# Reinforcement Learning based Sequential and Robust Bayesian Optimal Experimental Design

by

Wanggang Shen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in the University of Michigan
2023

Doctoral Committee:

Assistant Professor Xun Huan, Chair
Assistant Professor Nikola Banovic
Professor Krishna Garikipati
Professor Youssef Marzouk, MIT

Wanggang Shen
wgshen@umich.edu
ORCID iD: 0000-0002-6824-9393

# ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Xun Huan, for his unwavering and continuous support and encouragement throughout my doctoral journey. His expertise and invaluable insights have been crucial in shaping the direction of my research, and his constructive feedback has been essential in refining my research work. Engaging in my PhD study under the mentorship of Xun has been an incredibly enjoyable experience. It is a tremendous honor for me to have commenced my doctoral journey at the same time he joined the faculty of the University of Michigan. I am also immensely grateful to the members of my thesis committee, Prof. Marzouk, Prof. Garikipati, and Prof. Banovic. Their support has significantly enriched the quality of this thesis.

I shall extend my sincere appreciation to all my labmates in the UQ-SciML group, colleagues, and friends for creating an excellent research and living environment. Special thanks to Jiayuan Dong for his contributions to this thesis, and to Chengyang Huang for helping to organize my dissertation defense.

I would like to acknowledge the financial support provided by the Defense Advanced Research Projects Agency (DARPA), the National Science Foundation (NSF), the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR), and the Michigan Institute for Computational Discovery and Engineering (MICDE), which made this research possible.

Lastly, I am thankful to my family for their constant support, and all the collaborators I have had the opportunity to work with during my projects.

# TABLE OF CONTENTS

# LIST OF FIGURES

FIGURE

# LIST OF TABLES

TABLE

# LIST OF APPENDICES

# ABSTRACT

Optimal experimental design (OED) is a statistical approach aimed at designing experiments in order to extract maximum information from them. It entails carefully selecting experimental conditions to effectively achieve specific objectives, such as minimizing the uncertainty associated with the model parameters. OED is highly valuable in various fields such as engineering, physics, chemistry, and biology to optimize the performance of a system or to gain a deeper understanding of a phenomenon. While conventional OED approaches predominantly focus on batch experimental designs that maximize expected information gain on model parameters, there remain active research questions that merit further investigation:

- How can we optimally design a sequence of experiments, and fully capture information offered by earlier experiments to adaptive update the later ones?
- How can we expand the OED objective function to include other design metrics beyond model parameter inference, such as model discrimination and goal-oriented predictions?
- How can we incorporate robustness into OED?

To address these questions, we first present a mathematical framework and computational methods to optimally design a finite number of sequential experiments. We formulate this sequential OED (sOED) problem as a finite-horizon partially observable Markov decision process (POMDP) in a Bayesian setting and with information-theoretic utilities. sOED then seeks an optimal design policy that incorporates elements of both feedback and lookahead, generalizing the suboptimal batch and greedy designs. We solve for the sOED policy numerically via policy gradient (PG) methods from reinforcement learning, and provide a derivation of the PG expression for sOED. Adopting an actor-critic approach, we parameterize the policy and value functions using deep neural networks and improve them using gradient estimates produced from simulated episodes of designs and observations. The overall PG-sOED method is validated on a linear-Gaussian benchmark, and its advantages over batch and greedy designs are demonstrated through a contaminant source inversion problem in a convection-diffusion field. Building upon sOED, we introduce variational sequential OED (vsOED) to further accelerate the designing process. Specifically, we adopt a lower bound estimator for the expected utility through variational approximation to the Bayesian posteriors.

The optimal design policy is solved numerically by simultaneously maximizing the variational lower bound and performing policy gradient updates. We demonstrate this general methodology for a range of OED problems targeting parameter inference, model discrimination, and goal-oriented prediction. These cases encompass explicit and implicit likelihoods, nuisance parameters, and physics-based partial differential equation models. Our vsOED results indicate substantially improved sample efficiency and reduced number of forward model simulations compared to previous sequential design algorithms.

In order to design experiments in a robust manner, we further introduce robust OED (rOED). We employ the utility variance as a measure of design robustness and introduce a variance-penalized objective formulation that tradeoff between maximizing expected utility (optimality) and minimizing utility variance (robustness). To accurately estimate the variance-penalized objective, we propose a double-nested Monte Carlo estimator, enhanced by efficient sampling techniques for improved efficiency. The accuracy and convergence of the proposed estimator is validated on benchmark examples and a sensor placement problem for source inversion in a diffusion field with building obstacles. Lastly, we formulate robust sequential OED (rsOED) that combines the principles of sequential design with the variance-penalized robust objective. We provide a solution algorithm enabled by deriving the policy gradient expressions of rsOED, and validate its performance on a nonlinear numerical example.

# CHAPTER 1

# Introduction

## 1.1 Background and motivation

### 1.1.1 Optimal experimental design

Experiments are indispensable for scientific research and play a crucial role in advancing knowledge and developing models. The data collected from experiments can provide valuable information for refining and validating our models, which is pivotal for understanding the underlying processes described by these models. However, conducting experiments and gathering data can be expensive and time-consuming, and not all experiments yield an equal amount of information. Therefore, carefully designed experiments have the potential to provide substantial resource savings.

Designs based on heuristics are generally not optimal, especially when dealing with high-dimensional, nonlinear complex systems under uncertain and noisy environments. Leveraging a model that simulates the experiment process, the research of *optimal experimental design (OED)* seeks to systematically quantify and maximize the value of experiments. In order to identify these high-value experiments, it is important to first specify a criterion that appropriately measures the value of an experiment. A relevant and suitable criterion choice can vary from problem to problem depending on the specific goals of the experiments. For example, when the goal is to learn particular unknown parameters of a model, the criterion may entail the degree of uncertainty reduction on those parameters; if the goal is to improve the prediction of certain quantities of interest (QoIs) computed by the model, the criterion may pertain to the reduction of predictive uncertainty of those QoIs; and if the goal is to select the most plausible model among a set of candidate models, the criterion may involve metrics for model selection.

Historically, OED with linear models [50, 4, 133] uses criteria based on the information matrix to maximize the value of experiments. Different operations on this matrix lead to the well-known alphabetical designs [18]: $A$-optimality minimizes the trace of the inverse of the information matrix, $D$-optimality maximizes the determinant of the information matrix, and $E$-optimality maximizes the minimum eigenvalue of the information matrix, etc. Bayesian OED further incorporates prior

and posterior distributions that reflect the uncertainty reduction from the experiment [34, 13, 33]. In particular, Bayesian $D$-optimal design generalizes to the nonlinear setting under an information-theoretic perspective [92] by capturing the expected Kullback–Leibler (KL) divergence from the prior to the posterior (equivalently, the expected information gain (EIG) on the model parameters).

Although Bayesian OED criteria can be evaluated analytically for the linear case, they are intractable for nonlinear models and must require numerical approximations [21, 53, 33, 108, 123]. Common approximation techniques include linearization on nonlinear models [21, 53] and Laplace approximation on the posterior distributions [97]. With advances in computing power and a need to tackle problems with greater size and complexity, strategy has shifted towards enhancing Bayesian OED capabilities to handle increasingly complex models without compromising on their nonlinear, non-Gaussian nature [110, 108]. For example, a double-nested Monte Carlo (MC) sampling technique has been proposed to estimate the EIG [124], and combined with sample reuse, surrogate modeling, and stochastic optimization method to create a computationally feasible framework for Bayesian OED with complex nonlinear systems [72]. Many advanced techniques have also been proposed to accelerate computation, improve the estimation accuracy, or tailor Bayesian OED to specific problems [143, 98, 150, 1, 123, 146, 114, 11, 82, 56, 51, 62, 120].

The OED problem becomes more challenging when nuisance (or auxiliary, ancillary) parameters are present—that is, additional parameters that carry uncertainty but not targeted for inference. Numerical techniques such as the double-nested MC can no longer be directly applied since the nuisance parameters need to be marginalized out, requiring yet another MC loop. To tackle this need, [51] introduces a layered multiple importance sampling technique with an additional MC marginalization, while [114] proposes to use a semi-implicit nested MC estimator to estimate the expected utility with nuisance parameters, and [56] presents a variational OED framework that can handle implicit likelihoods with nuisance parameters by learning a variational approximation for both the likelihood and marginal likelihood.

While much of the above work focused on OED criteria targeting the model parameters, in many scenarios prioritizing the uncertainty of a model's QoI prediction becomes even more crucial. For example, in engineering design, the ultimate goal may entail computing the maximum deformation of a structure under a load (the QoI), while reducing the uncertainty of the structure's material properties such as Young's modulus and yield stress (the model parameters) would only be intermediate steps needed toward computing the goal QoI. By adopting an OED criterion that reflects the information gain directly on those QoIs would constitute a goal-oriented OED (GOOED) formulation, which may lead to designs that differ significantly from their non-goal-oriented counterparts.

GOOED also brings additional computational challenges, since it needs to incorporate an additional parameter-to-QoI mapping. To address this issue, [28] proposes the Optimal Experimental Design for Prediction (OED4P) framework. This framework focuses on optimizing experimental

design for predictions based on push-forward models. It includes two types of problems: the inverse problem for updating the probability density function (PDF) of key input parameters based on observation data, and the forward problem for measuring EIG through another push-forward model. The OED4P framework updates the distribution of key model parameters through a new probability measure, which is similar to Bayes' theorem but uses an initial probability instead of evidence in the denominator. It then generates an updated distribution of model predictions through sampling methods and calculates the KL divergence between the updated and initial probabilities. This framework helps reduce the expense of traditional Bayesian inference by avoiding the calculation of evidence, but the framework departs from the Bayesian update of uncertainty and can be difficult to apply in practice. Approaches more closely following the Bayesian framework include [6] that focuses on linear mappings in order to allow analytical Gaussian posterior and posterior-predictive distributions. An efficient computational method by [155] further uses an offline-online decomposition and low-rank approximation to reduce the complexity of high dimensional QoIs, but remained with linear models. GOOED for nonlinear observation and prediction mappings remains an open and active area of research.

When multiple candidate models are available, OED for model discrimination aims to design experiments to effectively differentiate between multiple candidate models, rather than solely focusing on model parameters or predictions. Various utility functions for model discrimination have been proposed, including total separation that accounts for the difference between the posterior predictive means of candidate models [122, 102, 103], $T$-optimality criterion which maximizes the minimal deviation between a null model and an alternative [5], and mutual information between the model indicator and the observations [19, 17, 32, 46, 65] that is widely used.

In addition to the OED challenges for handling nonlinear models, nuisance parameters, goal-oriented QoIs, and multi-models each in its own, another key research gap is there does not exist a unified OED framework that can incorporate them simultaneously. Furthermore, all the OED methods mentioned above are designed for static (batch) experiments, and do not accommodate the adaptive design of a sequence of experiments. We seek to fill these gaps through the work of this thesis.

### 1.1.2 Sequential optimal experimental design

When multiple experiments can be performed sequentially, common OED strategies become suboptimal. *Batch* (static) design decides all experiments *a priori* and does not offer any opportunity to adapt to new observations (i.e., no feedback). *Greedy* (myopic) design [20, 44, 32, 136, 45, 46, 79, 64, 81] plans only for the *next* experiment and lacks consideration for future consequences (i.e., no lookahead). It is easy to relate, even from everyday experience (e.g.,

driving a car, planning an event), that a lack of feedback (adaptation) and lookahead (foresight) can lead to suboptimal decision-making.

A provably optimal formulation of sequential experimental design—we call it the sequential OED (sOED) [109, 147, 71, 74]—includes both elements of feedback and lookahead. As we will show in this thesis, sOED generalizes both the batch and greedy designs. The main features of sOED are twofold. First, sOED works with design *policies* (i.e., functions that adaptively suggest what experiment to perform depending on what has transpired). Second, sOED designs for *all remaining experiments*, therefore it captures the effect of each design decision on the entire design horizon.

Following [71, 74], sOED can be formally and mathematically formulated using a state-space representation, specifically via a partially observable Markov decision process (POMDP). In this approach, a belief state is formed based on the Bayesian posterior describing the uncertainty of a hidden state (i.e., of the unknown model parameter), thereby turning the POMDP into a belief Markov decision process (MDP) [95]. The formalization through state-space modeling reveals the essence of sequential design: its ability to adapt. Adaptation must be done in response to *something*, and this "something" is what defines the state. The intersection of sequential design and state-space modeling is largely missing in the current OED literature .

The POMDP emerging from sOED is atypical and challenging: finite horizon, continuous distributions, infinite state space, continuous designs and observations, sampling-only transitions where each is a Bayesian inference, and information measures as rewards. Off-the-shelf POMDP algorithms (e.g., [31, 94, 30, 87, 75]) are not directly suitable to accommodate this problem. Solution attempts for sOED have also been sparse, for example [29, 60, 119, 24, 36, 111, 149] largely limit to discrete settings or do not use a Bayesian framework with information criteria. More recent efforts for Bayesian sOED [71, 74] employ approximate dynamic programming (ADP) and transport maps but remain computationally expensive. Elsewhere, [55] introduces the Deep Adaptive Design (DAD) that efficiently amortizes the inference cost by learning a policy network that instantaneously returns the next design given previous designs and observations, thereby greatly accelerating the online deployment speed. A variant, the Implicit DAD (iDAD) [76], is further furnished with the ability to accommodate implicit likelihoods. Since both DAD and iDAD use the forward model (i.e. parameter-to-observable map) derivative, [14] proposed to learn the policy using RL without this requirement. However, both this RL algorithm and DAD employ a nested Monte Carlo (MC) EIG lower bound that scales with $O(n^2)$ forward model evaluations, which remains costly since *each* forward model run may entail a partial differential equation (PDE) solve in many engineering and science settings [154]. Moreover, these advanced sequential design methods solely focus on criteria targeting the model parameters, while there is a noticeable lack of research on sOED tailored for tasks such as model discrimination and goal-oriented QoIs, particularly when nuisance parameters

are involved.

### 1.1.3 Robust optimal experimental design

Conventional Bayesian OED studies primarily focus on maximizing the *expected* utility (e.g., the *expected* information gain) of experiments, acknowledging the inherent randomness resulting from the stochastic nature of experimental observations that cannot be predetermined. As a result, it is necessary to consider all possible scenarios by taking the expectation over observations. However, these studies typically ignore the risk or spread associated with the utility itself. While a design that maximizes the expected utility is optimal in expectation, it does not guarantee a high utility outcome once the experiment is conducted and the data is collected. Figure 1.1 illustrates this point by showing the histograms of utility for two different designs. Design #1 has a slightly higher expected utility of 5 compared to design #2 which stands at 4.8, making design #1 superior under the criterion of maximizing the expected utility. However, the utility of design #1 exhibits a significantly larger variance. This implies that there is a non-small probability of obtaining a utility lower than 3 if an experiment is conducted under this design. On the other hand, design #2 offers much higher certainty of the utility, with very small probability for the utility to be lower than 4. In this case, despite design #1 having a higher utility in expectation, design #2 can be reasonably considered as a better design due to its more stable outcome. This aligns with the principle of loss or risk aversion, especially when considering potential experimental costs. The possibility of a low utility outcome with design #1 may result in a much larger loss, taking into account the expense of the experiment.

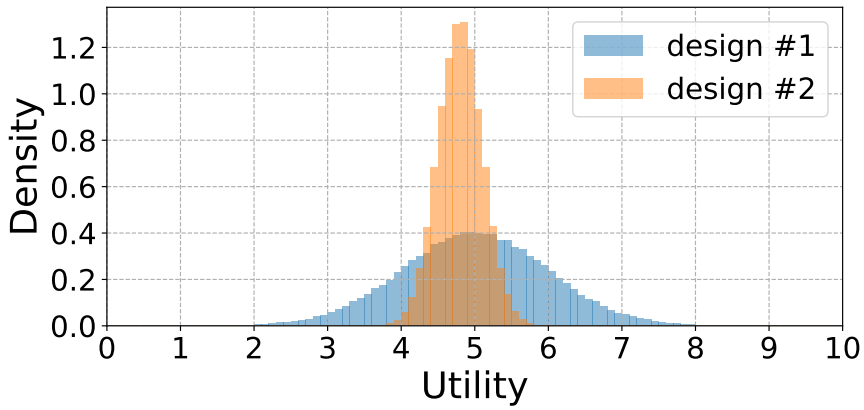



Figure 1.1: Utility histograms of two example designs.

The example above highlights the need for a more robust approach to experimental design, which we refer to as *robust OED* (rOED). rOED aims to find a design that not only has a high expected utility, but also a stable outcome. There is already existing literature on robust design, such as

Taguchi methods in product quality control, which addresses variation in product performance due to uncontrollable environmental factors by varying them along with the controllable design factors [140, 49, 86, 10, 25, 22]. Additionally, a variance-penalized criterion has been proposed for response adaptive design of clinical trials to evaluate performance based on both expected total responses and the variance of responses [156, 89]. In the experimental design field, clustering local optimal designs have been used to construct a robust experimental design for multivariate generalized linear models [43], tolerating model violations by augmenting the optimal design points with space-filling points [85], and a worst-case formulation with a min-max (or max-min) optimization [148, 84, 137]. [67] further compares the worst-case design with Bayesian OED, and finds that Bayesian OED is already robust to some extent compared with non-Bayesian OED, as it marginalizes out the uncertainty by taking the expectation over the prior. More research has been done to make the Bayesian OED more robust against prior misspecifications by using classes of priors [13, 12, 40, 145]. Nevertheless, there remains a lack of investigation on identifying and minimizing the variability of the experimental utility within the Bayesian OED framework.

### 1.1.4 Robust sequential optimal experimental design

The idea of robustness can be further combined with sequential experiments to form the robust sequential OED (rsOED). The objective of rsOED is to enhance the robustness of sOED by effectively controlling the variance of the rewards/utilities associated with the experimental outcomes.

There is no existing work specifically for rsOED, however, significant progress has been made in the field of reinforcement learning to improve the robustness of agent behaviors. For instance, exponential utilities are used to model risk-averse behaviors [70, 16, 9, 112], value-at-risk (VaR) and conditional-value-at-risk (CVaR) are also utilized to improve the robustness of policy by focusing on eliminating bad rewards [48, 121, 35, 141, 157], and reinforcement learning algorithms are also developed for the worst-case criterion [68, 104].

In addition to these risk-sensitive methods, variance-based methods are also widely adopted due to their high interpretability, and mean-minus-variance is commonly used due to its simplicity [52, 135, 151, 101]. In order to estimate the variance, an indirect method is proposed using the first and second moment of the rewards [135], and has been widely studied [152, 142, 88, 118]. For example, [142, 88] developed an actor-critic method for mean-minus-variance optimization, and [152] extends the usage of the indirect estimator to $\lambda$-returns. Meanwhile, a direct estimator of the variance is also proposed [131], and [77] proposes a variance-penalized on-policy and off-policy actor-critic method based on the direct estimator, and provides the policy gradient for maximizing the variance-penalized objective, however, their work is limited to stochastic policy, and there is a lack of variance-penalized reinforcement learning algorithms specifically designed for deterministic

policies and continuous action spaces.

## 1.2 Objectives and outline

Despite significant recent advances and wide-ranging applications in the field of OED, the research on sOED is relatively limited, particularly in leveraging reinforcement learning techniques. In addition, existing sOED research is computationally intensive in terms of forward model evaluations and only concentrates on the inference of model parameters as the objective of experimental design. Moreover, there is a significant research void in the area of controlling the utility variance within the Bayesian OED framework. This gap exists not only in the context of batch (non-sequential) design but also in sequential designs.

We propose to tackle these research challenges in this thesis via two main avenues. First, we aim to develop computationally efficient sOED methods for a range of OED problems targeting parameter inference, model discrimination and goal-oriented prediction, even in the presence of nuisance parameters. Second, we want to develop numerical techniques that enhance the robustness of both OED and sOED by effectively controlling the variance of utilities.

More specifically, the main objectives of this thesis can be summarized as follows.

- To develop computationally efficient methods for solving sOED problems using techniques from reinforcement learning. Specifically, it is achieved via the following sub-objectives:
  - To derive the policy gradient expressions for finite-horizon sOED to enable gradient-based optimization.
  - To leverage the expressive capabilities of deep neural networks to approximate the policy and value function, as well as to serve as surrogate models for expensive forward models.
  - To employ advanced reinforcement learning techniques, such as replay buffer and the target network.
- To rigorously formulate the variational sequential optimal experimental design (vsOED) framework, which uses the variational approximation to the Bayesian posteriors to form a lower bound estimator for expected utility, avoiding the need for computationally intensive information gain calculations.
- To extend vsOED to handle various OED problems including parameter inference, model discrimination, and goal-oriented prediction, even when nuisance parameters are present.

- To formulate the robust optimal experimental design (rOED) framework, which enhances the robustness of batch (non-sequential) design by introducing a penalty on the variance of utilities to the objective function. This involves developing numerical techniques for estimating the variance-penalized objective function, and analyzing the bias and variance of the estimator.

- To formulate the robust sequential optimal experimental design (rsOED) framework, combining the principles of sOED and rOED. This includes deriving the policy gradient expressions for the variance of the total rewards for rsOED, and utilizing variational posterior approximation to accelerate the computation.

- To validate sOED, vsOED, rOED and rsOED using numerical examples, including those involving computationally-intensive PDE-based models.

The dissertation is organized as follows. In Chapter 2, we present a thorough formulation for sOED and its problem statement, detail the PG-sOED algorithm and its numerical setup and demonstrate PG-sOED on a number of numerical examples. In Chapter 3, we introduce vsOED within a unified framework, offer numerical algorithms to solve vsOED problems and demonstrate its efficiency over baseline methods through various illustrative examples. In Chapter 4, we introduce the formulation of the variance-penalized rOED, propose a double-nested MC estimator to estimate the variance-penalized criterion, and provide the convergence order of this estimator. We present numerical examples to validate the convergence of the proposed estimator and show the value of rOED in real physical problems. In Chapter 5, we present the problem statement of rsOED, provide the policy gradient expressions and the corresponding MC estimator, and validate rsOED on a numerical example. The last chapter, Chapter 6, provides concluding remarks and discussions for future work.

