Chaim, Arthur Fishman, and Jidong Zhou. 2018. "Search and categorization." *International Journal of Industrial Organization*, 57: 225–254.

Fishman, Arthur, and Nadav Levy. 2015. "Search costs and investment in quality." *The Journal of Industrial Economics*, 63(4): 625–641.

Fradkin, Andrey. 2017. "Search, matching, and the role of digital marketplace design in enabling trade: Evidence from airbnb."

Galenianos, Manolis, and Alessandro Gavazza. 2017. "A structural model of the retail market for illicit drugs." *American Economic Review*, 107(3): 858–96.

Ghose, Anindya, and Sang Pil Han. 2014. "Estimating demand for mobile applications in the new economy." *Management Science*, 60(6): 1470–1488.

Ghose, Anindya, Panagiotis G Ipeirotis, and Beibei Li. 2014. "Examining the impact of ranking on consumer behavior and search engine revenue." *Management Science*, 60(7): 1632–1654.

Giulietti, Monica, Michael Waterson, and Matthijs Wildenbeest. 2014. "Estimation of search frictions in the b ritish electricity market." *The Journal of Industrial Economics*, 62(4): 555–590.

Goeree, Michelle Sovinsky. 2008. "Limited information and advertising in the US personal computer industry." *Econometrica*, 76(5): 1017–1074.

Goldfarb, Avi, and Catherine Tucker. 2019. "Digital economics." *Journal of Economic Literature*, 57(1): 3–43.

Hausman, Jerry A, and Timothy F Bresnahan. 2008. *5. Valuation of New Goods under Perfect and Imperfect Competition.* University of Chicago Press.

Heiss, Florian, and Viktor Winschel. 2008. "Likelihood approximation by numerical integration on sparse grids." *journal of Econometrics*, 144(1): 62–80.

Hong, Han, and Matthew Shum. 2006. "Using price distributions to estimate search costs." *The RAND Journal of Economics*, 37(2): 257–275.

Honka, Elisabeth. 2014. "Quantifying search and switching costs in the US auto insurance industry." *The RAND Journal of Economics*, 45(4): 847–884.

Honka, Elisabeth, Ali Hortaçsu, and Maria Ana Vitorino. 2017. "Advertising, consumer awareness, and choice: Evidence from the US banking industry." *The RAND Journal of Economics*, 48(3): 611–646.

Honka, Elisabeth, and Pradeep Chintagunta. 2017. "Simultaneous or sequential? search strategies in the us auto insurance industry." *Marketing Science*, 36(1): 21–42.

Hortaçsu, Ali, and Chad Syverson. 2004. "Product differentiation, search costs, and competition in the mutual fund industry: A case study of S&P 500 index funds." *The Quarterly journal of economics*, 119(2): 403–456.

Hristakeva, Sylvia. 2019. "Vertical contracts with endogenous product selection: An empirical analysis of vendor-allowance contracts." *Available at SSRN 3506265*.

Huang, Justin T. 2018. "Visibility Policy, Seller Incentives, and Pricing Dynamics in a Digital Goods Marketplace."

Janssen, Rebecca, Reinhold Kesler, Michael Kummer, and Joel Waldfogel. 2021. "GDPR and the Lost Generation of Innovative Apps."

Jolivet, Grégory, and Hélène Turon. 2019. "Consumer search costs and preferences on the internet." *The Review of Economic Studies*, 86(3): 1258–1300.

Kim, Jun B, Paulo Albuquerque, and Bart J Bronnenberg. 2010. "Online demand under limited consumer search." *Marketing science*, 29(6): 1001–1023.

Kim, Jun B, Paulo Albuquerque, and Bart J Bronnenberg. 2017. "The probit choice model under sequential search with an application to online retailing." *Management Science*, 63(11): 3911–3929.

Koulayev, Sergei. 2014. "Search for differentiated products: identification and estimation." *The RAND Journal of Economics*, 45(3): 553–575.

Lam, H Tai. 2021. "Platform Search Design and Market Power."

Layne-Farrar, Anne, and Josh Lerner. 2011. "To join or not to join: Examining patent pool participation and rent sharing rules." *International Journal of Industrial Organization*, 29(2): 294–303.

Lee, Kwok Hao, and Leon Musolff. 2021. "Entry Into Two-Sided Markets Shaped By Platform-Guided Search."

Lerner, Josh, and Jean Tirole. 2004. "Efficient patent pools." *American Economic Review*, 94(3): 691–711.

Lerner, Josh, Marcin Strojwas, and Jean Tirole. 2007. "The design of patent pools: The determinants of licensing rules." *The RAND Journal of Economics*, 38(3): 610–625.

Leyden, Benjamin T. 2019. "There's an app (update) for that: Understanding product updating under digitization." Working paper, Cornell University.

Leyden, Benjamin T. 2021. "Platform Design and Innovation Incentives: Evidence from the Product Ratings System on Apple's App Store."

Li, Xing, Timothy Bresnahan, and Pai-Ling Yin. 2016. "Paying incumbents and customers to enter an industry: Buying downloads." *Available at SSRN 2834564*.

Mazzeo, Michael J, Katja Seim, and Mauricio Varela. 2018. "The welfare consequences of mergers with endogenous product choice." *The Journal of Industrial Economics*, 66(4): 980–1016.

McFadden, Daniel, et al. 1973. "Conditional logit analysis of qualitative choice behavior."