# CHAPTER 2

# Sequential Optimal Experimental Design

**Sequential optimal experimental design (sOED)** involves the optimal design of a sequence of experiments by leveraging newly acquired information (i.e., *adaptation* or *feedback*) and anticipating future effects (i.e., *lookahead*). In this chapter, we present a general mathematical formulation to sOED featuring a state-space representation under the belief Markov decision process (MDP) framework. We prove the sOED's optimality over batch and greedy designs, and illuminate the inherent higher computational cost of greedy design compared to sOED by contrasting their information gain reward structures. We then introduce new, computationally efficient methods to solve the sOED problem using actor-critic techniques from reinforcement learning (RL). Specifically, we will derive the policy gradient (PG) formulas for sOED to enable gradient-based optimization, and employ deep neural network (DNNs) to achieve expressive parameterization of the policy functions. We call this new method the PG-sOED algorithm. We validate PG-sOED on a benchmark example and demonstrate its advantages over other design baselines (e.g., batch and greedy designs) via a sensor movement problem for contaminant source inversion in a convection-diffusion field. Notably, we provide explanations for the resulting policy behaviors using knowledge about the underlying physical process.

In this chapter, we only focus on sOED for model parameter inference, the extension to other design objectives (e.g., model discrimination, goal-oriented prediction) will be introduced in Chapter 3.

The content of this chapter corresponds to the author's publication [130], and the code is available at: https://github.com/wgshen/sOED.

## 2.1 Problem formulation

### 2.1.1 Background

Consider designing a finite[1] number of $N$ experiments indexed by $k = 0, 1, \ldots, N-1$. The integer $k$ then represents the number of experiments completed thus far (e.g., $k = 0$ refers to the first experiment before any has been conducted, and $k = N-1$ refers to the last experiment where $N-1$ has been previously completed). While the decision of how many experiments to perform (i.e., choice of $N$) is important, it is not considered in this thesis; instead, we assume $N$ is always given and fixed. Let $\theta \in \mathbb{R}^{N_\theta}$ denote the vector of uncertain model parameters we seek to learn from the experiments, $d_k \in \mathcal{D}_k \subseteq \mathbb{R}^{N_d}$ the design vector for the $k$th experiment, (e.g., experiment conditions), $y_k \in \mathbb{R}^{N_y}$ the observation vector from the $k$th experiment, (i.e., experiment measurements), and $N_\theta$, $N_d$, and $N_y$ respectively the dimensions of parameter, design, and observation spaces. While the notation above suggests continuous-valued $\theta$, $d_k$, and $y_k$, discrete or mixed settings can be accommodated as well. For simplicity, we let $N_d$ and $N_y$ be constant across all experiments, but this is not required.

A Bayesian approach treats $\theta$ as a random vector. After performing the $k$th experiment, the conditional probability density function (PDF) for $\theta$ is updated via Bayes' rule:

$$p(\theta|d_k, y_k, I_k) = \frac{p(y_k|\theta, d_k, I_k)p(\theta|I_k)}{p(y_k|d_k, I_k)} \tag{2.1}$$

where $I_k = [d_0, y_0, \ldots, d_{k-1}, y_{k-1}]$ (and $I_0 = \emptyset$) is the information sequence collecting the design and observation records from all experiments before the $k$th experiment; $p(\theta|I_k)$ is the prior PDF (prior to the $k$th experiment), $p(y_k|\theta, d_k, I_k)$ is the likelihood, $p(y_k|d_k, I_k)$ is the marginal likelihood (or model evidence), and $p(\theta|d_k, y_k, I_k)$ is the posterior PDF. The prior depicts the state of uncertainty about $\theta$ before the $k$th experiment, and the posterior represents the updated state of uncertainty after having observed the outcome from the $k$th experiment. Equation (2.1) also simplifies $p(\theta|d_k, I_k) = p(\theta|I_k)$ since the prior should not be affected by the upcoming design.

The likelihood describes the observable $y_k$ through a forward model $G_k$ that governs the underlying process for the $k$th experiment (e.g., via solving a system of partial differential equations (PDEs)). For example, a popular likelihood emerges from the observation model

$$y_k = G_k(\theta, d_k; I_k) + \epsilon_k, \tag{2.2}$$

where $\epsilon_k$ is an additive noise (e.g., measurement noise). The inclusion of $I_k$ in $G_k$ signifies

[1] In experimental design, the experiments are generally expensive and limited in number. Finite and small values of $N$ are therefore of primary interest. This is in contrast to RL that often deals with infinite horizon.

that model behavior may be affected by previous experiments. Each evaluation of the likelihood $p(y_k|\theta, d_k, I_k) = p_\epsilon(y_k - G_k(\theta, d_k; I_k))$ thus involves a forward model solve (e.g., the PDEs), typically the most expensive parts of the overall computation.

Lastly, the posterior after the $k$th experiment $p(\theta|d_k, y_k, I_k) = p(\theta|I_{k+1})$ becomes the prior for the $(k+1)$th experiment and can be similarly inserted back into Eqn. (2.1). Hence, Bayes' rule can be consistently and recursively applied for a sequence of multiple experiments, and it has long been demonstrated as an extended logic for expressing and updating uncertainty as new evidence becomes available [38].

### 2.1.2 Sequential optimal experimental design formulation

We now present the general formulation for sOED, posed as an MDP. An MDP is defined by a 4-tuple: {state space ($\mathcal{X}$), design (action) space ($\mathcal{D}$), transition dynamics ($\mathcal{F}$), and reward function ($g$)}. A policy ($\pi$) further maps from state to action, thus it determines the rule for taking a design (action) when at a particular state. All these entities are introduced in detail below. An overview flowchart describing sOED is presented in Fig. 2.1 to accompany the definitions below.

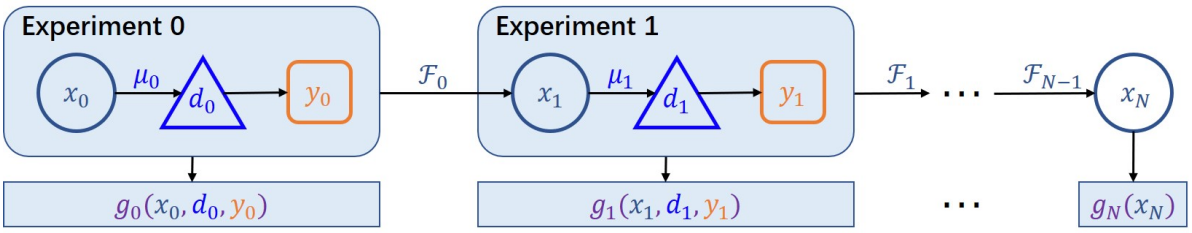



Figure 2.1: Flowchart of the process involved in a $N$-experiment sOED.

**State.** The state vector $x_k = [x_{k,b}, x_{k,p}] \in \mathcal{X}_k$ captures the state of the system and environment before conducting the $k$th experiment. It encompasses a belief state $x_{k,b}$ representing the state of uncertainty about $\theta$, and a physical state $x_{k,p}$ carrying all other non-uncertain state information needed for subsequent experiments. Since $\theta$ is not directly observable and can be only inferred from observations $y_k$ through Eqn. (2.1), this also corresponds to a POMDP for $\theta$ (or a belief-MDP on $x_k$ since the belief and physical states in $x_k$ are fully observable).

Conceptually, a *realization of the belief state* manifests as a posterior random variable $(x_{k,b} = x'_{k,b}) = (\theta|I_k = I'_k)$. To represent such a random variable, one may use, for example, its PDF, cumulative distribution function, or characteristic function, but these all require some functional approximation in practice. Alternatively, one can track $I_k$ directly to capture $x_{k,b}$ without any approximation and without needing to perform Bayesian inference explicitly since $I_k$ is the trivial

sufficient statistic for the posterior[2,3]. However, the dimension of $I_k$ grows with $k$, but is always upper-bounded in the finite-experiment case here. Regardless of representation, the belief state space is uncountably infinite (i.e., the possible posteriors that can be realized is uncountably infinite), and hence it is not a discrete or finite-state system. We will further describe our numerical techniques for the belief state in Sec. 2.2.2.1 and Sec. 2.2.2.3.

**Design (action) and policy.** Sequential experimental design is adaptive in nature. It involves building policies mapping from the state space to the design space, $\pi = \{\mu_k : \mathcal{X}_k \mapsto \mathcal{D}_k, k = 0, \ldots, N-1\}$, where the design for the $k$th experiment is determined by the state via $d_k = \mu_k(x_k)$. Thus, sOED is inherently adaptive and computes designs based on the current state which depends on the previous experiments and their outcomes. We focus on deterministic policies in this work.

**System dynamics (state transition).** The system dynamics $x_{k+1} = \mathcal{F}_k(x_k, d_k, y_k)$ describes the transition from state $x_k$ to state $x_{k+1}$ after performing the $k$th experiment under design $d_k$ and observing $y_k$. For the belief state, the prior $x_{k,b}$ transitions to the posterior $x_{k+1,b}$ via Bayes' rule in Eqn. (2.1). The physical state, if present, evolves according to the relevant physical process. We note that while the policy is deterministic, the transition from $x_k$ to $x_{k+1}$ is in fact stochastic since the observation $y_k$ is random.

**Utility (reward).** $g_k(x_k, d_k, y_k) \in \mathbb{R}$ denotes the immediate reward obtained from performing the $k$th experiment, and it may depend on the state, design, and observation values. Similarly, $g_N(x_N) \in \mathbb{R}$ denotes the terminal reward catching any other reward measure that can only be computed after all experiments are completed. We will provide specific examples of reward structure pertaining to information measures in Sec. 2.1.4.

**sOED problem statement.** The sOED problem seeks the design policy that solves the following optimization problem: from a given initial state $x_0$,

$$\begin{aligned} \pi^* = \underset{\pi=\{\mu_0,\ldots,\mu_{N-1}\}}{\arg\max} \quad & U(\pi) \\ \text{s.t.} \quad & d_k = \mu_k(x_k) \in \mathcal{D}_k, \\ & x_{k+1} = \mathcal{F}_k(x_k, d_k, y_k), \qquad \text{for} \quad k = 0, \ldots, N-1, \end{aligned} \tag{2.3}$$

[2] $I_k$ collects the complete history of experiments and their observations, therefore is a sufficient statistic for $x_k$ by definition. Hence, if $I_k$ is known, then the full state $x_k$ is equivalently represented. All of these are conditioned on a given initial $x_0$ (which includes the prior on $\theta$), but for simplicity we will omit this conditioning when writing the PDFs in this thesis, with the understanding that it is always implied.

[3] It is possible for $\theta|I_k$'s with different $I_k'$'s to have the same PDF (or distribution or characteristic function), for example simply by exchanging the experiments. Hence, the mappings from $I_k$ to these portrayals (PDF, distribution, characteristic functions) are non-injective. This may be problematic when considering transition probabilities of the belief state, but avoided if we keep to our root definition of belief state based on $I_k$, which remains unique.

where

$$U(\pi) = \mathbb{E}_{y_0,\ldots,y_{N-1}|\pi,x_0}\left[\sum_{k=0}^{N-1} g_k(x_k, d_k, y_k) + g_N(x_N)\right] \tag{2.4}$$

is the expected total utility functional. If $x_0$ is unknown or stochastic, another expectation can be taken over $x_0$.

There are several traits of the sOED problem that makes it uniquely challenging: finite horizon; unobservable $\theta$; uncountably infinite state space; continuous design and observation spaces; intractable and sample-only transitions; each transition requiring a Bayesian inference and a potentially expensive forward model evaluation; and information measures as rewards.

### 2.1.3 Generalization of suboptimal experimental design strategies

We illustrate that both batch and greedy designs are special cases of the expected utility in Eqn. (2.4). That is, sOED generalizes these design strategies.

Batch OED designs all $N$ experiments together prior to performing any of those experiments. Hence, it is non-adaptive by definition and cannot make use of new information acquired from any of the $N$ experiments to help adjust the design of other experiments. Mathematically, batch design seeks static design (instead of a policy) over the joint design space $\mathcal{D} := \mathcal{D}_0 \times \mathcal{D}_1 \times \cdots \times \mathcal{D}_{N-1}$:

$$(d_0^{\text{ba}}, \ldots, d_{N-1}^{\text{ba}}) = \underset{(d_0,\ldots,d_{N-1})\in\mathcal{D}}{\arg\max}\ \mathbb{E}_{y_0,\ldots,y_{N-1}|d_0,\ldots,d_{N-1},x_0}\left[\sum_{k=0}^{N-1} g_k(x_k, d_k, y_k) + g_N(x_N)\right], \tag{2.5}$$

subject to the system dynamics. In other words, the design $d_k$ is chosen independent of $x_k$ (for $k > 0$). The suboptimality of batch design becomes clear once realizing Eqn. (2.5) is equivalent to the sOED formulation in Eqn. (2.3) but restricting all $\mu_k$ to be only constant functions—that is, sOED corresponds to a constraint relaxation of the batch optimization problem. Therefore, $U(\pi^*) \geq U(\pi^{\text{ba}} = d^{\text{ba}})$.

Greedy design is a type of sequential experimental design and produces a policy. It optimizes only for the immediate reward at each experiment:

$$\begin{aligned}
\mu_k^{\text{gr}} &= \underset{\mu_k}{\arg\max}\, \mathbb{E}_{y_k|x_k,\mu_k(x_k)}\left[g_k(x_k, \mu_k(x_k), y_k)\right], \qquad k = 0, \ldots, N-2, \\
\mu_{N-1}^{\text{gr}} &= \underset{\mu_{N-1}}{\arg\max}\, \mathbb{E}_{y_{N-1}|x_{N-1},\mu_{N-1}(x_{N-1})}\left[g_{N-1}(x_{N-1}, \mu_{N-1}(x_{N-1}), y_{N-1}) + g_N(x_N)\right],
\end{aligned} \tag{2.6}$$

without needing to subject to the system dynamics since the policy functions $\mu_k^{\text{gr}}$ are decoupled. $U(\pi^*) \geq U(\pi^{\text{gr}})$ follows trivially.

### 2.1.4 Information measures as experimental design rewards

In this section, we formulate reward functions that measure the information gained from the sequence of experiments. The formulation also illuminates the inherent computational disadvantage of greedy design compared to sOED. Lindley's seminal paper [92] proposes to use the mutual information between the parameter and observation as the expected utility, and Ginebra [61] provides more general criteria for proper information measures for OED. In particular, mutual information is equal to the expected KL divergence from the prior to the posterior, quantifying the farness between these two distributions. A larger divergence corresponds to a greater degree of belief update—and hence information gain—resulting from the experiment and its observation.

Following Lindley, we demonstrate the use of KL divergence in two sensible sequential design reward structures. The first, call it the *terminal-information-gain (TIG) formulation*, involves clumping the information gain from all $N$ experiments in the terminal reward (without loss of generality, we omit non-information reward contributions):

$$g_k(x_k, d_k, y_k) = 0, \qquad k = 0, \dots, N-1 \tag{2.7}$$

$$\begin{aligned} g_N(x_N) &= D_{\mathrm{KL}}\left( p(\cdot|I_N) \,||\, p(\cdot|I_0) \right) \\ &= \int_\Theta p(\theta|I_N) \ln\left[\frac{p(\theta|I_N)}{p(\theta|I_0)}\right] d\theta. \end{aligned} \tag{2.8}$$

The second, call it the *incremental-information-gain (IIG) formulation*, uses incremental information gain from each experiment in their respective immediate rewards:

$$\begin{aligned} g_k(x_k, d_k, y_k) &= D_{\mathrm{KL}}\left(p(\cdot|I_{k+1}) \,||\, p(\cdot|I_k)\right) \\ &= \int_\Theta p(\theta|I_{k+1}) \ln\left[\frac{p(\theta|I_{k+1})}{p(\theta|I_k)}\right] d\theta, \qquad k = 0, \dots, N-1 \end{aligned} \tag{2.9}$$

$$g_N(x_N) = 0. \tag{2.10}$$

We denote $U_T(\pi)$ the sOED expected utility defined in Eqn. (2.4) subject to the constraints in Eqn. (2.3) for a given policy $\pi$ while using the TIG formulation in Eqn. (2.7) and (2.8), and $U_I(\pi)$ be the same except using the IIG formulation in Eqn. (2.9) and (2.10). It is important to note that in this chapter, both TIG and IIG formulations only focus on the information gain (or equivalently KL divergence) on parameter inference, and the extension to model discrimination and QoI prediction will be discussed in Chapter 3.

**Theorem 1** (Terminal-incremental equivalence in sOED for parameter inference)**.** $U_T(\pi) = U_I(\pi)$ *for any policy* $\pi$.

A proof is provided in Appendix A.1. As a result, the two reward formulations lead to the same

sOED problem.

Notably, greedy design can only be formed using the IIG formulation in Eqn. (2.9) and (2.10) (if using TIG formulation, greedy design would optimize the zero in Eqn. (2.7)). Consequently, greedy design has a major computational disadvantage: it must compute the posteriors and incremental KL divergence terms for all intermediate experiments in order to evaluate the rewards. In contrast, sOED may use the TIG formulation, which only requires computing the posterior and KL divergence once after the final experiment is complete. Moreover, together with Theorem 1, we have $U_T(\pi^*) = U_I(\pi^*) \geq U_I(\pi^{\text{gr}})$ (i.e., sOED achieves higher expected utility than greedy design regardless of whether sOED uses the TIG or IIG formulation).

## 2.2 Numerical methods for sOED

We approach the sOED problem by explicitly parameterizing the policy functions. We then derive gradient of the expected utility with respect to the policy parameters so to enable gradient-based optimization of the policy—this is known as the PG method [132, 91, 139, 78, 41, 125, 105, 126, 99, 96, 8]. A key benefit of explicit policy parameterization is that the policy can be optimized entirely offline, and only needs to be evaluated online without additional optimization iterations. This is in contrast to ADP-sOED approaches [71, 74] and greedy design where a new optimization must be performed online in order to identify the next experimental design, a much slower process. In the following, we first derive the exact PG expression in Sec. 2.2.1. We then present numerical methods in Sec. 2.2.2 to estimate this exact PG expression.

### 2.2.1 Derivation of the policy gradient

In the new PG-sOED method, each policy function $\mu_k$ is parameterized with parameters $w_k$ ($k = 0, \ldots, N-1$), and denoted by the shorthand form $\mu_{k,w_k}$. The overall policy $\pi$ is therefore parameterized by $w = \{w_k, \forall k\} \in \mathbb{R}^{N_w}$ and denoted by $\pi_w$, where $N_w$ is the dimension of the overall policy parameter vector. The sOED problem stated in Eqn. (2.3) and (2.4) then updates to:

$$
\begin{aligned}
w^* = \arg\max_{w} \quad & U(w) \\
\text{s.t.} \quad & d_k = \mu_{k,w_k}(x_k) \in \mathcal{D}_k, \\
& x_{k+1} = \mathcal{F}_k(x_k, d_k, y_k), \qquad \text{for} \quad k = 0, \ldots, N-1,
\end{aligned}
\tag{2.11}
$$

from a given initial state $x_0$, where

$$U(w) = \mathbb{E}_{y_0,\ldots,y_{N-1}|\pi_w,x_0}\left[\sum_{k=0}^{N-1} g_k(x_k, d_k, y_k) + g_N(x_N)\right]. \tag{2.12}$$

We now aim to derive the gradient $\nabla_w U(w)$ in order to leverage gradient-based optimization to solve the sOED problem.

Before presenting the gradient expression, we first introduce the *action-value function* (or *Q-function*). The Q-function following policy $\pi_w$ and at the $k$th experiment is

$$Q_k^{\pi_w}(x_k, d_k) = \mathbb{E}_{y_k,\ldots,y_{N-1}|\pi_w,x_k,d_k}\left[g_k(x_k, d_k, y_k) + \sum_{t=k+1}^{N-1} g_t(x_t, \mu_{t,w_t}(x_t), y_t) + g_N(x_N)\right] \tag{2.13}$$

$$= \mathbb{E}_{y_k|x_k,d_k}\left[g_k(x_k, d_k, y_k) + Q_{k+1}^{\pi_w}(x_{k+1}, \mu_{k+1,w_{k+1}}(x_{k+1}))\right] \tag{2.14}$$

$$Q_N^{\pi_w}(x_N, \cdot) = g_N(x_N). \tag{2.15}$$

for $k = 0, \ldots, N-1$, where $x_{k+1} = \mathcal{F}_k(x_k, d_k, y_k)$. The Q-function is the expected cumulative remaining reward for performing the $k$th experiment at the given design $d_k$ from a given state $x_k$ and thereafter following policy $\pi_w$.

**Theorem 2** (Policy gradient)**.** *The gradient of the expected utility in Eqn.* (2.12) *with respect to the policy parameters (i.e., the policy gradient) is*

$$\nabla_w U(w) = \sum_{k=0}^{N-1} \mathbb{E}_{x_k|\pi_w,x_0}\left[\nabla_w \mu_{k,w_k}(x_k) \nabla_{d_k} Q_k^{\pi_w}(x_k, d_k)\Big|_{d_k=\mu_{k,w_k}(x_k)}\right]. \tag{2.16}$$

We provide a proof in Appendix A.2, which follows the proof strategy for an infinite-horizon MDP given by [132] .

### 2.2.2 Numerical estimation of the policy gradient

The PG in Eqn. (2.16) cannot be evaluated in closed form and needs be approximated numerically. We propose a Monte Carlo (MC) estimator:

$$\nabla_w U(w) \approx \frac{1}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1} \nabla_w \mu_{k,w_k}(x_k^{(i)}) \nabla_{d_k^{(i)}} Q_k^{\pi_w}(x_k^{(i)}, d_k^{(i)})\Big|_{d_k^{(i)}=\mu_{k,w_k}(x_k^{(i)})} \tag{2.17}$$

where superscript indicates the $i$th episode (i.e., trajectory instance) generated from MC sampling. Note that the *sampling* only requires a given policy and does not need any Q-function. Specifically, for the $i$th episode, we first sample a hypothetical "true" $\theta^{(i)}$ from the prior belief state $x_{0,b}$ and

freeze it for the remainder of this episode—that is, all subsequent $y_k^{(i)}$ will be generated from this $\theta^{(i)}$. We then compute $d_k^{(i)}$ from the current policy $\pi_w$, sample $y_k^{(i)}$ from the likelihood $p(y_k|\theta^{(i)}, d_k^{(i)}, I_k^{(i)})$, for all experiments $k = 0, \ldots, N-1$. The same procedure is then repeated for all episodes $i = 1, \ldots, M$. The episode sample size $M$ can be selected based on indicators such as the MC standard error. While we propose to employ a fixed $\theta^{(i)}$ for the entire $i$th episode, one may also choose to resample $\theta_k^{(i)}$ at each stage $k$ from the updated posterior belief state $x_{k,b}^{(i)}$. These two approaches are mathematically equivalent (see Appendix A.3), but the former is computationally much easier since it does not require working with any intermediate posteriors. Once the gradient estimate is available, policy network optimization can be performed using gradient-based optimization methods such as stochastic gradient ascent and Adam [80].

From Eqn. (2.17), the MC estimator for PG entails computing the gradients $\nabla_w \mu_{k,w_k}(x_k^{(i)})$ and $\nabla_{d_k^{(i)}} Q_k^{\pi_w}(x_k^{(i)}, d_k^{(i)})$. While the former can be obtained through the parameterized policy functions, the latter requires parameterization of the Q-functions as well. We thus parameterize both the policy and Q-functions—this is known as an actor-critic method. Furthermore, we adopt the representation techniques from Deep Q-Network (DQN) [106] and DDPG [91], and use DNNs to represent the policy and Q-functions. We present these details next.

#### 2.2.2.1 Policy network

Conceptually, one needs to construct an individual DNN $\mu_{k,w_k}$ to approximate $\mu_k : \mathcal{X}_k \mapsto \mathcal{D}_k$ for each $k$. Instead, we combine them together into a single function $\mu_w(k, x_k)$, which then requires only a single DNN for the entire policy at the cost of a higher input dimension. Subsequently, the $\nabla_w \mu_{k,w_k}(x_k^{(i)}) = \nabla_w \mu_w(k, x_k^{(i)})$ term from Eqn. (2.17) can be obtained via DNN back-propagation. Below, we discuss the architecture design of such a DNN, with particular focus on its input layer.

For the first input component, i.e., the stage index $k$, instead of passing in the integer, we opt to use a zero-indexed one-hot encoding taking the form of a unit vector:

$$k \qquad \longrightarrow \qquad e_k = [0, \ldots, 0, \underbrace{1}_{k\text{th}}, 0, \ldots, 0]^T. \tag{2.18}$$

We choose one-hot encoding because the stage index is an ordered categorical variable rather than a quantitative variable (i.e., it has notion of ordering but no notion of metric). Furthermore, these unit vectors are always orthogonal, which we observe to offer good numerical performance to the policy network. The tradeoff is that the dimension of representing $k$ is increased from 1 to $N$.

For the second component, i.e., the state $x_k$ (including both $x_{k,b}$ and $x_{k,p}$), we represent it in a

nonparametric manner as suggested in Sec. 2.1.2:

$$x_k \qquad \longrightarrow \qquad I_k = (d_0, y_0, \ldots, d_{k-1}, y_{k-1}). \tag{2.19}$$

To accommodate states up to stage $(N-1)$ (i.e., $x_{N-1}$), we use a fixed total dimension of $(N-1)(N_d+N_y)$ for this representation, where for $k < (N-1)$ the entries of $\{d_l, y_l \,|\, l \geq k\}$ (experiments that have not happened yet) are padded with zeros (see Eqn. (2.20)). This representation method provides two major advantages: (a) representation of belief state without any numerical approximation, and (b) intermediate belief states (i.e., $x_{k,b}$ for $k < N$) do not need to be computed since the policy network can directly take input of $I_k$. As a result, only a single final Bayesian inference conditioned on all designs and observations needs be performed at the end of each episode (this is in contrast to greedy design that requires all intermediate Bayesian posteriors and incremental KL divergence terms to be computed). We note that without the presence of $e_k$, it would not be possible for the actor or the critic to distinguish between scenarios such as: whether at stage $k$, or at a later stage but the designs and observations from stage $k$ are zero (i.e. the padded values). Furthermore, even though the mapping from $I_k$ to $x_k$ is not injective, whether utilizing $I_k$ or $x_k$ as the state representation (i.e., inputting into the policy and the value functions) leads to the same optimal policy and maximum expected utility value (see Appendix A.4).

Putting together these two components, the overall input layer for the policy network $\mu_w(k, x_k)$ has the form

$$I_k^{actor} = [\underbrace{\ e_k\ }_{N}, \overbrace{\ d_0\ }^{N_d}, \ldots, d_{k-1}, \underbrace{0, \ldots, 0}_{N_d(N-1-k)}, \overbrace{\ y_0\ }^{N_y}, \ldots, y_{k-1}, \underbrace{0, \ldots, 0}_{N_y(N-1-k)}]^T, \tag{2.20}$$

where we also indicate the zero-paddings for entries corresponding to future experiments $l \geq k$. The overall input layer has a total dimension of $N + (N-1)(N_d + N_y)$.

The remainder of the policy network is relatively straightforward. The output layer is an $N_d$-dimensional vector representing $d_k$ and the network architecture can be chosen by the user. We have experimented with dense layers, and experience suggests 2–3 hidden layers often achieve good performance for our numerical cases. Further hyperparameter tuning may be performed but it is not pursued in this chapter.

Lastly, we emphasize that $\mu_w(k, x_k)$ is not trained in a supervised learning manner; instead, it is updated iteratively via PG en route to maximizing $U(w)$.

#### 2.2.2.2 Q-network

Under the actor-critic setup, we build DNNs $Q^{\pi_w}_{k,\nu_k}$ (parameterized by $\nu_k$) to approximate $Q^{\pi_w}_k : \mathcal{X}_k \times \mathcal{D}_k \mapsto \mathbb{R}$ for $k = 0, \ldots, N-1$. In a similar manner as the policy network, we combine $Q^{\pi_w}_{k,\nu_k}$ into a single function to form the Q-network $Q^{\pi_w}_\nu(k, x_k, d_k)$. Subsequently, the $\nabla_{d^{(i)}_k} Q^{\pi_w}_k(x^{(i)}_k, d^{(i)}_k)$ term from Eqn. (2.17) can be estimated by $\nabla_{d^{(i)}_k} Q^{\pi_w}_\nu(k, x^{(i)}_k, d^{(i)}_k)$, which can be obtained via DNN back-propagation. The input layer takes the same form as the policy network, except we augment extra entries for $d_k$ (i.e., $I^{critic}_k = [I^{actor}_k, d_k]$). The overall input dimension is $N + (N-1)(N_d + N_y) + N_d$. The network output is a scalar.

The Q-network is trained in a supervised learning manner from the MC episodes generated for Eqn. (2.17), by finding $\nu$ that minimizes the following loss function built based on Eqn. (2.14):

$$\mathcal{L}(\nu) = \frac{1}{M} \sum_{i=1}^{M} \sum_{k=0}^{N-1} \left[ Q^{\pi_w}_\nu(k, x^{(i)}_k, d^{(i)}_k) - \left( g_k(x^{(i)}_k, d^{(i)}_k, y^{(i)}_k) + Q^{\pi_w}_{k+1}(x^{(i)}_{k+1}, d^{(i)}_{k+1}) \right) \right]^2 \tag{2.21}$$

where $d^{(i)}_k = \mu_w(k, x^{(i)}_k)$ and $Q^{\pi_w}_N(x^{(i)}_N, \cdot) = g_N(x^{(i)}_N)$. It is worth noting that $Q^{\pi_w}_{k+1}(x^{(i)}_{k+1}, d^{(i)}_{k+1})$ does not depend on $\nu$, but in practice is often approximated by $Q^{\pi_w}_\nu(k+1, x^{(i)}_{k+1}, d^{(i)}_{k+1})$ (for $k = 0, \ldots, N-2$)[4]. When minimizing the loss, the gradient contribution with respect to $\nu$ from this term is therefore ignored. Additionally, while in this chapter we always use a fixed sample $\theta^{(i)}$ to generate the entire $i$th episode (see description following Eqn. (2.17)), we can show the resulting Q-network indeed converges to the true Q-function defined in Eqn. (2.14) (see Appendix A.5).

#### 2.2.2.3 Evaluation of Kullback-Leibler rewards

A final step needed to construct the Q-network (by minimizing Eqn. (2.21)) is to evaluate the immediate and terminal rewards $g_k$ and $g_N$. Having established the equivalence of TIG and IIG formulations in Sec. 2.1.4, we focus on the former in this chapter since it only requires the KL divergence in $g_N$ at the end of each episode. Adopting the state representation via $I_k$ (Sec. 2.2.2.1), we do not need to explicitly form the intermediate posteriors. Instead, we only require a single Bayesian inference to obtain $p(\theta|I_N)$ and use it to calculate the KL divergence in Eqn. (2.8).

In general, the posteriors will be non-standard distributions and the KL divergence must be approximated numerically. For small $N_\theta$ (e.g., $\leq 4$), we discretize the $\theta$-space on a grid and estimate its posterior PDF pointwise. However, higher $N_\theta$ would require more scalable techniques, such as Markov chain Monte Carlo (MCMC) coupled with kernel density estimation or likelihood-free ratio estimation [144], variational inference [15] and transport maps [71]. Estimating KL divergence

[4]The use of an approximate Q-value in the next (i.e., $k+1$) stage rather than expanding further with $g_{k+1}$, $g_{k+2}$, etc. makes this a *one-step lookahead* approximation. This is not to be confused with greedy/myopic design, which does not include any future value term.

for high dimensional $\theta$-space goes beyond the scope of this thesis, however in Chapter 3, a novel "one-point estimate" technique is proposed and utilized to circumvent the need for explicit KL divergence calculations.

#### 2.2.2.4 Exploration versus exploitation

The tradeoff between exploration and exploitation is an important consideration when optimizing the policy. Exploration searches under-explored regions while exploitation focuses on region deemed promising based on current knowledge. Insufficient exploration may strand the policy search in a local optimum and insufficient exploitation may lack convergence. A mixed strategy to balance the two is prudent [27, 90], for example through the commonly used epsilon-greedy technique [138].

We inject exploration to policy optimization by adding a perturbation to our deterministic policy *only* when generating the MC episodes (Eqn. (2.17)) during training, *not* during testing. Hence exploration is solely used to aid the training. The exploration design becomes:

$$d_k = \mu_k(x_k) + \epsilon_{\text{explore}} \tag{2.22}$$

where $\epsilon_{\text{explore}} \sim \mathcal{N}(0, \mathbb{I}_{N_d}\sigma^2_{\text{explore}})$. If perturbed outside $\mathcal{D}_k$, it can be moved back to the closest location inside the feasible region. The value of $\sigma_{\text{explore}}$ reflects the degree of exploration and should be selected based on the problem context. For example, a reasonable approach is to set a large $\sigma_{\text{explore}}$ early in the algorithm and reduce it gradually. More advanced techniques have been proposed to reach a better exploration, for instance, by adding noise to the policy network parameters instead of the design variable [116, 54]; however, these strategies are beyond the scope of our paper.

### 2.2.3 Pseudocode for the overall algorithm

We present the detailed algorithm for PG-sOED in Algorithm 1. We re-emphasize that the exploration perturbation is *only* used in generating the MC episodes on line 5, but not used anywhere else (e.g., when evaluating the policy). Furthermore, we point out that when using the TIG formulation (Eqn. (2.7) and (2.8)), the posterior is used solely in the terminal reward, while immediate rewards do not require any posterior or KL divergence calculations but may include other non-information-based contributions. Conversely, in the IIG formulation (Eqn. (2.9) and (2.10)), immediate rewards do incorporate intermediate posterior and KL divergence calculations. In the numerical demonstrations of this chapter, we only focus on the TIG formulation.

**Algorithm 1:** The PG-sOED algorithm.

Define all components in Sec. 2.1.2;
Set initial state $x_0$, policy updates $L$, MC sample size $M$, policy and Q-network architectures, learning rate $\alpha$ for policy update, exploration scale $\sigma_{\text{explore}}$;
Initialize policy and Q-network parameters $w$ and $\nu$;
**for** $l = 1, \ldots, L$ **do**
    Simulate $M$ episodes: sample $\theta \sim x_{0,b}$, and then for $k = 0, \ldots, N-1$ sample $d_k = \mu_w(k, x_k) + \epsilon_{\text{explore}}$ and $y_k \sim p(y_k|\theta, d_k, I_k)$;
    Store the full information vectors from all episodes $\{I_N^{(i)}\}_{i=1}^M$, from which the intermediate $\{I_1^{(i)}, I_2^{(i)}, \ldots, I_{N-1}^{(i)}\}$ can also be formed trivially;
    Compute and store immediate and terminal rewards for all episodes $\{g_k^{(i)}\}_{i=1}^M$, $k = 0, \ldots, N$;
    Update $\nu$ by minimizing the loss in Eqn. (2.21);
    Update $w$ by gradient ascent: e.g., $w = w + \alpha \nabla_w U(w)$ for stochastic gradient ascent, where $\nabla_w U(w)$ is estimated through Eqn. (2.17);
    (Optional) Reduce $\alpha$ and $\sigma_{\text{explore}}$;
**end for**
Return optimized policy $\pi_w$;

## 2.3 Numerical results and discussions

We present two groups of examples to demonstrate PG-sOED. The first is a linear-Gaussian problem (Sec. 2.3.1) that offers a closed form solution due to its conjugate prior. This problem serves as a benchmark to validate PG-sOED and illustrate its superior computational speed over an existing ADP-sOED baseline. The second entails a sensor movement problem for contaminant source inversion in a convection-diffusion field (Sec. 2.3.2). It is divided into four cases with increasing complexity, each with a different illustration purpose. The purpose of Case 1 is to highlight the difference between sOED and greedy design, while Case 2 additionally draws contrast against batch design. Case 3 further features a higher dimensional parameter space, and Case 4 demonstrates a much longer sequence of experiments. We explain the behavior of the resulting policies using knowledge about the underlying convection-diffusion physics.

### 2.3.1 Linear-Gaussian benchmark

We adopt the linear-Gaussian problem from [71, 74] as a benchmark for validating PG-sOED. The observation model takes the form

$$y_k = G(\theta, d_k) + \epsilon_k = \theta d_k + \epsilon_k, \tag{2.23}$$

where the forward model is linear in $\theta$ and $\epsilon_k \sim \mathcal{N}(0, 1^2)$. The benchmark designs for $N = 2$ experiments, with prior $\theta \sim \mathcal{N}(0, 3^2)$ and design constrained in $d_k \in [0.1, 3]$. The resulting conjugate form renders all subsequent posteriors to be analytically Gaussian, thus allowing the optimal policies to be computed in closed form. There is no physical state for this problem. The stage and terminal rewards are

$$g_k(x_k, d_k, y_k) = 0, \quad k = 0, 1 \tag{2.24}$$

$$g_N(x_N) = D_{\mathrm{KL}}\left(p(\,\cdot|I_N) \,||\, p(\cdot|I_0)\,\right) - 2\left(\ln \sigma_N^2 - \ln 2\right)^2 \tag{2.25}$$

where $\sigma_N^2$ represents the variance of the final belief state, and the additive penalty in $g_N$ is purposefully inserted to make the problem more challenging.

The rewards are calculated by discretizing the $\theta$ space onto a uniform grid with 50 nodes. It is worth noting that testing with 1000 nodes has been conducted and the results indicate that increasing the number of nodes from 50 to 1000 does not have a significant impact on the outcomes. We solve this sOED problem both by ADP-sOED [74] and PG-sOED. For PG-sOED, we set $L = 100$, $M = 1000$, $\alpha = 0.15$, and $\sigma_{\text{explore}} = 0.2$ (decrease by factor of 0.95 per policy update). Both the policy network and Q-network contain two hidden layers with ReLU activation, and each hidden layer has 80 nodes. The architectures of the policy network (actor) and the Q-network (critic) are presented in Table 2.1 and Table 2.2, where the *Linear mapping* in Table 2.1 maps the output value to be within the design bounds. While we observed even a low $M = 10$ yielded good performance, $M = 1000$ is used to further reduce MC error in the demonstration. Stochastic gradient ascent is utilized for both the optimization of actor and critic networks. Both ADP-sOED and PG-sOED are implemented using Python and executed within the same computational environment.

Table 2.1: Architecture of the actor.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | $I_k^{actor}$ | $N + (N-1)(N_d + N_y)$ | - |
| H1 | Dense | 80 | ReLU |
| H2 | Dense | 80 | ReLU |
| H4 | Dense | $N_d$ | Sigmoid |
| Output | Identity | $N_d$ | Linear mapping |

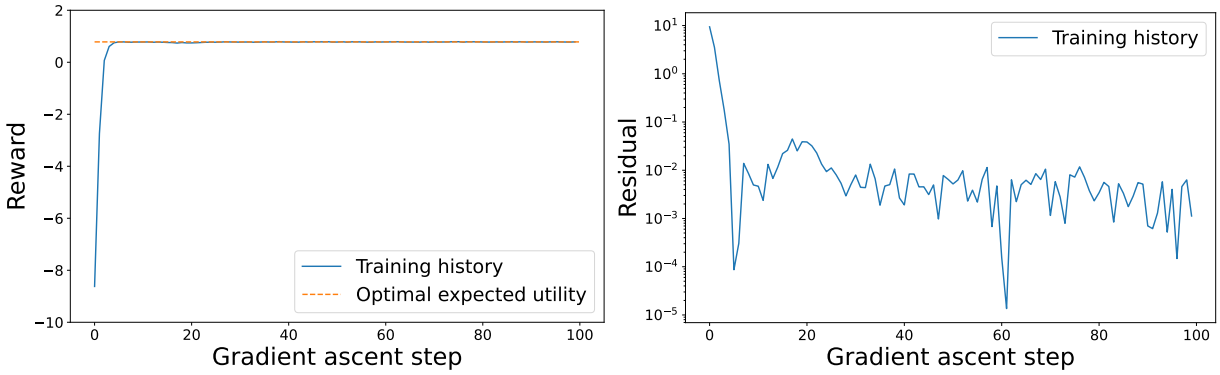
To assess the policies found by ADP-sOED and PG-sOED, we sample $10^4$ episodes using their final policies and compute their total rewards. ADP-sOED yields a mean total reward of $0.775 \pm 0.006$ and PG-sOED also $0.775 \pm 0.006$, where the $\pm$ is the MC standard error. Both match extremely well with the analytical result $U(\pi^*) \approx 0.783$ [71, 74] where the discrepancy of

Table 2.2: Architecture of the critic.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | $I_k^{critic}$ | $N + (N-1)(N_d + N_y) + N_d$ | - |
| H1 | Dense | 80 | ReLU |
| H2 | Dense | 80 | ReLU |
| Output | Dense | 1 | - |

PG-sOED is attributed primarily to the NN hyperparameters (e.g., NN architectures, learning rates, etc.). These results support that both ADP-sOED and PG-sOED have found the optimal policy.

Figures 2.2a and 2.2b present the convergence history for the expected utility and residual ($|U(\pi^*) - U(w)|$) as a function of the PG-sOED iterations. The convergence is rapid, reaching over 3 orders of magnitude residual reduction within 30 iterations. The much lower initial expected utility (around $-8.5$) also indicates that a random policy (from random initialization) performs much worse than the optimized policy.




(a) Expected reward history

(b) Residual history $|U(\pi^*) - U(\pi)|$

Figure 2.2: Convergence history of PG-sOED.

Table 2.3 compares the computational costs between ADP-sOED and PG-sOED obtained using a single 2.6 GHz CPU on a MacBook Pro laptop. The timing values reflect 30 gradient ascent updates for PG-sOED in the training stage, and 1 policy update (the minimum needed) for ADP-sOED. PG-sOED produces orders-of-magnitude speedups compared to ADP-sOED, especially the extremely low testing time (i.e., using the policy online during the experimental campaign after the policy has been constructed offline) achieving 0.0002 seconds per experiment (4 seconds per $10^4$ episodes with $N = 2$ experiments per episode). This drastic speedup is due to ADP-sOED being a value-based approach where each policy evaluation needs to solve a (stochastic) optimization problem, while PG-sOED only requires a single forward pass of its policy-network free of any

optimization or forward model evaluations. The fast online speed makes PG-sOED an excellent candidate for real-time design situations.

Table 2.3: Comparison of computational costs between ADP-sOED and PG-sOED.

| | **Training time (s)** | **Forward model evaluations** | **Testing time (s)** |
|---|---|---|---|
| ADP-sOED | 837 | $5.3 \times 10^8$ | 24,396 |
| PG-sOED | 24 | $3.1 \times 10^6$ | 4 |

Figure 2.3 depicts the difference of expected utility values obtained from the TIG formulation and IIG formulation, obtained analytically using the MATLAB symbolic mathematics and then evaluated numerically. The differences are all on the order of $10^{-15}$, which is near the numerical limit of double precision. This provides empirical validation to the equivalence between the expected utilities using the TIG formulation and the IIG formulation, which was stated and proven earlier in Theorem 1.

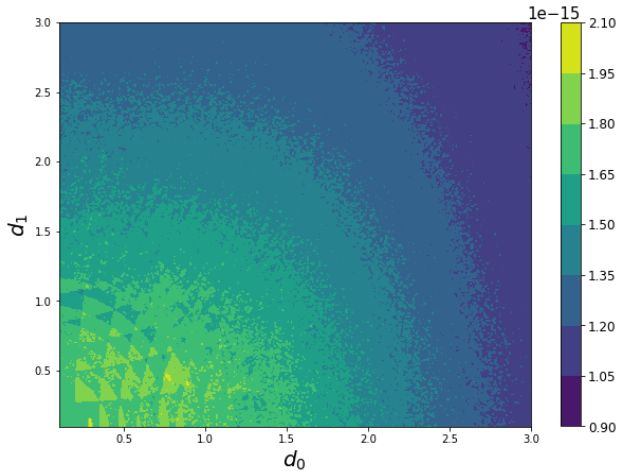



Figure 2.3: The difference of expected utilities using the TIG formulation and the IIG formulation.