Mehta, Nitin, Surendra Rajiv, and Kannan Srinivasan. 2003. "Price uncertainty and consumer search: A structural model of consideration set formation." *Marketing science*, 22(1): 58–84.

Mendelson, Haim, and Ken Moon. 2016. "Growth and customer loyalty: Evidence from the app economy."

Mendelson, Haim, and Ken Moon. 2018. "Modeling success and engagement for the app economy." 569–578.

Miller, Nathan H, Marc Remer, Conor Ryan, and Gloria Sheu. 2016. "Pass-through and the prediction of merger price effects." *The Journal of Industrial Economics*, 64(4): 683–709.

Miller, Nathan H, Marc Remer, Conor Ryan, and Gloria Sheu. 2017. "Upward pricing pressure as a predictor of merger price effects." *International Journal of Industrial Organization*, 52: 216–247.

Moraga-González, José Luis, and Matthijs R Wildenbeest. 2008. "Maximum likelihood estimation of search costs." *European Economic Review*, 52(5): 820–848.

Moraga-González, José Luis, Zsolt Sándor, and Matthijs R Wildenbeest. 2018. "Consumer Search and Prices in the Automobile Market."

Moraga-González, José L, Zsolt Sándor, and Matthijs R Wildenbeest. 2015. "Consumer search and prices in the automobile market."

Murry, Charles, and Yiyi Zhou. 2020. "Consumer search and automobile dealer colocation." *Management Science*, 66(5): 1909–1934.

Nevo, Aviv. 2001. "Measuring market power in the ready-to-eat cereal industry." *Econometrica*, 69(2): 307–342.

Nosko, Chris. 2010. "Competition and quality choice in the cpu market."

Nosko, Chris, and Steven Tadelis. 2015. "The limits of reputation in platform markets: An empirical analysis and field experiment." National Bureau of Economic Research.

Orhun, A Yeşim, Sriram Venkataraman, and Pradeep K Chintagunta. 2016. "Impact of competition on product decisions: Movie choices of exhibitors." *Marketing Science*, 35(1): 73–92.

Orlov, Eugene. 2015. "The effect of the internet on performance and quality: Evidence from the airline industry." *Review of Economics and Statistics*, 97(1): 180–194.

Pires, Tiago. 2016. "Costly search and consideration sets in storable goods markets." *Quantitative Marketing and Economics*, 14(3): 157–193.

Ratchford, Brian T. 2009. "Consumer search behavior and its effect on markets."

Reisinger, Markus, and Emanuele Tarantino. 2019. "Patent pools, vertical integration, and downstream competition." *The RAND Journal of Economics*, 50(1): 168–200.

Roberts, John H, and James M Lattin. 1991. "Development and testing of a model of consideration set composition." *Journal of Marketing Research*, 28(4): 429–440.

Salz, Tobias. 2020. "Intermediation and competition in search markets: An empirical case study." National Bureau of Economic Research.

Santos, Babur De Los, Ali Hortaçsu, and Matthijs R Wildenbeest. 2017. "Search with learning for differentiated products: Evidence from e-commerce." *Journal of Business & Economic Statistics*, 35(4): 626–641.

Seiler, Stephan. 2013. "The impact of search costs on consumer behavior: A dynamic approach." *Quantitative Marketing and Economics*, 11(2): 155–203.

Seim, Katja. 2006. "An empirical model of firm entry with endogenous product-type choices." *The RAND Journal of Economics*, 37(3): 619–640.

Sheu, Gloria, and Charles Taragin. 2020. "Simulating Mergers in a Vertical Supply Chain with Bargaining."

Singh, Amandeep, Kartik Hosanagar, and Aviv Nevo. 2021. "Network Externalities and Cross-Platform App Development in Mobile Platforms." *Available at SSRN 3911638*.

Small, Kenneth A, and Harvey S Rosen. 1981. "Applied welfare economics with discrete choice models." *Econometrica: Journal of the Econometric Society*, 105–130.

Stahl, Dale O. 1989. "Oligopolistic pricing with sequential consumer search." *The American Economic Review*, 700–712.

Stigler, George J. 1961. "The economics of information." *Journal of political economy*, 69(3): 213–225.

Tadelis, Steven, and Florian Zettelmeyer. 2015. "Information disclosure as a matching mechanism: Theory and evidence from a field experiment." *American Economic Review*, 105(2): 886–905.

Ursu, Raluca M. 2018. "The power of rankings: Quantifying the effect of rankings on online consumer search and purchase decisions." *Marketing Science*, 37(4): 530–552.

Vives, Xavier, et al. 2001. "Oligopoly Pricing: Old Ideas and New Tools." *MIT Press Books*, 1.

Wang, Peichun. 2017. "Innovation is the new competition: Product portfolio choices with product life cycles."

Wang, Quan, Beibei Li, and Param Vir Singh. 2018. "Copycats vs. original mobile apps: A machine learning copycat-detection method and empirical analysis." *Information Systems Research*, 29(2): 273–291.

Watson, Randal. 2009. "Product variety and competition in the retail market for eyeglasses." *The Journal of Industrial Economics*, 57(2): 217–251.

Weitzman, Martin L. 1979. "Optimal search for the best alternative." *Econometrica: Journal of the Econometric Society*, 641–654.

Wollmann, Thomas G. 2018. “Trucks without bailouts: Equilibrium product characteristics for commercial vehicles.” *American Economic Review*, 108(6): 1364–1406.

Yao, Song, and Carl F Mela. 2011. “A dynamic model of sponsored search advertising.” *Marketing Science*, 30(3): 447–468.