## 2.3.2 Contaminant source inversion in a convection-diffusion field

### 2.3.2.1 Problem setup

The next group of demonstrations entails mobile sensor design in a convection-diffusion field (e.g., of a chemical contaminant plume). The contaminant concentration $G$ at time $t$ and location $z = [z_x, z_y]$ within a two-dimensional rectangular domain is governed by the convection-diffusion

PDE:

$$\frac{\partial G(z,t;\theta)}{\partial t} = \nabla^2 G - u(t)\cdot\nabla G + S(z,t;\theta), \qquad z \in [z_L, z_R]^2, \quad t > 0, \tag{2.26}$$

where $u = [u_x, u_y] \in \mathbb{R}^2$ is a time-dependent convective velocity, and $\theta = [\theta_x, \theta_y, \theta_h, \theta_s] \in \mathbb{R}^4$ is the source parameter residing within the source function

$$S(z,t;\theta) = \frac{\theta_s}{2\pi\theta_h^2}\exp\left(-\frac{(\theta_x - z_x)^2 + (\theta_y - z_y)^2}{2\theta_h^2}\right). \tag{2.27}$$

Here $\theta_x$ and $\theta_y$ denote the source location, and $\theta_h$ and $\theta_s$ denote the source width and source strength. The initial condition is $G(z,0;\theta) = 0$, and homogeneous Neumann boundary condition (i.e., zero-flux) is imposed for all sides of the domain. We solve the PDE numerically using second-order finite volume method on a uniform grid of size $\Delta z_x = \Delta z_y = 0.01$ and a second-order fractional step method for time-marching with stepsize $\Delta t = 5.0 \times 10^{-4}$. Figure 2.4 provides an example illustrating the evolution of solution $G$ over time.

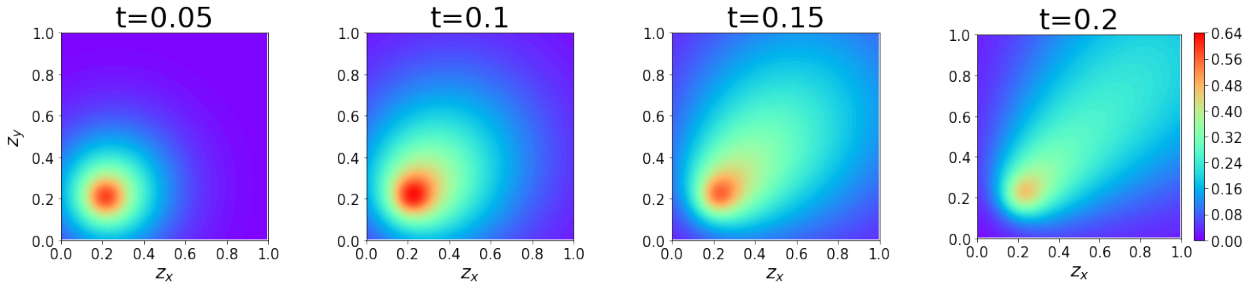



Figure 2.4: Sample numerical solution of the concentration field $G$ at different time snapshots. The solution is solved in a wider computational domain $[-1,2]^2$ but displayed here in $[0,1]^2$. In this case, $\theta = [0.210, 0.203, 0.05, 2]$ and the convection grows over time with $u_x = u_y = 10t/0.2$. Isotropic diffusion dominates early on and the plume stretches towards the convective direction over time.

For the design problem, we have a vehicle with sensing equipment for measuring the contaminant concentration $G$, and the vehicle can be relocated at fixed time intervals. We seek to determine where we should relocate this vehicle such that its measurements can lead to the best inference of the source parameters $\theta$. We consider $N$ measurement opportunities at times $t_k$ for $k = 0, \ldots, N-1$. The vehicle starts with initial belief state $x_{0,b} = (\theta|I_0)$ (i.e., prior on $\theta$) and initial physical state $x_{0,p}$ (i.e., initial vehicle location). The design variable is the displacement of the vehicle from its current location. The physical state is updated via

$$x_{k+1,p} = x_{k,p} + d_k. \tag{2.28}$$

At the new physical location, a noisy measurement of the contaminant concentration is obtained in the form

$$y_k = G(z = x_{k+1,p}, t_k; \theta) + \epsilon_k \left(1 + |G(x_{k+1,p}, t_k; \theta)|\right) \tag{2.29}$$

where $\epsilon_k \sim \mathcal{N}(0, \sigma_\epsilon^2)$, thus the observation noise is affected by the signal magnitude. Once the new measurement is acquired, the belief state is updated from $x_{k,b} = (\theta|I_k)$ to $x_{k+1,b} = (\theta|I_{k+1})$ through Bayes' rule. The reward functions are

$$g_k(x_k, d_k, y_k) = -c_q f_c(d_k), \quad k = 0, \ldots, N-1 \tag{2.30}$$

$$g_N(x_N) = D_{\mathrm{KL}}\left(p(\cdot|I_N) \,||\, p(\cdot|I_0)\right), \tag{2.31}$$

where $c_q$ is a parameter that reflects the relative weight of the movement cost. The immediate reward reflects a cost on the vehicle movement where $f_c(d_k)$ denotes the specific movement cost function that may also depend on the convection velocity.

We explore four cases for the convection-diffusion problem, with their detailed settings summarized in Table 2.4. The four cases involve different number of experiments and measurement times: Case 1 measures at $t_0 = 0.15$ and $t_1 = 0.32$; Case 2 at $t_0 = 0.05$ and $t_1 = 0.2$, Case 3 at $t_k = 0.05(k+1)$ $(k = 0, \ldots, 3)$, and Case 4 at $t_k = 0.012(k+1)$ $(k = 0, \ldots, 14)$. For PG-sOED, we set $L = 300$ for Case 1–3 and $L = 3000$ for Case 4. All cases use $M = 1000$, $\alpha = 0.01$ with the Adam optimizer, and $\sigma_{\text{explore}} = 0.05$. Performance evaluation of design policies is done using $10^4$ test episodes.

Table 2.4: Setup of the four cases for contaminant source inversion in a convection-diffusion field.

| | **Case 1** | **Case 2** | **Case 3** | **Case 4** |
|---|---|---|---|---|
| Number of experiments | $N = 2$ | | $N = 4$ | $N = 15$ |
| Prior of $\theta_x$ and $\theta_y$ | $\theta_x, \theta_y \sim \mathcal{U}([0,1])$ | | | |
| Prior of $\theta_h$ | $\theta_h = 0.05$ | | $\theta_h \sim \mathcal{U}([0.02, 0.1])$ | |
| Prior of $\theta_s$ | $\theta_s = \begin{cases} 0 \text{ if } t < 0.16 \\ 2 \text{ if } t \geq 0.16 \end{cases}$ | $\theta_s = 2$ | $\theta_s \sim \mathcal{U}([0,5])$ | |
| Initial physical state | $x_{0,p} = [0.5, 0.5]$ | | | |
| Design constraint | $d_k \in [-0.25, 0.25]^2$ | | $x_{k,p} \in [0,1]^2$ | |
| Velocity field | $u_x = u_y = 0$ | $u_x = u_y = 10t/0.2$ | | |
| Noise scale | $\sigma_\epsilon = 0.1$ | $\sigma_\epsilon = 0.05$ | | |
| Cost function $f_c(d_k)$ | $\|d_k\|^2$ | | $\|d_k\| - \frac{\sqrt{2}}{40} d_k \cdot u(t_k)$ | |
| Cost coefficient | $c_q = 0.5$ | $c_q = 0$ | $c_q = 0.2$ | $c_q = 0$ |

#### 2.3.2.2 Surrogate model

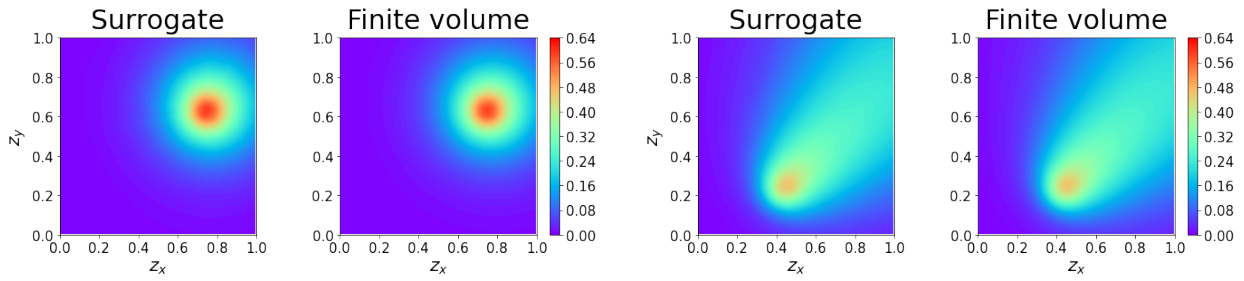
Solving the forward model Eqn. (2.26) using finite volume is still computationally viable for PG-sOED, but expensive. One strategy to accelerate the computation is to employ surrogate models to replace the original forward model. We use DNNs to construct surrogate models of $G(z, t_k; \theta)$ for $k = 0, \ldots, N-1$. Each DNN uses a 5-dimensional input layer taking $z$ and $\theta$ except for $\theta_s$ (note that $G$ is linearly proportional to $\theta_s$), five hidden layers with 40, 80, 40, 20, and 10 nodes, and a scalar output $G$. The architecture of the surrogate forward model is summarized in Table 2.5. A dataset is generated by solving for $G$ on 2000 samples of $\theta$ drawn from its prior distribution. These concentration values are then first restricted to only the domain that is reacheable by the vehicle (due to the design constraint), then shuffled across $\theta$ and split 80% for training and 20% for testing. We achieve test mean-squared-errors of around $10^{-6}$ for all surrogate models. Figure 2.5 provides an example comparing the concentration contours from $t = 0.05$ and $t = 0.2$ of Case 2 using the DNN surrogates (left column) and finite volume (right column), appearing nearly identical. More importantly, the surrogate models provide a significant speedup over the finite volume solver by a factor of $10^5$.

Table 2.5: Architecture of the surrogate forward model.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | $[z_x, z_y, \theta_x, \theta_y, \theta_h]$ | 5 | - |
| H1 | Dense | 40 | ReLU |
| H2 | Dense | 80 | ReLU |
| H3 | Dense | 40 | ReLU |
| H4 | Dense | 20 | ReLU |
| H5 | Dense | 10 | ReLU |
| Output | Dense | 1 | - |

#### 2.3.2.3 Case 1

Case 1 is diffusion-only, its purpose it to comparing PG-sOED with greedy design. We begin by offering a physical intuition about high-value design locations via Fig. 2.6 that plots the expected utility for a *single*-experiment design. The key insight is that high-value experiments are at the corners of the domain. This can be explained by the isotropic nature of diffusion process that carries information about distance but not direction, thereby leading to posterior distributions that resemble an arc of a circle (Fig. 2.7). Combined with the rectangular domain geometry and Neumann boundary conditions, the "covered area" of high-probability posterior is smallest (i.e.,




(a) $t = 0.05$ (b) $t = 0.2$

Figure 2.5: Comparison of the concentration field $G$ at $t = 0.05$ and $t = 0.2$ for Case 2 using the DNN surrogate (left column) and finite volume (right column). The surrogate solutions appear very accurate.

least uncertain), averaged over all possible $\theta$ source locations, when the measurements are taken at the corners.

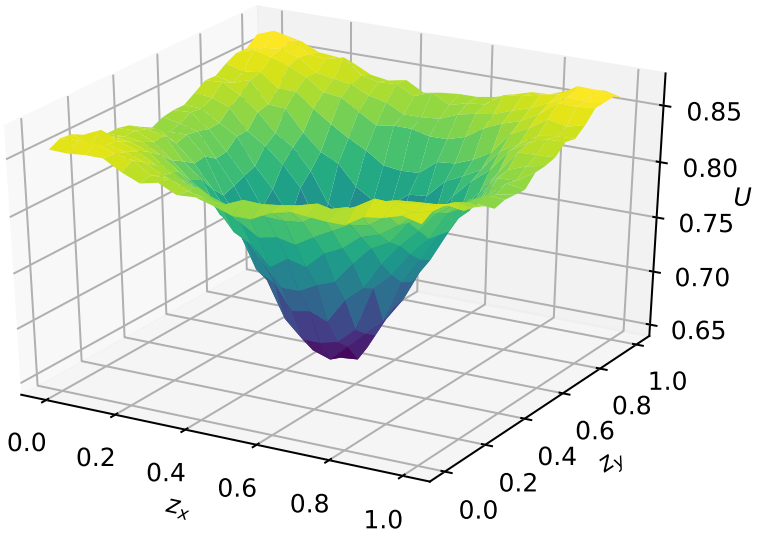



Figure 2.6: Case 1. Expected utility for one-experiment design at $t = 0.32$. The best design locations are at the corners.

With the insight that corners are good, understanding the behavior of PG-sOED becomes easier. Figure 2.8a shows the posterior contours after 1 and 2 experiments (i.e., $p(\theta|I_1)$ and $p(\theta|I_2)$) of an episode instance when using the PG-sOED policy; Fig. 2.8b displays those for the greedy design policy. In each plot, the purple star represents the true source location for that episode, the red dot represents the physical state (vehicle location), and the red line segment tracks the vehicle displacement (design) from the preceding location.

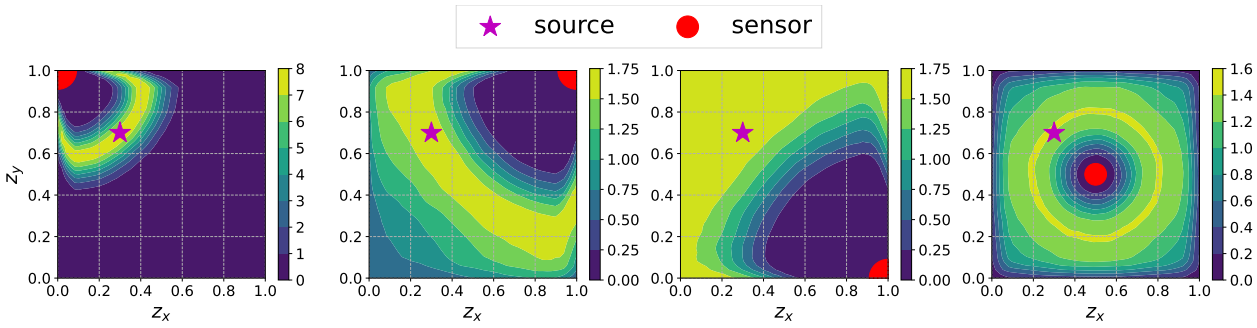
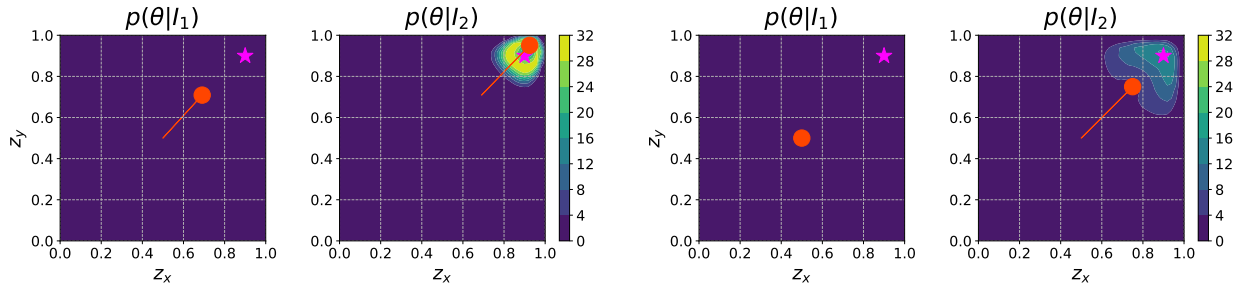



Figure 2.7: Case 1. Posterior PDF contours for the one-experiment design under different design locations (red dot) and a sample source location (purple star). The posteriors exhibit shapes resemble an arc of a circle, due to the isotropic nature of diffusion and the domain geometry.

In PG-sOED (Fig. 2.8a), the first design moves the vehicle towards a corner despite the source is off at $t_0$ and that no concentration is measured, incurring a negative reward $g_0 = -0.040$ due to the movement penalty. The greedy design realizes the source is off and remains at the initial location (center), keeping its reward at $g_0 = 0$. At this point, it would appear greedy design is performing better. The source then becomes active in the second experiment at $t_1$, and both PG-sOED and greedy shift the vehicle towards a corner. However, PG-sOED is able to arrive much closer to the corner and obtains a more informative measurement compared to greedy design, since PG-sOED has already made a head start in the first experiment. With a "sacrifice" of seemingly fruitless first experiment, PG-sOED is able to better position the vehicle for a much more lucrative second experiment, such that the expected *total* reward over the entire design horizon is maximized (total reward = 2.941 for PG-sOED versus total reward = 1.959 for greedy). We further generate $10^4$ test episodes under different samples of true $\theta$ and collect their realized total rewards in Fig. 2.9. The mean total reward for PG-sOED is $0.615 \pm 0.007$, higher than greedy design's $0.552 \pm 0.005$, where the $\pm$ is the MC standard error. The information-gain component of this mean total reward for PG-sOED is $0.712 \pm 0.007$, which is also higher than greedy design's $0.614 \pm 0.005$. This indicates PG-sOED's ability to find efficient tradeoffs of the movement cost in order to achieve a much higher information gain.

#### 2.3.2.4 Case 2

Case 2 incorporates convection in addition to diffusion, its aim is to compare PG-sOED with both greedy and batch designs. In Fig. 2.10, we plot the physical states $x_{1,p}$ and $x_{2,p}$ (i.e., vehicle locations after the first and second experiments) from $10^4$ episodes sampled from PG-sOED, greedy, and batch designs. We observe both PG-sOED and batch design initially move the vehicle towards the top-right (convective direction) and then turn back; greedy design roughly moves in the opposite direction. Notably, $d_1$ (design for the second experiment) for batch design is fixed regardless of the




(a) PG-sOED, **total reward** = **2.941**
(information gain = 3.038, movement cost = 0.097)

(b) Greedy, total reward = 1.959
(information gain = 2.022, movement cost = 0.063)

Figure 2.8: Case 1. An episode instance obtained by PG-sOED and greedy design. The purple star represents the true $\theta$, red dot represents the physical state (vehicle location), red line segment tracks the vehicle displacement (design) from the preceding location, and contours plot the posterior PDF.

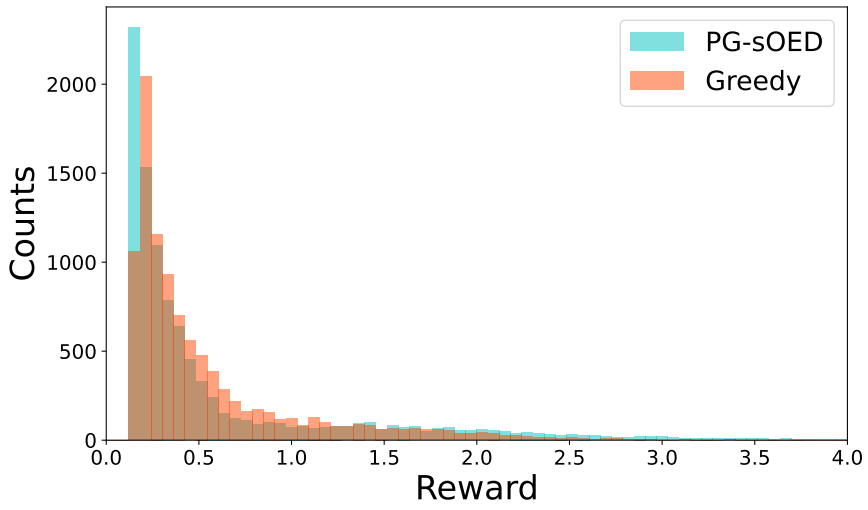



Figure 2.9: Case 1. Histograms of total rewards from $10^4$ test episodes from PG-sOED and greedy design. The mean total reward for PG-sOED is $0.615 \pm 0.007$, higher than greedy design's $0.552 \pm 0.005$.

outcome of the first experiment, which is in contrast to PG-sOED and greedy that are adaptive.

We explain the policy behaviors through Fig. 2.11, which plots the expected utility contours respectively for performing a *single*-experiment design at $t = t_0 = 0.05$ and $t = t_1 = 0.2$. For $t_0$ in Fig. 2.11a, the optimal design is around $(0.3, 0.3)$, which explains the initial movement of greedy design towards the bottom-left. For $t_1$, however, Fig. 2.11b reveals that the top-right region becomes more informative. Physically, this is a result of the convection velocity growing larger towards the top-right direction, and more information can be obtained if we "catch" the flow at a downstream position. This phenomenon explains why PG-sOED and batch design both move towards the top-right even in the first experiment, since both can plan for a more informative second experiment.

Returning to the two-experiment design, Fig. 2.12 summarizes the total rewards from $10^4$ test episodes with PG-sOED reaching the highest mean at $1.344 \pm 0.008$ followed by batch design's $1.264 \pm 0.007$ and greedy design's $1.178 \pm 0.010$. The advantage of PG-sOED is greater over greedy and less over batch, suggesting a more prominent role of lookahead than adaptation in this case. From the histograms, greedy design has many low-reward episodes, which correspond to scenarios when the true source location is in the upper-right (greedy design's first move is always to the bottom-left). At the same time, greedy design has a similar distribution of high-reward episodes as sOED because it is able to adapt. In contrast, batch design does not have many low-reward episodes since its first move is always to the upper-right. It also has fewer high-reward episodes due to its inability to adapt.

Lastly, we provide some examples of posteriors resulting from different episodes. Figure 2.13 presents scenarios where PG-sOED visibly achieves a narrower posterior compared to greedy and batch designs, which is also reflected quantitatively through the higher total reward. Meanwhile, there are also scenarios where PG-sOED achieves a lower total reward, such as those shown in Fig. 2.14. Since the true $\theta$ is not known when designing the experiments, PG-sOED thus optimizes the *expected* total reward (i.e., averaged) over all possible such scenarios.

#### 2.3.2.5 Cases 3 and 4

The last two cases, Cases 3 and 4, demonstrate the use of PG-sOED for a higher dimensional parameter space and longer sequence of experiments. Note that the movement cost for Case 3 has an additional term that penalizes when moving against the convective flow.

Case 3 additionally incorporates source strength $\theta_s$ and width $\theta_h$ as unknown parameters for a total of $N_\theta = 4$, and increases the number of experiments to $N = 4$. From Fig. 2.15, we see that PG-sOED's mean total reward ($3.435 \pm 0.016$) outperforms both greedy ($3.057 \pm 0.015$) and batch ($2.856 \pm 0.012$) designs. The information-gain component of the mean total reward for PG-sOED ($3.763 \pm 0.016$) is also noticeably better than greedy ($3.258 \pm 0.016$) and batch ($3.104 \pm 0.012$)

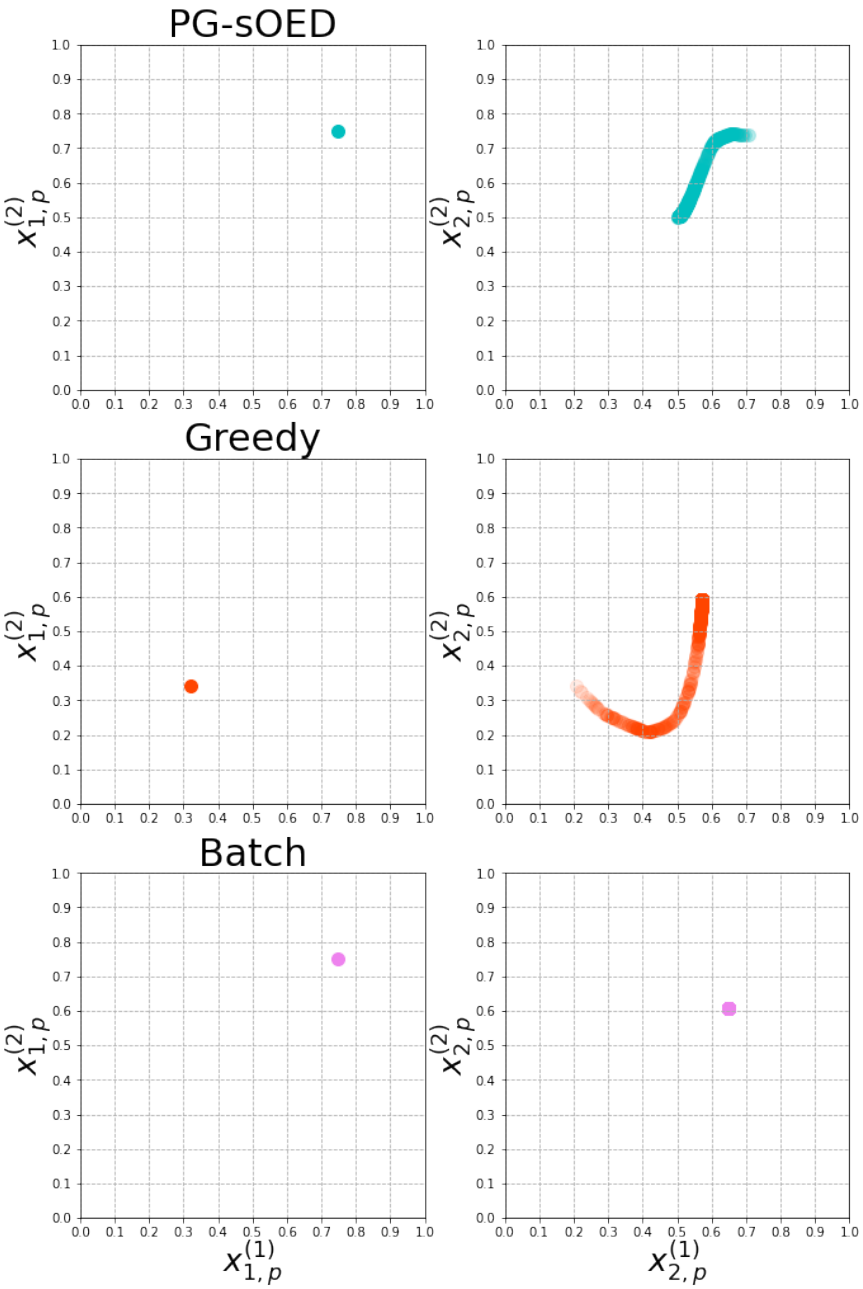
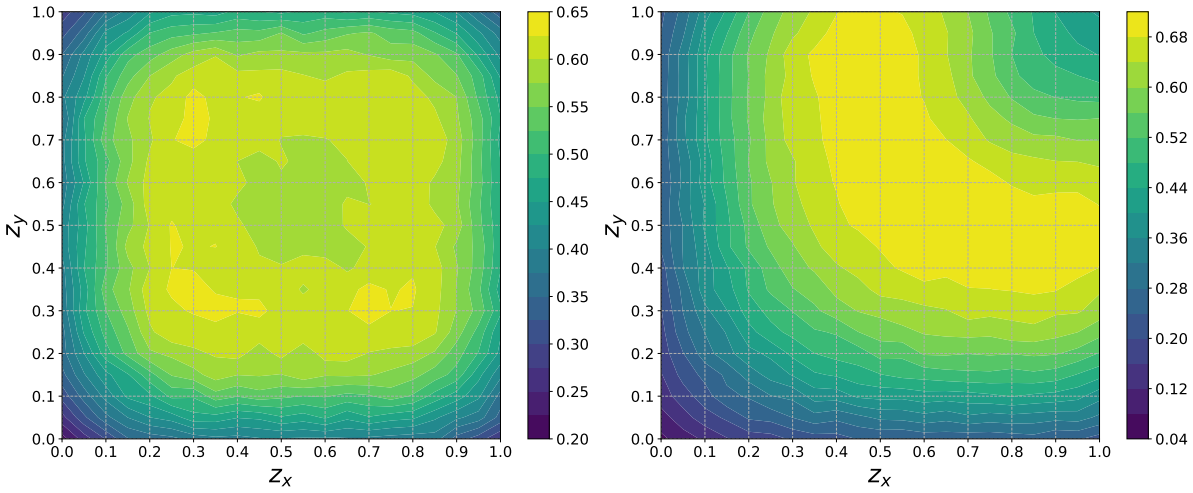
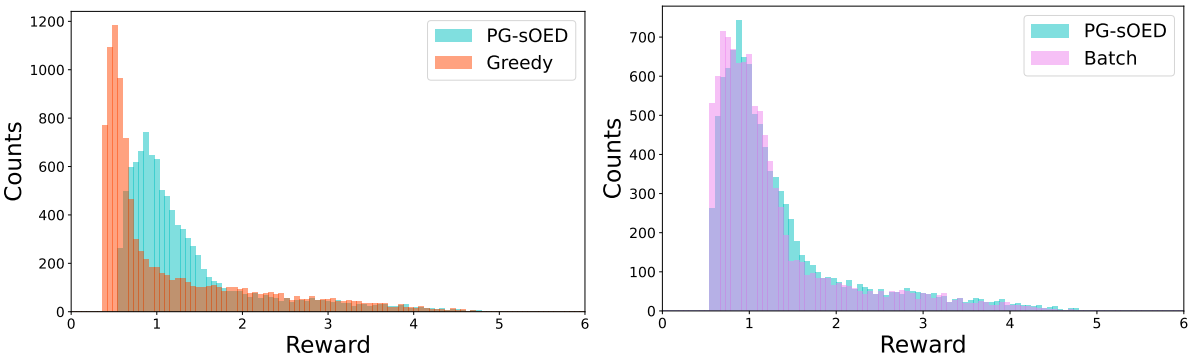



Figure 2.10: Case 2. Vehicle locations of episodes obtained from PG-sOED, greedy, and batch designs.



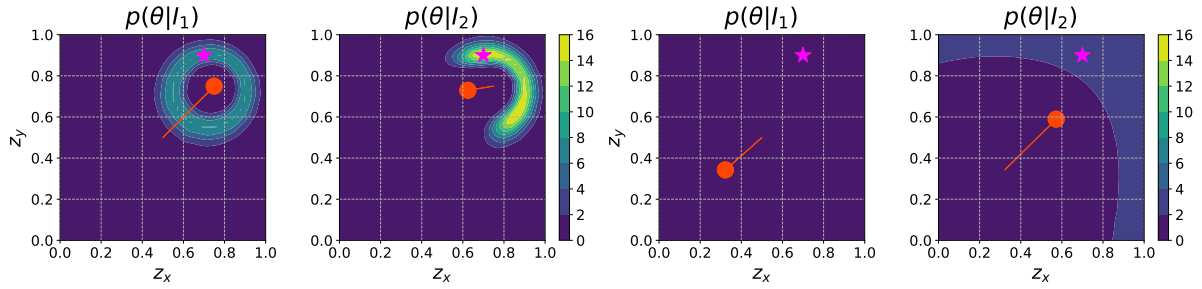

(a) $t = 0.05$ (b) $t = 0.2$

Figure 2.11: Case 2. Expected utility versus sensor location if conducting a single-experiment design at $t = 0.05$ and $t = 0.2$.

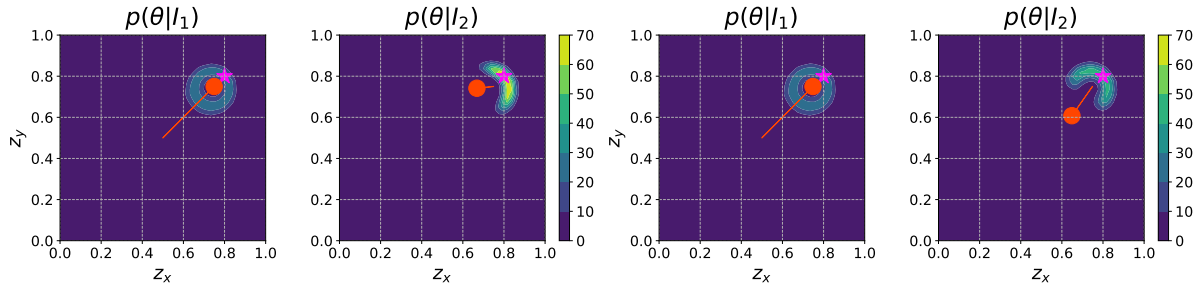



(a) PG-sOED versus greedy (b) PG-sOED versus batch

Figure 2.12: Case 2. Histograms of total rewards from $10^4$ test episodes generated using PG-sOED, greedy, and batch designs. The mean total reward for PG-sOED is $1.344 \pm 0.008$, higher than greedy design's $1.178 \pm 0.010$ and batch design's $1.264 \pm 0.007$.

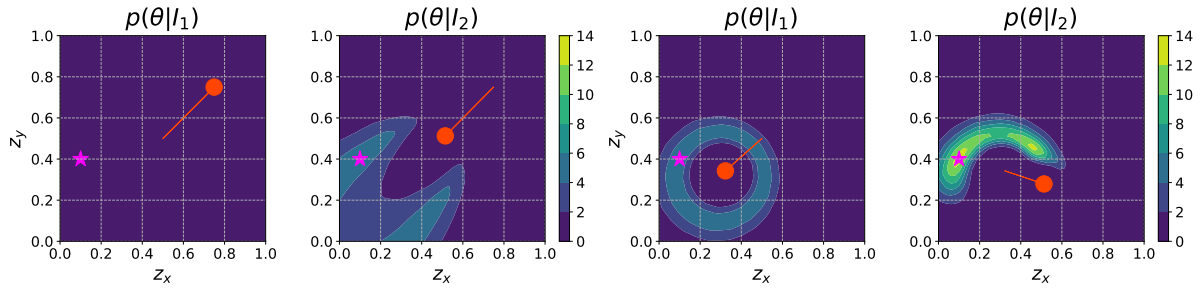



(a) PG-sOED, $\theta = (0.7, 0.9)$, **total reward = 2.020** (b) Greedy, $\theta = (0.7, 0.9)$, total reward = 0.493




(c) PG-sOED, $\theta = (0.8, 0.8)$, **total reward = 3.497** (d) Batch, $\theta = (0.8, 0.8)$, total reward = 3.120

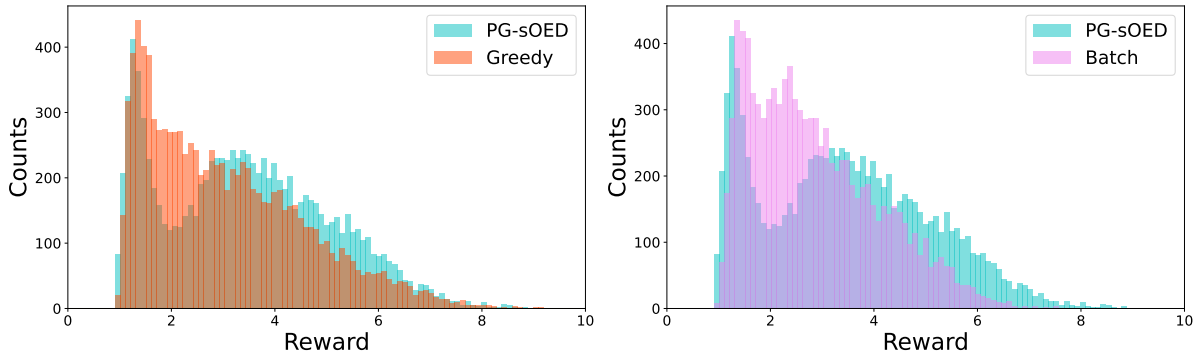
Figure 2.13: Case 2. Examples of episode instances where PG-sOED outperforms greedy and batch designs. The purple star represents the true $\theta$, red dot represents the physical state (vehicle location), red line segment tracks the vehicle displacement (design) from the preceding location, and contours plot the posterior PDF.




(a) PG-sOED, $\theta = (0.1, 0.4)$, total reward = 1.076 (b) Greedy, $\theta = (0.1, 0.4)$, **total reward = 1.687**

Figure 2.14: Case 2. Examples of episode instances where greedy design outperforms PG-sOED. The purple star represents the true $\theta$, red dot represents the physical state (vehicle location), red line segment tracks the vehicle displacement (design) from the preceding location, and contours plot the posterior PDF.

designs. In particular, PG-sOED features a prominent bimodal distribution of the total rewards, but also a heavier tail to the right leading to an overall greater mean compared to greedy and batch designs. Figure 2.16 shows the physical states $x_{k+1,p}$ (i.e., vehicle locations after the $k$-th experiments) from $10^4$ test episodes following PG-sOED, greedy, and batch designs. As expected, batch design produces identical movement paths since it is non-adaptive, while the PG-sOED and greedy designs scatter into the entire domain. Notably, we do not observe much movement to the left and bottom regions over the four experiments. This behavior can be explained by that (i) moving towards the bottom-left is against the convective direction and requires a higher movement cost, and (ii) the bottom-left offers lower information since the convection carries the contaminant towards the top-right.




(a) PG-sOED versus greedy

(b) PG-sOED versus batch

Figure 2.15: Case 3. Histograms of total rewards from $10^4$ test episodes generated using PG-sOED, greedy, and batch designs. The mean total reward for PG-sOED is $3.435 \pm 0.016$, higher than greedy design's $3.057 \pm 0.015$ and batch design's $2.856 \pm 0.012$.

We explain the policy behaviors with the aid of Fig. 2.17, which shows the marginal posterior PDF contours from episode instances obtained by PG-sOED, greedy, and batch designs. From the figure, we observe the first move for PG-sOED is towards the bottom-right, which appears counter-intuitive since the convective flows is towards the top-right. Moreover, the expected utility for designing a single experiment at $t_0$, shown in Fig. 2.18, further suggests "staying still" is favored since its maximum is near the center of the domain; indeed this is realized and adopted by the greedy and batch designs as seen in Fig. 2.17. However, moving towards the bottom-right is in fact an excellent design *for the long-term utility*. This can be explained by recognizing that the top-right region indeed offers the most informative measurements and all of the design strategy will eventually suggest an experiment there. However, taking a measurement in bottom-right region provides information from an orthogonal direction that enhances the future information gain from those top-right measurements. For example, if the true source location is near the bottom, the

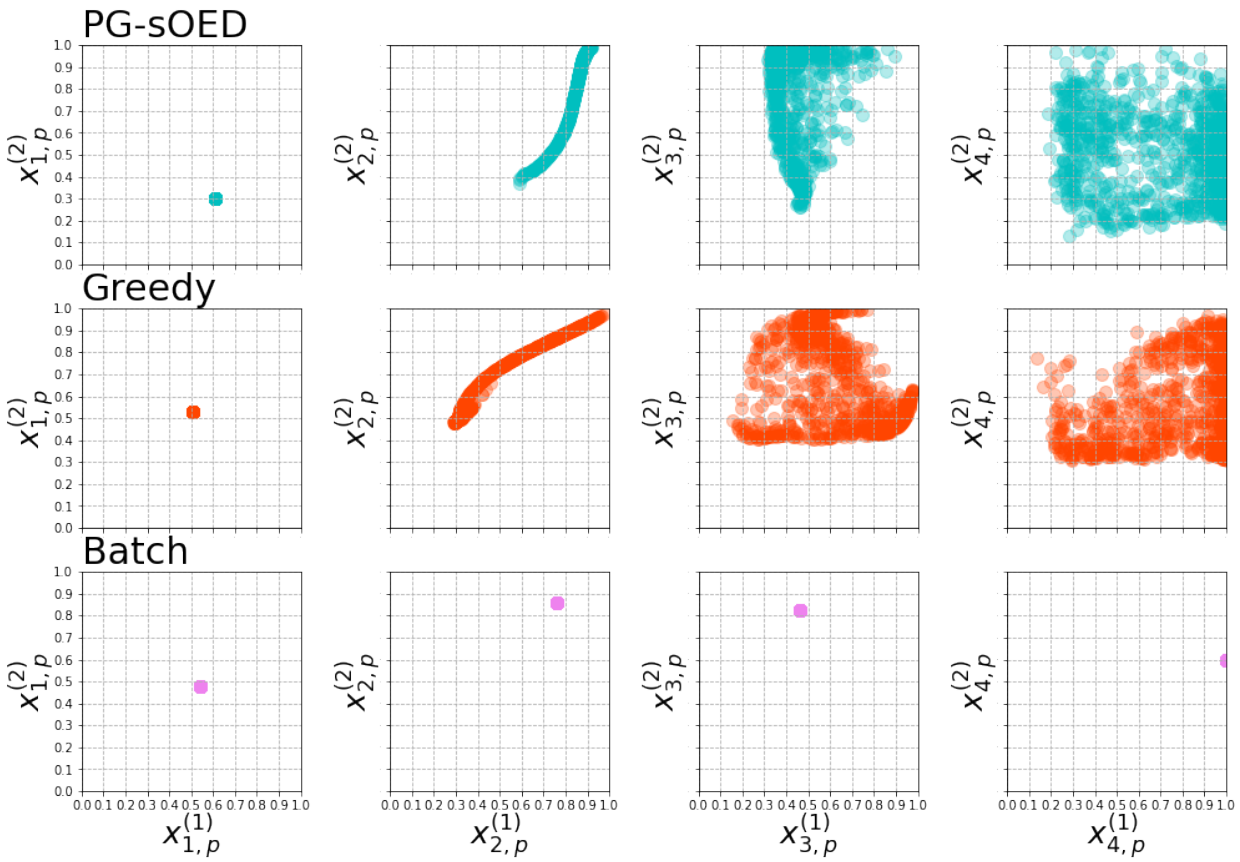



Figure 2.16: Case 3. Vehicle locations from $10^4$ test episodes generated using PG-sOED, greedy, and batch designs (rows) for experiments 1–4 (columns).

bottom-right experiment can detect its presence and adapt future experiments there to gain more information; if the true source location is near the top, the bottom-right experiment can detect its absence and substantially narrow down the posterior probability there and guide future experiments back towards the top-right. Furthermore, the best opportunity to move in this "off-stream" direction is in the first experiment, where the convective speed is lowest and so the penalty for moving against the convective flow is minimum. Overall, PG-sOED is able to reveal this low-cost information-orthogonal first move that enhances the value of subsequent experiments.

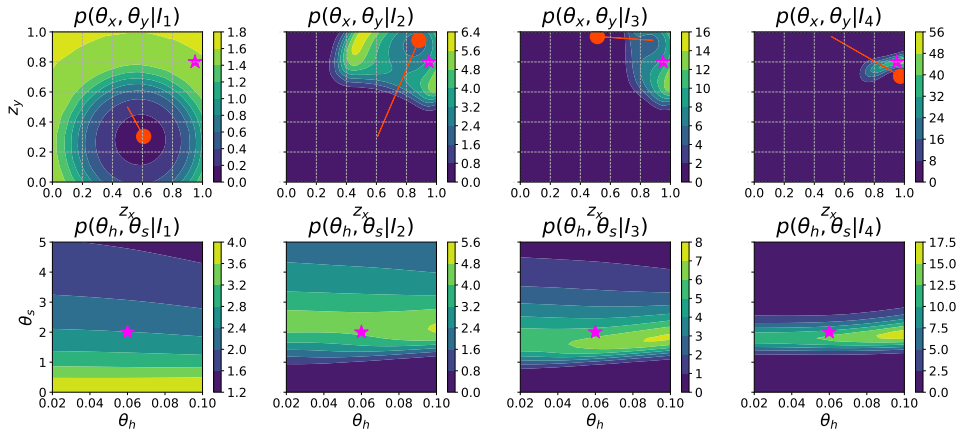
Lastly, Case 4 extends to designing $N = 15$ experiments using PG-sOED. When designing more experiments such as in this case, greedy design becomes very expensive and no longer practical due to its need for repeated Bayesian inference and incremental KL-divergence estimates at every experiment. This is in contrast to PG-sOED which remains inexpensive since it requires just a single terminal Bayesian inference and KL-divergence for each episode. We present the histogram of total rewards for PG-sOED and batch designs (greedy no longer practical) in Fig. 2.19, where the advantage of PG-sOED appears prominent.

## 2.4 Summary

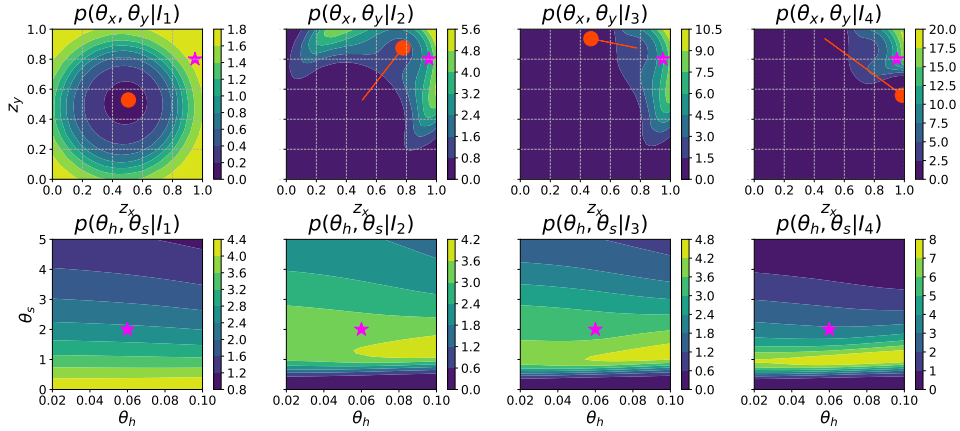
In this chapter, we introduce a comprehensive mathematical formulation for sOED that incorporates a state-space representation. We provide a proof of sOED's optimality, highlighting its superiority over batch and greedy designs. We then introduce new, computationally efficient methods to solve the sOED problem using policy gradient method (**PG-sOED**). We derive the PG expressions for sOED, enabling gradient-based optimization, and utilize deep neural networks to learn the policy function, value function and surrogate model.

The key contributions and novelty of our PG-sOED method are summarized as follows.

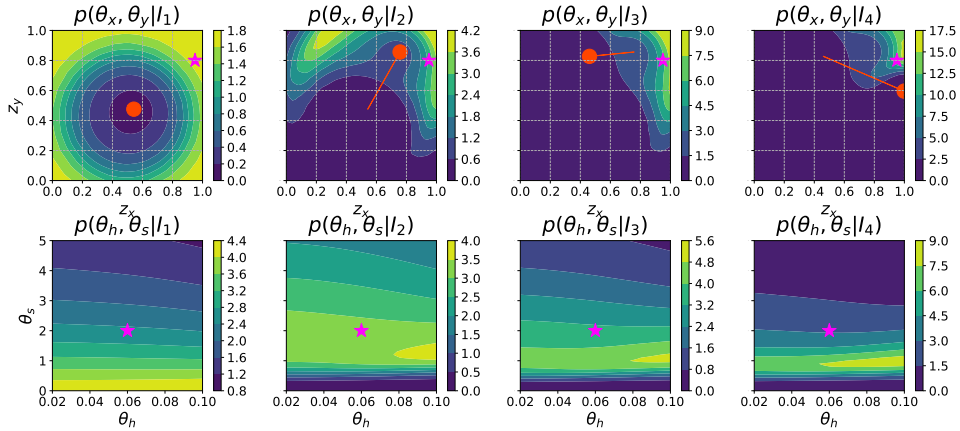
- We formulate the sOED problem as a finite-horizon POMDP under a Bayesian setting and with information-theoretic utilities. This formulation bridges the concepts of sequential experimental design with state-space modeling.
- We show that sOED generalizes the commonly-used batch and greedy design strategies.
- We provide a proof demonstrating the equivalence of the objective function when utilizing terminal information gain and incremental information gain.
- We present the new PG-sOED algorithm by deriving its policy gradient expressions, forming its Monte Carlo estimator, and adopting DNN parameterizations for the policy and value functions.




(a) PG-sOED, $\theta = (0.95, 0.8, 0.06, 0.2)$, **total reward = 3.818** (information gain = 4.156, movement cost = 0.338)




(b) Greedy, $\theta = (0.95, 0.8, 0.06, 0.2)$, total reward = 2.699 (information gain = 2.964, movement cost = 0.265)




(c) Batch, $\theta = (0.95, 0.8, 0.06, 0.2)$, total reward = 2.380 (information gain = 2.628, movement cost = 0.248)

Figure 2.17: Case 3. Example episode instances using PG-sOED, greedy and batch designs.

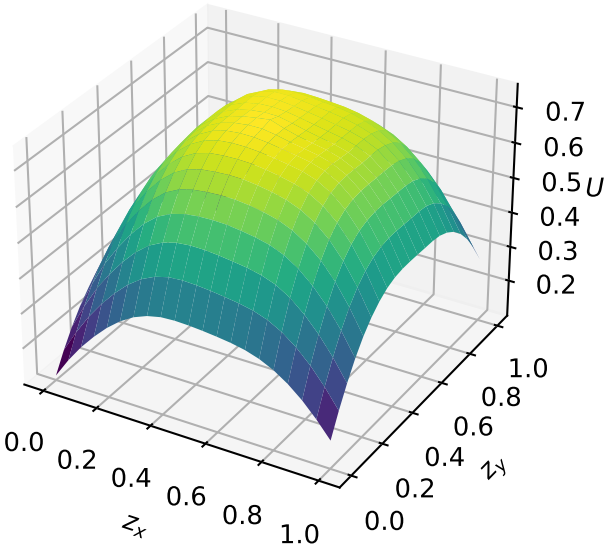



Figure 2.18: Case 3. Expected utility for one-experiment design at $t_1 = 0.05$. The best design location is the domain center.

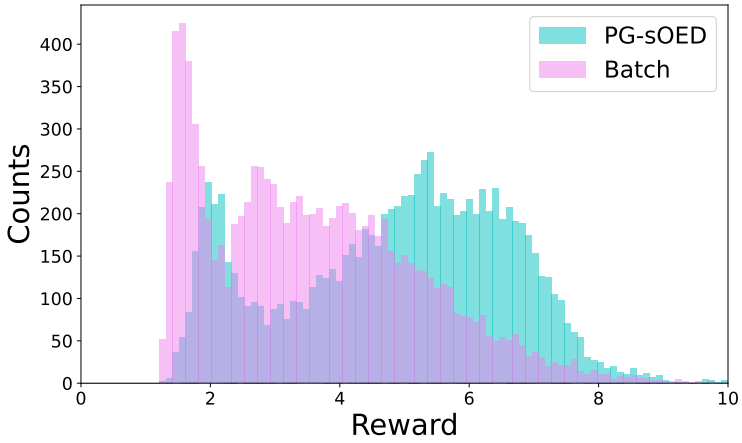



Figure 2.19: Case 4. Histograms of total rewards from $10^4$ test episodes generated using PG-sOED and batch designs. The mean total reward for PG-sOED is $4.853 \pm 0.018$, higher than batch design's $3.581 \pm 0.016$.

- We validate PG-sOED on a benchmark example and demonstrate its advantages over other design baselines via a sensor movement problem for contaminant source inversion in a convection-diffusion field. Notably, we provide explanations for the resulting policy behaviors using knowledge about the underlying physical process.
- We make available our PG-sOED code at https://github.com/wgshen/sOED.

# CHAPTER 3

# Variational Sequential Optimal Experimental Design

The sequential optimal experimental design (sOED) framework introduced in Chapter 2 faces two significant challenges. The first is the expensive computations for estimating the Kullback-Leibler (KL) divergence reward terms, especially in high-dimensional parameter spaces. The second is that sOED is only formulated for OED targeting single-model parameter inference, and cannot tackle scenarios with multiple models and different design objectives beyond parameter inference (e.g., OED for model discrimination, goal-oriented prediction, etc.). In this chapter, we introduce **variational sequential optimal experimental design (vsOED)** that provides an enhanced mathematical framework and new numerical methods to overcome these challenges.

This chapter begins with a brief review on the formulation of sOED, and introduces the new entities and notations needed for accommodating multi-model scenarios, and a unified reward structure that encompasses information gain for model discrimination, parameter inference, and goal-oriented prediction, even in the presence of nuisance parameters. We further adopt a lower bound estimator for the expected utility through variational approximation to the Bayesian posteriors in order to bypass the intensive calculations of KL divergence.

We then introduce the numerical methods for solving vsOED problems. The optimal design policy is obtained by simultaneously maximizing the variational lower bound and performing policy gradient updates, utilizing advanced reinforcement learning (RL) techniques, such as replay buffer and target network. Finally, we demonstrate vsOED for a range of OED problems targeting parameter inference, model discrimination, and goal-oriented prediction. These cases encompass explicit and implicit likelihoods, nuisance parameters, and physics-based partial differential equation (PDE) models. The results indicate substantially improved sample efficiency and reduced number of forward model simulations compared to existing sequential design algorithms.

The content of this chapter corresponds to the author's publication [129], and the code is available at: `https://github.com/wgshen/vsOED`.

## 3.1 Problem formulation

### 3.1.1 Background

Similar to the framework of sOED in Chapter 2, we focus on OED for a finite total of $N$ experiments indexed by $k \in \{0, \dots, N-1\}$, where each experiment can be conducted under design $d_k \in \mathcal{D}_k \subseteq \mathbb{R}^{N_d}$ and produces an observation $y_k \in \mathbb{R}^{N_y}$. The information sequence of all past experiments' designs and observations is denoted by $I_k = [d_0, y_0, \dots, d_{k-1}, y_{k-1}]$ (with $I_0 = \emptyset$). However, in contrast to sOED that only considers a single model with unknown parameters, in vsOED we will further consider a discrete set of $\mathcal{M}$ candidate models indexed by $m \in \{1, \dots, \mathcal{M}\}$ for describing the experimental process. Each model contains unknown parameters of interest (PoIs) $\theta_m \in \mathbb{R}^{N_{\theta_m}}$ we wish to learn from the experiments, nuisance parameters $\eta_m \in \mathbb{R}^{N_{\eta_m}}$ that are uncertain but not targeted for learning, and associated predictive quantities of interest (QoIs) $z_m \in \mathbb{R}^{N_{z_m}}$ that only depend on this model's parameters. For simplicity, we present these variables to be continuous and their dimensions remain constant across experiments; however this is not a requirement. The relationships of these entities may be summarized via an observation model

$$y_k = G_k(\theta_m, \eta_m, d_k; m, I_k) + \epsilon_k \tag{3.1}$$

where $G_k$ is the observation forward mapping and $\epsilon_k$ is the observation noise, and a predictive model

$$z_m = H(\theta_m, \eta_m; m) \tag{3.2}$$

where $H$ is the predictive forward mapping. In many engineering and science systems, the forward mappings $G_k$ and $H$ involve the most expensive computations (e.g., solving systems of PDEs). Hence, the number of forward solves is often used as the unit for computational cost assessments.

Adopting a Bayesian approach, after the $k$th experiment is carried out, the joint probability density function (PDF) on $m$, $\theta_m$, $\eta_m$ can be updated following Bayes' rule:

$$\begin{aligned} p(m, \theta_m, \eta_m | d_k, y_k, I_k) &= \frac{p(y_k | m, \theta_m, \eta_m, d_k, I_k)\, p(m, \theta_m, \eta_m | I_k)}{p(y_k | d_k, I_k)} \\ &= P(m | I_{k+1})\, p(\theta_m, \eta_m | m, I_{k+1}), \end{aligned} \tag{3.3}$$

where $p(m, \theta_m, \eta_m | I_k)$ is the prior, $p(y_k | m, \theta_m, \eta_m, d_k, I_k)$ is the likelihood, $p(y_k | d_k, I_k)$ is the marginal likelihood; and the joint posterior $p(m, \theta_m, \eta_m | d_k, y_k, I_k)$ (and similarly for the prior) can be factored into product of $P(m | I_{k+1})$ the posterior probability mass function (PMF) of model and $p(\theta_m, \eta_m | m, I_{k+1})$ the posterior PDF of parameters conditioned on model $m$. In the remainder

of this chapter, we adopt the convention where when $m$ is not explicitly mentioned, conditioning on $m$ is implied through other variables' subscripts, e.g., $p(\theta_m, \eta_m|I_k) = p(\theta_m, \eta_m|m, I_k)$. Upon propagating the parameter posterior through $H$, the posterior-predictive PDF for $z_m$ becomes

$$p(z_m|I_{k+1}) = \int_{\Theta,\mathcal{H}} p(\theta_m, \eta_m|I_{k+1})\, p(z_m|\theta_m, \eta_m)\, d\theta_m\, d\eta_m. \tag{3.4}$$

If $H$ is a deterministic model, then $p(z_m|\theta_m, \eta_m)$ collapses to a Dirac delta function.

The posterior after the $k$th experiment $p(m, \theta_m, \eta_m|d_k, y_k, I_k) = p(m, \theta_m, \eta_m|I_{k+1})$ serves as the prior for the $(k+1)$th experiment and is again applied to Eqn. (3.3). Hence, the Bayesian framework can be naturally and recursively used for sequential experiment.

### 3.1.2 Sequential optimal experimental design formulation

Below we briefly review the MDP-baseformulation of sOED and the core components of a Markov decision process (MDP) for better understanding.

**State.** $x_k = [x_{k,b}, x_{k,p}] \in \mathcal{X}_k$ is the state of the system and environment prior to the $k$th experiment. The belief state $x_{k,b}$ fully captures the state of uncertainty in $m$, $\theta_m$, $\eta_m$, and $z_m$, and the physical state $x_{k,p}$ tracks any non-uncertain design-relevant quantities. The belief state conceptualizes as the posterior following a Bayesian paradigm for updating the rational belief about an outcome's plausibility [38, 47]. In practice, this amounts to numerically representing the posteriors in Eqn. (3.3) and (3.4) or their sufficient statistics. We adopt the trivial sufficient statistics of the posterior, $I_k$, which also captures the physical state since it records the history of all past experiments; hence, we will adopt $x_k = I_k$ for the rest of this chapter. The effectiveness of using $I_k$ to represent $x_k$ has already been verified in the numerical cases presented in Sec. 2.3. The main drawback is that $\dim(I_k)$ grows with $k$, however it is always capped due to finite $N$.

**Design (action) and policy.** $\pi = \{\mu_k : \mathcal{X}_k \mapsto \mathcal{D}_k\}_{k=0}^{N-1}$ is the deterministic policy mapping from state space to design (action) space. The design for the $k$th experiment is thus $d_k = \mu_k(I_k)$.

**State transition.** $x_{k+1} = \mathcal{F}_k(x_k, d_k, y_k)$ describes the transition from state $x_k$ to state $x_{k+1}$ after conducting the $k$th experiment under design $d_k$ and observing $y_k$. Since we represent the state using $I_k$, the transition is simply a concatenation $I_{k+1} = [I_k, d_k, y_k]$.

**Utility (reward).** $g_k(I_k, d_k, y_k) \in \mathbb{R}$ denotes the immediate reward from the $k$th experiment, and $g_N(I_N) \in \mathbb{R}$ is the terminal reward that can be only computed after all experiments are completed. Examples of information gain (IG)-based rewards will be provided in Sec. 3.1.3 and Sec. 3.1.4.

**Problem statment.** The sOED problem seeks the policy that maximizes the expected utility $U$:

$$\pi^* = \underset{\pi=\{\mu_0,\dots,\mu_{N-1}\}}{\arg\max} \quad \left\{ U(\pi) = \mathbb{E}_{I_N|\pi,I_0}\left[\sum_{k=0}^{N-1} g_k(I_k, d_k, y_k) + g_N(I_N)\right]\right\} \tag{3.5}$$
$$\text{s.t.} \quad d_k = \mu_k(I_k) \in \mathcal{D}_k,$$
$$I_{k+1} = [I_k, d_k, y_k], \qquad \text{for} \quad k = 0, \dots, N-1, .$$

This sOED framework has been shown to generalize the batch and sequential greedy designs in Sec. 2.1.3.

### 3.1.3 Experimental design utilities

Similar to Sec. 2.1.4, we propose two IG-based reward formulations incorporating various design objectives.

1) *Terminal-information-gain (TIG)* targets the overall IG (KL divergence) from all $N$ experiments via the terminal reward (without loss of generality, contributions from non-information reward are omitted):

$$g_k(I_k, d_k, y_k) = 0, \qquad k = 0, \dots, N-1 \tag{3.6}$$
$$g_N(I_N) = \alpha_{\mathcal{M}} D_{\mathrm{KL}}\left( P(m|I_N) \,||\, P(m) \right) + \mathbb{E}_{m|I_N}\left[\alpha_\Theta D_{\mathrm{KL}}\left( p(\theta_m|I_N) \,||\, p(\theta_m) \right) + \alpha_Z D_{\mathrm{KL}}\left( p(z_m|I_N) \,||\, p(z_m) \right)\right], \tag{3.7}$$

where $\alpha_{\mathcal{M}} \in [0,1]$ (for model), $\alpha_\Theta \in [0,1]$ (for PoIs) and $\alpha_Z \in [0,1]$ (for QoIs) are the weights/switches of IG from the different variables. For example, setting $\alpha_{\mathcal{M}} = 1$ and $\alpha_\Theta = \alpha_Z = 0$ reduces to only IG for model probability (OED for model discrimination); $\alpha_\Theta = 1$ and $\alpha_{\mathcal{M}} = \alpha_Z = 0$ reduces to IG on PoIs (OED for inference); $\alpha_Z = 1$ and $\alpha_\Theta = \alpha_{\mathcal{M}} = 0$ reduces to IG on QoIs (OED for goal-oriented prediction). In the special case when $\alpha_{\mathcal{M}} = \alpha_\Theta = 1$ and $\alpha_Z = 0$, or $\alpha_{\mathcal{M}} = \alpha_Z = 1$ and $\alpha_\Theta = 0$, the terminal reward is equivalent to $D_{\mathrm{KL}}\left( p(m, \theta_m|I_N) \,||\, p(m, \theta_m) \right)$ and $D_{\mathrm{KL}}\left( p(m, z_m|I_N) \,||\, p(m, z_m) \right)$, respectively (see Appendix B.1). When nuisance parameter $\eta_m$ is absent, one should not set both $\alpha_\Theta$ and $\alpha_Z$ to 1, since the IG on $z_m$ is fully absorbed into the IG on $\theta_m$ (Appendix B.2).

2) *Incremental-information-gain (IIG)* adopts incremental IG for the immediate rewards:

$$g_k(I_k, d_k, y_k) = \alpha_{\mathcal{M}} D_{\mathrm{KL}}\left( P(m|I_{k+1}) \,||\, P(m|I_k) \right) + \mathbb{E}_{m|I_{k+1}}\Big[\alpha_\Theta D_{\mathrm{KL}}\left( p(\theta_m|I_{k+1}) \,||\, p(\theta_m|I_k) \right) + \alpha_Z D_{\mathrm{KL}}\left( p(z_m|I_{k+1}) \,||\, p(z_m|I_k) \right)\Big], \qquad k = 0, \dots, N-1 \tag{3.8}$$
$$g_N(I_N) = 0. \tag{3.9}$$

We denote $U_T(\pi)$ to be the resulting sOED expected utility from Eqn. (3.5) when adopting the TIG rewards (Eqn. (3.6) and (3.7)), and $U_I(\pi)$ when adopting the IIG rewards (Eqn. (3.8) and (3.9)).

**Theorem 3** (Terminal-incremental equivalence)**.** *$U_T(\pi) = U_I(\pi)$ for any policy $\pi$.*

A proof is provided in Appendix B.3. Hence, both formulations induce the same sOED problem.

### 3.1.4 One-point estimate for rewards

Direct evaluation of the expected utility requires repeated KL divergence (integral) estimates. Naïve estimates using grid discretization or MC integration would be highly expensive. Similar to [55], we propose one-point estimates that are much less costly. The 1) *one-point-TIG* formulation involves

$$\dot{g}_k(I_k, d_k, y_k) = 0, \qquad k = 0, \dots, N-1 \tag{3.10}$$

$$\dot{g}_N(I_N) = \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_N)}{P(\dot{m})} + \alpha_\Theta \ln \frac{p(\dot{\theta}_m|I_N)}{p(\dot{\theta}_m)} + \alpha_Z \ln \frac{p(\dot{z}_m|I_N)}{p(\dot{z}_m)}; \tag{3.11}$$

and the 2) *one-point-IIG* involves

$$\begin{aligned} \dot{g}_k(I_k, d_k, y_k) = \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_{k+1})}{P(\dot{m}|I_k)} + \alpha_\Theta \ln \frac{p(\dot{\theta}_m|I_{k+1})}{p(\dot{\theta}_m|I_k)} \\ + \alpha_Z \ln \frac{p(\dot{z}_m|I_{k+1})}{p(\dot{z}_m|I_k)}, \qquad k = 0, \dots, N-1 \end{aligned} \tag{3.12}$$

$$\dot{g}_N(I_N) = 0. \tag{3.13}$$

In the above, $\dot{m}$, $\dot{\theta}_m$, and $\dot{z}_m$ are the "true" sample values that generated the sequence $I_N$ appearing in the conditionals. The corresponding *one-point estimate* expected utility is:

$$\dot{U}(\pi) = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \mathbb{E}_{I_N|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \left[ \sum_{k=0}^{N-1} \dot{g}_k(I_k, d_k, y_k) + \dot{g}_N(I_N) \right]. \tag{3.14}$$

We denote $\dot{U}_T(\pi)$ to be the resulting expected utility from Eqn. (3.14) when adopting the one-point-TIG rewards (Eqn. (3.10) and (3.11)), and $\dot{U}_I(\pi)$ when adopting the one-point-IIG rewards (Eqn. (3.12) and (3.13)).

**Theorem 4** (One-point estimate equivalence)**.** *$U_T(\pi) = \dot{U}_T(\pi) = \dot{U}_I(\pi) = U_I(\pi)$ for any policy $\pi$.*

A proof is provided in Appendix B.4. Hence, both the original sOED and one-point estimate formulations, using either TIG or IIG, induce the same sOED problem. We note that for $\dot{U}_I(\pi)$, all the intermediate posteriors cancel out and only the prior $p(\cdot)$ (i.e., $p(\cdot|I_0)$) and the final posterior $p(\cdot|I_N)$ survive. However, working with intermediate posteriors in the incremental rewards can

lead to denser rewards that improves numerical performance [14]. For any expected utility form, the prior term $p(\cdot)$ may be omitted since it would only result in an objective shift and not affect the arg-max (see Appendix B.5). For cases when the prior is difficult to compute (e.g., prior-predictive $p(z_m) = \int_{\Theta,\mathcal{H}} p(\theta_m, \eta_m)\, p(z_m|\theta_m, \eta_m)\, d\theta_m\, d\eta_m$ that needs to marginalize out $\theta_m$ and $\eta_m$), we will drop the prior term and use the shifted expected utility for policy optimization.

The one-point estimates proposed in Sec. 3.1.4 still require posterior density evaluations. Inspired by [56], we replace the true posteriors $p(\cdot|I_k)$ with variational posterior approximations $q(\cdot|I_k; \phi_{(\cdot)})$, forming a lower bound estimator to the expected utility. The 1) *variational-one-point-TIG* becomes

$$\dot{g}_k(I_k, d_k, y_k; \phi) = 0, \qquad k = 0, \dots, N-1 \tag{3.15}$$

$$\dot{g}_N(I_N; \phi) = \alpha_{\mathcal{M}} \ln \frac{q(\dot{m}|I_N; \phi_{\mathcal{M}})}{P(\dot{m})} + \alpha_{\Theta} \ln \frac{q(\dot{\theta}_m|I_N; \phi_{\Theta_m})}{p(\dot{\theta}_m)} + \alpha_Z \ln \frac{q(\dot{z}_m|I_N; \phi_{Z_m})}{p(\dot{z}_m)}; \tag{3.16}$$

and the 2) *variational-one-point-IIG* becomes

$$\begin{aligned}\dot{g}_k(I_k, d_k, y_k; \phi) = \alpha_{\mathcal{M}} \ln \frac{q(\dot{m}|I_{k+1}; \phi_{\mathcal{M}})}{q(\dot{m}|I_k; \phi_{\mathcal{M}})} + \alpha_{\Theta} \ln \frac{q(\dot{\theta}_m|I_{k+1}; \phi_{\Theta_m})}{q(\dot{\theta}_m|I_k; \phi_{\Theta_m})} \\ + \alpha_Z \ln \frac{q(\dot{z}_m|I_{k+1}; \phi_{Z_m})}{q(\dot{z}_m|I_k; \phi_{Z_m})}, \qquad k = 0, \dots, N-1\end{aligned} \tag{3.17}$$

$$\dot{g}_N(I_N; \phi) = 0, \tag{3.18}$$

with the understanding that $q(\cdot|I_0; \phi_{(\cdot)})$ is $p(\cdot|I_0)$. The corresponding *variational one-point estimate* expected utility is:

$$\dot{U}(\pi; \phi) = \mathbb{E}_{\dot{m}, \dot{\theta}_m, \dot{\eta}_m, \dot{z}_m} \mathbb{E}_{I_N|\pi, I_0, \dot{m}, \dot{\theta}_m, \dot{\eta}_m} \left[ \sum_{k=0}^{N-1} \dot{g}_k(I_k, d_k, y_k; \phi) + \dot{g}_N(I_N; \phi) \right]. \tag{3.19}$$

Following similar ideas in non-sequential OED [56, 7], we show this to be a lower bound of the expected utility.

**Theorem 5** (Variational lower bound)**.** *$\dot{U}(\pi; \phi) \leq \dot{U}(\pi) = U(\pi)$ for any policy $\pi$ and variational posterior parameter $\phi$. The bound is tight if and only if $q(\cdot|I_N; \phi_{(\cdot)}) = p(\cdot|I_N)$ (except the trivial case when $\alpha_{\mathcal{M}} = \alpha_{\Theta} = \alpha_Z = 0$).*

A proof is provided in Appendix B.6. The **variational sOED (vsOED)** problem thus entails

finding the optimal variational approximation and the optimal policy to maximize the lower bound:

$$\begin{aligned} \pi^*, \phi^* = \arg\max_{\pi,\phi} \quad & \dot{U}(\pi; \phi) \\ \text{s.t.} \quad & d_k = \mu_k(I_k) \in \mathcal{D}_k, \\ & I_{k+1} = [I_k, d_k, y_k], \qquad \text{for} \quad k = 0, \ldots, N-1. \end{aligned} \tag{3.20}$$

We note that the tightness of the bound does not depend on the quality of the intermediate variational posteriors (i.e., $q(\cdot|I_k; \phi_{(\cdot)})$ for $k = 1, \ldots, N-1$) due to their cancellations, and low-quality intermediate posterior approximation may be used (see Appendix B.7). For example, when all intermediate posteriors are approximated by the prior, then the *one-point-IIG* collapses to the *one-point-TIG*. However, as we show in the results, good intermediate posterior approximations can lead to better numerical performance. Non-IG-based reward contributions (that do not depend on the posteriors) can also be incorporated into all previously introduced expected utility formulations without affecting any of the theorem results.

In our implementation, we employ a neural network (NN) to approximate the model posterior $q(\dot{m}|I_k; \phi_{\mathcal{M}})$, which takes $d_k$'s and $y_k$'s and uses a softmax output activation to produce each model's probability. For parameter posteriors $q(\dot{\theta}_m|I_k; \phi_{\Theta_m})$ and $q(\dot{z}_m|I_k; \phi_{Z_m})$, we use independent Gaussian mixture models (GMMs) with NNs predicting the GMM weights, means, and standard deviations. Truncated Gaussian is used for parameter with compact support. We also include results using normalizing flows (NFs) for parameter posterior approximations, where the NF work is contributed by collaborator Jiayuan Dong. Further details are in Sec. 3.2.2 and 3.2.3.

## 3.2 Numerical Methods for vsOED

### 3.2.1 Policy gradient and variational gradient

We employ gradient-based methods to numerically solve for the optimal vsOED policy. To extract gradient, we explicitly parameterize policy $\pi$ by $w \in \mathbb{R}^{N_w}$ and denote the parameterized policy as $\pi_w$. Learning the policy (i.e. *actor*) explicitly offers significantly faster online usage compared to dynamic programming sOED [74] and myopic design that require solving new optimization problems at run time, which has been shown in Sec. 2.3.1. Using parameterized policy, the vsOED

problem becomes:

$$\begin{aligned} w^*, \phi^* = \arg\max_{w,\phi} \quad & \dot{U}(w;\phi) \\ \text{s.t.} \quad & d_k = \mu_{k,w}(I_k) \in \mathcal{D}_k, \\ & I_{k+1} = [I_k, d_k, y_k], \qquad \text{for} \quad k = 0, \dots, N-1. \end{aligned} \tag{3.21}$$

The gradient of the expected utility with respect to $\phi$ can be trivially shown (Leibniz rule) to be:

$$\nabla_\phi \dot{U}(w;\phi) = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \mathbb{E}_{I_N|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \left[ \sum_{k=0}^{N-1} \nabla_\phi \dot{g}_k(I_k, d_k, y_k;\phi) + \nabla_\phi \dot{g}_N(I_N;\phi) \right]. \tag{3.22}$$

The policy gradient can be derived near-identically following the proof in Appendix A.2 except that the expressions for vsOED involve an additional outer expectation over $\dot{m}, \dot{\theta}_m, \dot{\eta}_m, \dot{z}_m$; therefore the derivation is not repeated. The vsOED policy gradient is:

$$\nabla_w \dot{U}(w;\phi) = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \sum_{k=0}^{N-1} \mathbb{E}_{I_k|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \left[ \nabla_w \mu_{k,w}(I_k) \nabla_{d_k} Q_k^{\pi_w}(I_k, d_k) \Big|_{d_k=\mu_{k,w}(I_k)} \right], \tag{3.23}$$

where $Q_k^{\pi_w}$ is the *actor-value function* (i.e. *critic*) that quantifies the expected cumulative remaining reward for conducting $k$th experiment at design $d_k$ and state $I_k$ and thereafter following policy $\pi_w$.

In our implementation, we use NNs to parameterize both the actor and critic (architecture in Sec. 3.2.4), with the critic parameters being $\nu \in \mathbb{R}^{N_\nu}$. Details about the overall numerical algorithm are provided in the following sections, including the NN architectures (Sec. 3.2.2 to 3.2.4), MC estimate of the variational gradient (Sec. 3.2.5.1) and policy gradient (Sec. 3.2.5.3), formulation and training of critic networks under TIG and IIG (Sec. 3.2.5.2), exploration strategy (Sec. 3.2.5.4), and hyperparameter tuning (Sec. 3.2.5.5). The overall pseudocode is summarized in Algorithm 2.

**Algorithm 2:** The vsOED algorithm.

1: Initialize variational parameters $\phi$, actor (policy) parameters $w$, critic parameters $\nu$;
2: **for** $l = 1, \dots, n_{\text{update}}$ **do**
3: Simulate $n_{\text{episode}}$ episodes: sample $m$, $\theta_m$, $\eta_m$ and $z_m$ from the prior, and then for $k = 0, \dots, N-1$ sample $d_k = \mu_{k,w}(I_k) + \epsilon_{\text{explore}}$ and $y_k \sim p(y_k|m, \theta_m, \eta_m, d_k, I_k)$;
4: Update newly generated information sequences $\left\{ I_N^{(i)} \right\}_{i=1}^{n_{\text{episode}}}$ into replay buffer;
5: Sample $n_{\text{batch}}$ episodes from the replay buffer, update $\phi$ and $\left\{ \dot{g}_k^{(i)} \right\}_{i=1}^{n_{\text{batch}}}$ using sampled batch;
6: Estimate gradients and update $\nu$ and $w$ via gradient ascent using sampled batch;
7: **end for**
8: Return optimized policy network $\pi_w$;

### 3.2.2 Neural network architecture of model posterior predictor

The overall architecture of a NN-based posterior predictor for model probability ($q(m|I_k;\phi_{\mathcal{M}})$) is shown in Table 3.1; the same architecture is utilized for all numerical cases in this chapter. More specifically, the model posterior predictor takes $I_k$ as input, and outputs the log-probabilities of each candidate model $\ln q(m|I_k;\phi_{\mathcal{M}})$. Separate model posterior predictors are trained for each stage when the IIG formulation is used. However, as shown earlier in Appendix B.7, the quality of the intermediate variational posterior approximations does not directly contribute to the accuracy of the overall variational expected utility estimate, and thus one may elect to train these intermediate model posterior predictors more "loosely", for example, by using smaller NN architectures and with shared weights among the NNs.

Table 3.1: Architecture of the NN-based model posterior predictor.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | $I_k$ | $k(N_d + N_y)$ | - |
| H1 | Dense | 256 | ReLU |
| H2 | Dense | 256 | ReLU |
| H3 | Dense | 256 | ReLU |
| Output | Dense | $\mathcal{M}$ | LogSoftmax |

### 3.2.3 Neural network architectures of parameter and predictive quantity posterior predictors

We introduce the GMM- and NF-based posterior predictors for PoIs ($q(\theta_m|I_k;\phi_{\Theta_m})$) and QoIs ($q(z_m|I_k;\phi_{Z_m})$). We adopt the same architectures for both the PoI and QoI posterior predictors, and so only introduce them in the context of PoIs below but with the understanding that the same applies to the QoIs. Similar to the model posterior predictor, separate PoI and QoI posterior predictors are trained for each stage when the IIG formulation is used.

#### 3.2.3.1 Independent Gaussian Mixture Models

An independent GMM approximates a complex distribution through a weighted sum of multiple independent Gaussians:

$$q(\theta_m|I_k;\phi_{\Theta_m}) = \sum_{i=1}^{n_{\text{mixture}}} w_i(I_k;\phi_{\Theta_m})\,\mathcal{N}(\theta_m;\mu_i(I_k;\phi_{\Theta_m}),\Sigma_i(I_k;\phi_{\Theta_m})), \tag{3.24}$$

where for the $i$th Gaussian, $w_i(I_k; \phi_{\Theta_m})$ is its mixture weight, $\mu_i(I_k; \phi_{\Theta_m}) \in \mathbb{R}^{N_{\theta_m}}$ is its mean, and $\Sigma_i(I_k; \phi_{\Theta_m}) \in \mathbb{R}^{N_{\theta_m} \times N_{\theta_m}}$ is its diagonal covariance matrix with the square root of the diagonal terms being the standard deviations. The weights, means, and standard deviations of the GMM are predicted using NNs, together referred to as the GMM net. These NNs share a common backend network that learns shared features. The architectures of the feature net, weight net, mean net and standard deviation net are provided in Table 3.2 to 3.4. The *Linear mapping* in Table 3.4 refers to the process of mapping the output to a specific range that is problem dependent. This mapping ensures that the predicted means and standard deviations of the GMM fall within the desired range. Additionally, an epsilon of $10^{-27}$ is added to Eqn. (3.24) to prevent numerical underflow. When some PoIs have compact support, independent truncated normal distributions [26] are used to replace the dimensions corresponding to those PoIs within the Gaussian distributions. The specific ranges of the linear mapping and the usage of truncated normal will be mentioned in each numerical case. The same GMM net architecture is used across all numerical cases.

Table 3.2: Architecture of the feature net of the GMM net.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | $I_k$ | $k(N_d + N_y)$ | - |
| H1 | Dense | 256 | ReLU |
| Output | Dense | 256 | ReLU |

Table 3.3: Architecture of the weight net of the GMM net.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | Feature($I_k$) | 256 | - |
| H1 | Dense | 256 | ReLU |
| H2 | Dense | 256 | ReLU |
| Output | Dense | $n_{\text{mixture}}$ | Softmax |

Table 3.4: Architecture of the mean net or standard deviation net of the GMM net.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | Feature($I_k$) | 256 | - |
| H1 | Dense | 256 | ReLU |
| H2 | Dense | 256 | ReLU |
| H3 | Dense | $n_{\text{mixture}} N_{\theta_m}$ | Sigmoid |
| Output | Identity | $n_{\text{mixture}} N_{\theta_m}$ | Linear mapping |

#### 3.2.3.2 Normalizing Flows

The NF setup in this section is contributed by collaborator Jiayuan Dong. A NF approximates a target random variable $\theta$ by finding an overall invertible mapping to this target from a standard normal of the same dimension, $\theta = g(\xi)$ (and $\xi = f(\theta)$ where $f = g^{-1}$), via a composition of successive invertible mappings. The PDFs of these random variables are related via

$$p_\Theta(\theta) = p_\xi(f(\theta))|\text{det}Df(\theta)| \tag{3.25}$$

where $Df(\theta)$ is the Jacobian of $f$ at $\theta$. Writing in a successive mapping form $\theta = g(\xi) = g_n \circ g_{n-1} \circ ... \circ g_1(\xi) = g_n(g_{n-1}(...(g_1(\xi))...))$ with $n \geq 1$ invertible transformations, the log density is

$$\ln p_\Theta(\theta) = \ln p_\xi(f_n \circ f_{n-1} \circ ... \circ f_1(\theta)) + \sum_{i=1}^{n} \ln |\text{det}Df_i \circ f_{i-1} \circ ...f_1(\theta)| \tag{3.26}$$

where $f(\theta) = f_n \circ f_{n-1} \circ ... \circ f_1(\theta)$ and $f_i = g_i^{-1}$. The successive transformations on $\xi$ can achieve a highly expressive density for the target variable $\theta$ [42].

To approximate the PoI posterior $q(\theta_m|I_k;\phi_{\Theta_m})$, we build NF $\xi = f(\theta)$ using an invertible neural network (INN) [42], and refer to the overall mapping as the NF net. INN partitions $\theta$ into two parts $\theta = [\theta_1, \theta_2]^T$ with approximately equal dimensions, and introduces invertible mappings

$$\begin{aligned} f_1(\theta) &= \begin{pmatrix} \theta_1 \\ \tilde{\theta}_2 = \theta_2 \odot \exp(s_1(\theta_1)) + t_1(\theta_1) \end{pmatrix} \\ f_2(f_1(\theta)) &= \begin{pmatrix} \tilde{\theta}_1 = \theta_1 \odot \exp(s_2(\tilde{\theta}_2)) + t_2(\tilde{\theta}_2) \\ \tilde{\theta}_2 \end{pmatrix} \end{aligned} \tag{3.27}$$

where $s_1, t_1$ map $\mathbb{R}^{n_{\theta_1}} \mapsto \mathbb{R}^{n_{\theta_2}}$ and $s_2, t_2$ map $\mathbb{R}^{n_{\theta_2}} \mapsto \mathbb{R}^{n_{\theta_1}}$, and $\odot$ denotes element-wise product. The Jacobian of $f_1$ is

$$\begin{bmatrix} \mathbb{I}_d & 0 \\ \frac{\partial f_1(\theta)}{\partial \theta_2} & \text{diag}(\exp[s_1(\theta_1)]) \end{bmatrix},$$

a lower triangular matrix with determinant $\exp[\sum_{j=1}^{n_{\theta_2}} s_1(\theta_1)_j]$. Similarly the Jacobian of $f_2$ is an upper triangular matrix with determinant $\exp[\sum_{j=1}^{n_{\theta_1}} s_2(\tilde{\theta}_2)_j]$. $s$'s and $t$'s can represented via, for example, NNs for their expressiveness. Multiple such transformations from Eqn. (3.27) can also be composed together to further increase expressiveness of the overall mapping; we use $n_{\text{trans}}$ to denote the number of such transformation.

To incorporate the dependency of posterior on $I_k$, the $s(\cdot)$ and $t(\cdot)$ are set up to also take $I_k$ as input. Similar to the GMM setup, $I_k$ is first fed into a feature network whose output has the same dimension as $I_k$. The architectures of the feature network, and the $s_1, t_1; s_2, t_2$ networks in NFs are provided in Table 3.5 to 3.7. Mirroring the GMM net, an epsilon of $10^{-27}$ is added to Eqn. (3.26) to prevent numerical underflow.

Table 3.5: Architecture of the feature net of the NF net. The first value under *Dimension* column is used for the source location problem in 3.3.2 and CES problem in 3.3.3; the second value is used for the SIR problem in 3.3.4.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | $I_k$ | $k(N_d + N_y)$ | - |
| H1 | Dense | 256 / 128 | ReLU |
| H2 | Dense | 256 / 128 | ReLU |
| H3 | Dense | 256 / None | ReLU |
| Output | Feature($I_k$) | $k(N_d + N_y)$ | - |

Table 3.6: Architecture of the $s_1$ and $t_1$ nets of the NF net. The first value under *Dimension* column is used for the source location problem in 3.3.2 and CES problem in 3.3.3; the second value is used for the SIR problem in 3.3.4.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | Feature($I_k$) + $\theta_1$ | $k(N_d + N_y) + n_{\theta_1}$ | - |
| H1 | Dense | 256 / 128 | ReLU |
| H2 | Dense | 256 / 128 | ReLU |
| H3 | Dense | 256 / 128 | ReLU |
| Output | $s_1(\cdot)$ or $t_1(\cdot)$ | $n_{\theta_2}$ | - |

### 3.2.4 Neural network architecture of actor and critic

The same architectures of the actor and critic networks described in Sec. 2.2.2.1 and 2.2.2.2 are adopted here. The actor $\mu_{k,w}$ learns a mapping from state $I_k$ to design $d_k$. Instead of learning separate actors for each stage, we combine them into a single actor. The overall input takes the form

$$I_k^{actor} = [e_k, \tilde{I}_k]$$

Table 3.7: Architecture of the $s_2$ and $t_2$ nets of the NF net. The first value under *Dimension* column is used for the source location problem in 3.3.2 and CES problem in 3.3.3; the second value is used for the SIR problem in 3.3.4.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | Feature($I_k$) + $\tilde{\theta}_2$ | $k(N_d + N_y) + n_{\theta_2}$ | - |
| H1 | Dense | 256 / 128 | ReLU |
| H2 | Dense | 256 / 128 | ReLU |
| H3 | Dense | 256 / 128 | ReLU |
| Output | $s_2(\cdot)$ or $t_2(\cdot)$ | $n_{\theta_1}$ | - |

where $e_k$ is an 0-indexed one-hot encoding vector of size $N$ to represent the current experiment stage:

$$e_k = [0, \ldots, 0, \underbrace{1}_{k\text{th}}, 0, \ldots, 0]^T,$$

and $\tilde{I}_k$ is a vector of fixed size $(N-1)(N_d + N_y)$ obtained by extending $I_k$ with zero-padding:

$$\tilde{I}_k = [\overbrace{d_0}^{N_d}, \ldots, d_{k-1}, \underbrace{0, \ldots, 0}_{N_d(N-1-k)}, \overbrace{y_0}^{N_y}, \ldots, y_{k-1}, \underbrace{0, \ldots, 0}_{N_y(N-1-k)}]^T.$$

The total dimension of $I_k^{actor}$ is $N + (N-1)(N_d + N_y)$. The inputs of the critic is

$$I_k^{critic} = [I_k^{actor}, d_k]$$

with total dimension $N + (N-1)(N_d + N_y) + N_d$. The output of the critic is a scalar. The architectures of the actor and critic are presented in Table 3.8 and Table 3.9, where the *Linear mapping* in Table 3.8 maps the output value to be within the design bounds. The same actor and critic architectures are used across all numerical cases in this chapter.

Other architectures have been proposed for constructing the actor and critic networks, such as the *encoder-pooling-emitter* structure used in [55, 76, 14]. These architectures leverage a permutation invariance property that arises from the conditional independence of likelihoods. However, it is important to note that in our case, the conditional independence of likelihoods does not hold, even if the likelihood function does not depend on past experience (i.e. if $p(y_k|m, \theta_m, \eta_m, d_k, I_k) = p(y_k|m, \theta_m, \eta_m, d_k)$). For instance, if the problem involves multiple models, the model posterior

Table 3.8: Architecture of the actor.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | $I_k^{actor}$ | $N + (N-1)(N_d + N_y)$ | - |
| H1 | Dense | 256 | ReLU |
| H2 | Dense | 256 | ReLU |
| H3 | Dense | 256 | ReLU |
| H4 | Dense | $N_d$ | Sigmoid |
| Output | Identity | $N_d$ | Linear mapping |

Table 3.9: Architecture of the critic.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | $I_k^{critic}$ | $N + (N-1)(N_d + N_y) + N_d$ | - |
| H1 | Dense | 256 | ReLU |
| H2 | Dense | 256 | ReLU |
| H3 | Dense | 256 | ReLU |
| Output | Dense | 1 | - |

can be shown to be not permutation invariant:

$$
\begin{aligned}
P(m|I_N) &\propto P(m) \prod_{k=0}^{N-1} \int_{\Theta,\mathcal{H}} p(y_k|m,\theta_m,\eta_m,d_k)\, p(\theta_m,\eta_m|m,I_k)\, d\theta_m\, d\eta_m \\
&= P(m) \prod_{k=0}^{N-1} p(y_k|m,I_k,d_k) \\
&\neq P(m) \prod_{k=0}^{N-1} p(y_k|m,d_k)
\end{aligned}
$$

since in the middle equation, we see that the factored marginal likelihoods in our case depend on the entire history $I_k$. Even when the problem only involves a single model (i.e. $\mathcal{M} = 1$), the PoI

posterior is not permutation invariant if nuisance parameters are present:

$$\begin{aligned} p(\theta_m|I_N) &\propto p(\theta_m) \prod_{k=0}^{N-1} \int_{\mathcal{H}} p(y_k|m,\theta_m,\eta_m,d_k) p(\eta_m|m,\theta_m,I_k)\, d\eta_m \\ &= p(\theta_m) \prod_{k=0}^{N-1} p(y_k|m,\theta_m,I_k,d_k) \\ &\neq p(\theta_m) \prod_{k=0}^{N-1} p(y_k|m,\theta_m,d_k). \end{aligned}$$

Therefore, the *encoder-pooling-emitter* is not applicable and not adopted. [76] further proposed to use long short-term memory networks (LSTM) as the history encoder when conditional independence does not hold. However, similar to the *encoder-pooling-emitter*, the backpropagation time of the LSTM encoder increases quadratically with horizon $N$, and becomes expensive for larger values of $N$.

### 3.2.5 Training details of the policy gradient based vsOED

#### 3.2.5.1 Training of the posterior approximation

Due to the cancellation of intermediate posteriors as shown in Appendix B.7, the variational gradient in Eqn. (3.22) only involves the gradient of the final variational posterior:

$$\begin{aligned} \nabla_\phi \dot{U}(w;\phi) = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \mathbb{E}_{I_N|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \Big[ & \alpha_{\mathcal{M}} \nabla_{\phi_{\mathcal{M}}} \ln q(\dot{m}|I_N;\phi_{\mathcal{M}}) \\ & + \alpha_\Theta \nabla_{\phi_{\Theta_m}} \ln q(\dot{\theta}_m|I_N;\phi_{\Theta_m}) \\ & + \alpha_Z \nabla_{\phi_{Z_m}} \ln q(\dot{z}_m|I_N;\phi_{Z_m}) \Big]. \end{aligned}$$

A Monte Carlo (MC) estimate for the variational gradient is:

$$\begin{aligned} \nabla_\phi \dot{U}(w;\phi) \approx \frac{1}{n_{\text{batch}}} \sum_{i=1}^{n_{\text{batch}}} \Big[ & \alpha_{\mathcal{M}} \nabla_{\phi_{\mathcal{M}}} \ln q(\dot{m}^{(i)}|I_N^{(i)};\phi_{\mathcal{M}}) \\ & + \alpha_\Theta \nabla_{\phi_{\Theta_m}} \ln q(\dot{\theta}_m^{(i)}|I_N^{(i)};\phi_{\Theta_m}) \\ & + \alpha_Z \nabla_{\phi_{Z_m}} \ln q(\dot{z}_m^{(i)}|I_N^{(i)};\phi_{Z_m}) \Big], \end{aligned}$$

where $\dot{m}^{(i)}, \dot{\theta}_m^{(i)}, \dot{\eta}_m^{(i)}, \dot{z}_m^{(i)} \sim p(m, \theta_m, \eta_m, z_m)$ and $I_N^{(i)} \sim p(I_N|\dot{m}^{(i)}, \dot{\theta}_m^{(i)}, \dot{\eta}_m^{(i)}, \pi_w)$, and the gradients can be obtained by, for example, PyTorch Autograd. In our implementation, we draw samples from a replay buffer in order to reduce computations and to enable off-policy learning.

If the IIG formulation is used, the intermediate variational posteriors can be trained in the same manner with their MC gradient estimates:

$$\frac{1}{n_{\text{batch}}} \sum_{i=1}^{n_{\text{batch}}} \Bigg[ \alpha_{\mathcal{M}} \nabla_{\phi_{\mathcal{M}}} \ln q(\dot{m}^{(i)}|I_k^{(i)}; \phi_{\mathcal{M}}) + \alpha_{\Theta} \nabla_{\phi_{\Theta_m}} \ln q(\dot{\theta}_m^{(i)}|I_k^{(i)}; \phi_{\Theta_m}) + \alpha_Z \nabla_{\phi_{Z_m}} \ln q(\dot{z}_m^{(i)}|I_k^{(i)}; \phi_{Z_m}) \Bigg]$$

for $k = 1, \ldots, N-1$. The optimization of the variational posterior approximation is carried out using Adam [80] for all numerical cases. Hyperparameter tuning will be discussed in Sec. 3.2.5.5, and specific hyperparameter settings will be specified for each case.

The variational posterior approximation is updated first during each outer iteration in Algorithm 2.

#### 3.2.5.2 More about the critic

The *action-value function* (i.e. *critic*) $Q_k^{\pi_w}(I_k, d_k)$ quantifies the expected cumulative remaining reward for conducting the $k$th experiment at design $d_k$ and state $I_k$, and thereafter following policy $\pi_w$. The critics formulated with the the one-point reward estimates using either the true posterior or variational posterior are respectively

$$Q_k^{\pi_w}(I_k, d_k) = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m|I_k} \mathbb{E}_{I_N|\pi_w,I_k,d_k,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \left[ \sum_{k=0}^{N-1} \dot{g}_k(I_k, d_k, y_k) + \dot{g}_N(I_N) \right]$$

$$Q_k^{\pi_w}(I_k, d_k; \phi) = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m|I_k} \mathbb{E}_{I_N|\pi_w,I_k,d_k,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \left[ \sum_{k=0}^{N-1} \dot{g}_k(I_k, d_k, y_k; \phi) + \dot{g}_N(I_N; \phi) \right].$$

Specifically, the critics under TIG take the form:

$$Q^{\pi_w}_{T,k}(I_k, d_k) = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m|I_k}\mathbb{E}_{I_N|\pi_w,I_k,d_k,\dot{m},\dot{\theta}_m,\dot{\eta}_m}\Bigg[ \\ \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_N)}{P(\dot{m})} + \alpha_{\Theta} \ln \frac{p(\dot{\theta}_m|I_N)}{p(\dot{\theta}_m)} + \alpha_Z \ln \frac{p(\dot{z}_m|I_N)}{p(\dot{z}_m)} \Bigg]$$

$$Q^{\pi_w}_{T,k}(I_k, d_k; \phi) = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m|I_k}\mathbb{E}_{I_N|\pi_w,I_k,d_k,\dot{m},\dot{\theta}_m,\dot{\eta}_m}\Bigg[ \\ \alpha_{\mathcal{M}} \ln \frac{q(\dot{m}|I_N;\phi_{\mathcal{M}})}{P(\dot{m})} + \alpha_{\Theta} \ln \frac{q(\dot{\theta}_m|I_N;\phi_{\Theta_m})}{p(\dot{\theta}_m)} + \alpha_Z \ln \frac{q(\dot{z}_m|I_N;\phi_{Z_m})}{p(\dot{z}_m)} \Bigg],$$

and the critics under IIG take the form:

$$Q^{\pi_w}_{I,k}(I_k, d_k) = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m|I_k}\mathbb{E}_{I_N|\pi_w,I_k,d_k,\dot{m},\dot{\theta}_m,\dot{\eta}_m}\Bigg[ \\ \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_N)}{P(\dot{m}|I_k)} + \alpha_{\Theta} \ln \frac{p(\dot{\theta}_m|I_N)}{p(\dot{\theta}_m|I_k)} + \alpha_Z \ln \frac{p(\dot{z}_m|I_N)}{p(\dot{z}_m|I_k)} \Bigg]$$

$$Q^{\pi_w}_{I,k}(I_k, d_k; \phi) = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m|I_k}\mathbb{E}_{I_N|\pi_w,I_k,d_k,\dot{m},\dot{\theta}_m,\dot{\eta}_m}\Bigg[ \\ \alpha_{\mathcal{M}} \ln \frac{q(\dot{m}|I_N;\phi_{\mathcal{M}})}{q(\dot{m}|I_k;\phi_{\mathcal{M}})} + \alpha_{\Theta} \ln \frac{q(\dot{\theta}_m|I_N;\phi_{\Theta_m})}{q(\dot{\theta}_m|I_k;\phi_{\Theta_m})} + \alpha_Z \ln \frac{q(\dot{z}_m|I_N;\phi_{Z_m})}{q(\dot{z}_m|I_k;\phi_{Z_m})} \Bigg].$$

**Remark 1:** The difference between the one-point-TIG and -IIG critics is

$$Q^{\pi_w}_{T,k}(I_k, d_k) - Q^{\pi_w}_{I,k}(I_k, d_k) = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m|I_k}\mathbb{E}_{I_N|\pi_w,I_k,d_k,\dot{m},\dot{\theta}_m,\dot{\eta}_m}\Bigg[ \\ \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_k)}{P(\dot{m})} + \alpha_{\Theta} \ln \frac{p(\dot{\theta}_m|I_k)}{p(\dot{\theta}_m)} + \alpha_Z \ln \frac{p(\dot{z}_m|I_k)}{p(\dot{z}_m)} \Bigg] \\ = \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m|I_k}\Bigg[ \\ \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_k)}{P(\dot{m})} + \alpha_{\Theta} \ln \frac{p(\dot{\theta}_m|I_k)}{p(\dot{\theta}_m)} + \alpha_Z \ln \frac{p(\dot{z}_m|I_k)}{p(\dot{z}_m)} \Bigg],$$

which is constant with respect to $d_k$. Since $\nabla_{d_k} Q^{\pi_w}_k(I_k, d_k)$ is used in the policy gradient (Eqn. (3.23)), then whether adopting $Q^{\pi_w}_{T,k}(I_k, d_k)$ or $Q^{\pi_w}_{I,k}(I_k, d_k)$ will result in the same policy gradient value.

**Remark 2:** The difference between $Q^{\pi_w}_{T,k}(I_k, d_k)$ and $Q^{\pi_w}_{T,k}(I_k, d_k; \phi)$ is

$$
\begin{aligned}
&Q^{\pi_w}_{T,k}(I_k, d_k) - Q^{\pi_w}_{T,k}(I_k, d_k; \phi) \\
&= \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m|I_k} \mathbb{E}_{I_N|\pi_w,I_k,d_k,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \Bigg[ \\
&\qquad \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_N)}{q(\dot{m}|I_N;\phi_{\mathcal{M}})} + \alpha_{\Theta} \ln \frac{p(\dot{\theta}_m|I_N)}{q(\dot{\theta}_m|I_N;\phi_{\Theta_m})} + \alpha_Z \ln \frac{p(\dot{z}_m|I_N)}{q(\dot{z}_m|I_N;\phi_{Z_m})} \Bigg] \\
&= \mathbb{E}_{I_N|\pi_w,I_k,d_k} \Bigg[ \alpha_{\mathcal{M}} D_{\mathrm{KL}} \left( P(m|I_N) \,||\, q(m|I_N;\phi_{\mathcal{M}}) \right) \\
&\qquad + \mathbb{E}_{m|I_N} \left[ \alpha_{\Theta} D_{\mathrm{KL}} \left( p(\theta_m|I_N) \,||\, q(\theta_m|I_N;\phi_{\Theta_m}) \right) + \alpha_Z D_{\mathrm{KL}} \left( p(z_m|I_N) \,||\, q(z_m|I_N;\phi_{Z_m}) \right) \right] \Bigg],
\end{aligned}
$$

which is an expected weighted KL divergence (with positive weights). It equals to zero if and only if the variational posterior approximations $q(\cdot|I_N;\phi_{(\cdot)})$ are equal to the true posteriors $p(\cdot|I_N)$ (except the trivial case when $\alpha_{\mathcal{M}} = \alpha_{\Theta} = \alpha_Z = 0$). In other words, $Q^{\pi_w}_{T,k}(I_k, d_k; \phi)$ forms a lower bound of $Q^{\pi_w}_{T,k}(I_k, d_k)$, and learning an accurate variational posterior approximation could also help reduce the bias in the critic.

**Remark 3:** The difference between $Q^{\pi_w}_{I,k}(I_k, d_k)$ and $Q^{\pi_w}_{I,k}(I_k, d_k; \phi)$ is

$$
\begin{aligned}
&Q^{\pi_w}_{I,k}(I_k, d_k) - Q^{\pi_w}_{I,k}(I_k, d_k; \phi) \\
&= \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m|I_k} \mathbb{E}_{I_N|\pi_w,I_k,d_k,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \Bigg[ \\
&\qquad \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_N)}{q(\dot{m}|I_N;\phi_{\mathcal{M}})} + \alpha_{\Theta} \ln \frac{p(\dot{\theta}_m|I_N)}{q(\dot{\theta}_m|I_N;\phi_{\Theta_m})} + \alpha_Z \ln \frac{p(\dot{z}_m|I_N)}{q(\dot{z}_m|I_N;\phi_{Z_m})} \\
&\qquad - \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_k)}{q(\dot{m}|I_k;\phi_{\mathcal{M}})} - \alpha_{\Theta} \ln \frac{p(\dot{\theta}_m|I_k)}{q(\dot{\theta}_m|I_k;\phi_{\Theta_m})} - \alpha_Z \ln \frac{p(\dot{z}_m|I_k)}{q(\dot{z}_m|I_k;\phi_{Z_m})} \Bigg] \\
&= \mathbb{E}_{I_N|\pi_w,I_k,d_k} \Bigg[ \alpha_{\mathcal{M}} D_{\mathrm{KL}} \left( P(m|I_N) \,||\, q(m|I_N;\phi_{\mathcal{M}}) \right) \\
&\qquad + \mathbb{E}_{m|I_N} \left[ \alpha_{\Theta} D_{\mathrm{KL}} \left( p(\theta_m|I_N) \,||\, q(\theta_m|I_N;\phi_{\Theta_m}) \right) + \alpha_Z D_{\mathrm{KL}} \left( p(z_m|I_N) \,||\, q(z_m|I_N;\phi_{Z_m}) \right) \right] \Bigg] \\
&\qquad - \alpha_{\mathcal{M}} D_{\mathrm{KL}} \left( P(m|I_k) \,||\, q(m|I_k;\phi_{\mathcal{M}}) \right) \\
&\qquad - \mathbb{E}_{m|I_k} \left[ \alpha_{\Theta} D_{\mathrm{KL}} \left( p(\theta_m|I_k) \,||\, q(\theta_m|I_k;\phi_{\Theta_m}) \right) + \alpha_Z D_{\mathrm{KL}} \left( p(z_m|I_k) \,||\, q(z_m|I_k;\phi_{Z_m}) \right) \right].
\end{aligned}
$$

Applying the triangle inequality, the difference is bounded by

$$
\begin{aligned}
&\left|Q_{I,k}^{\pi_w}(I_k, d_k) - Q_{I,k}^{\pi_w}(I_k, d_k; \phi)\right| \\
&\leq \mathbb{E}_{I_N|\pi_w, I_k, d_k}\Bigg[\alpha_{\mathcal{M}} D_{\mathrm{KL}}\left(\, P(m|I_N) \,||\, q(m|I_N; \phi_{\mathcal{M}})\,\right) \\
&\quad + \mathbb{E}_{m|I_N}\left[\alpha_{\Theta} D_{\mathrm{KL}}\left(\, p(\theta_m|I_N) \,||\, q(\theta_m|I_N; \phi_{\Theta_m})\,\right) + \alpha_Z D_{\mathrm{KL}}\left(\, p(z_m|I_N) \,||\, q(z_m|I_N; \phi_{Z_m})\,\right)\right]\Bigg] \\
&\quad + \alpha_{\mathcal{M}} D_{\mathrm{KL}}\left(\, P(m|I_k) \,||\, q(m|I_k; \phi_{\mathcal{M}})\,\right) \\
&\quad + \mathbb{E}_{m|I_k}\left[\alpha_{\Theta} D_{\mathrm{KL}}\left(\, p(\theta_m|I_k) \,||\, q(\theta_m|I_k; \phi_{\Theta_m})\,\right) + \alpha_Z D_{\mathrm{KL}}\left(\, p(z_m|I_k) \,||\, q(z_m|I_k; \phi_{Z_m})\,\right)\right].
\end{aligned}
$$

Therefore, the bias of the critic is contributed from both the bias of the final variational posterior $q(\cdot|I_N; \phi_{(\cdot)})$ and the bias of the intermediate variational posteriors $q(\cdot|I_k; \phi_{(\cdot)})$ for $k = 1, \ldots, N-1$. It equals to zero if and only if all the variational posteriors are equal to their corresponding true posteriors (except the trivial case when $\alpha_{\mathcal{M}} = \alpha_{\Theta} = \alpha_Z = 0$).

The critic network (parameterized by $\nu$) is updated after the update of the variational posterior approximation and the update of the *one-point IG* $\dot{g}_k(I_k, d_k, y_k; \phi)$ and $\dot{g}_N(I_N; \phi)$ during each outer iteration in Algorithm 2. It can be learned in a supervised learning manner by minimizing the loss function:

$$
\mathcal{L}(\nu) = \frac{1}{n_{\mathrm{batch}}} \sum_{i=1}^{n_{\mathrm{batch}}} \sum_{k=0}^{N-1} \left[Q_{k,\nu}^{\pi_w}(I_k^{(i)}, d_k^{(i)}) - \left(\dot{g}_k(I_k^{(i)}, d_k^{(i)}, y_k^{(i)}) + \gamma Q_{k+1,\nu}^{\pi_w}(I_{k+1}^{(i)}, d_{k+1}^{(i)})\right)\right]^2,
$$

where $\gamma \in [0, 1]$ is a discount factor used for regularization, and batch samples are drawn from the replay buffer. When the TIG formulation is used, all the stage rewards $\dot{g}_k$ for $k = 0, \ldots, N-1$ are 0 if there are no non-IG immediate rewards. In that case, the training of the critic network at early stages will be slow and may even lead to numerical divergence of the policy gradient when horizon $N$ is long. Therefore, we utilize the idea of REINFORCE [153] and modify the loss function for TIG to

$$
\begin{aligned}
\mathcal{L}(\nu) = \frac{1}{n_{\mathrm{batch}}} \sum_{i=1}^{n_{\mathrm{batch}}} \sum_{k=0}^{N-1} \Bigg[ & Q_{k,\nu}^{\pi_w}(I_k^{(i)}, d_k^{(i)}) \\
& - \psi\left(\dot{g}_k(I_k^{(i)}, d_k^{(i)}, y_k^{(i)}) + \gamma Q_{k+1,\nu}^{\pi_w}(I_{k+1}^{(i)}, d_{k+1}^{(i)})\right) \\
& - (1-\psi)\left(\sum_{t=k}^{N-1} \gamma^{t-k} \dot{g}_t(I_t^{(i)}, d_t^{(i)}, y_t^{(i)}) + \gamma^{N-k} \dot{g}_N(I_N^{(i)})\right)\Bigg]^2,
\end{aligned}
$$

where $\psi$ linearly increases from 0 to 1 during the training process. Moreover, the target network in

[91] is also utilized to enable off-policy learning with the update rate of the target network set to 0.1 across all numerical cases. The optimization of the critic network is carried out using Adam for all numerical cases.

#### 3.2.5.3 Numerical estimation of the policy gradient

The actor network is updated last during each outer iteration in Algorithm 2. The MC estimator of the policy gradient (Eqn. (3.23)) is

$$\nabla_w \dot{U}(w;\phi) \approx \frac{1}{n_{\text{batch}}} \sum_{i=1}^{n_{\text{batch}}} \sum_{k=0}^{N-1} \nabla_w \mu_{k,w}(I_k^{(i)}) \nabla_{d_k^{(i)}} Q_{k,\nu}^{\pi_w}(I_k^{(i)}, d_k^{(i)})\Big|_{d_k^{(i)}=\mu_{k,w}(I_k^{(i)})}.$$

In our implementation, we draw batch samples from a replay buffer. The optimization of the actor network is performed with Adam for all numerical cases.

#### 3.2.5.4 Exploration versus exploitation

To promote better exploration during the optimization process, the same exploration strategy as Sec. 2.2.2.4 is utilized by adding perturbation to the deterministic policy:

$$d_k = \mu_k(I_k) + \epsilon_{\text{explore}},$$

where $\epsilon_{\text{explore}}$ follows a zero-mean multivariate Gaussian distribution with a diagonal covariance. The covariance diagonal terms reflect the exploration length scale for each dimension of $d_k$. If perturbing to outside the feasible design region $\mathcal{D}_k$, it will be moved back to the closest feasible point. A good balance of exploration and exploitation is important for the numerical algorithm to find a good policy. Insufficient exploration limits the understanding of the environment, while too much exploration (i.e. insufficient exploitation) may lead to slow convergence. A reasonable strategy is to set a larger exploration in the early stages of training and then gradually reduce it. Details of these exploration settings will be specified for each numerical case in Sec. 3.3.

We emphasize that the policy exploration is employed solely during the training phase. During evaluation (testing), the deterministic policy is still utilized without additional exploration.

#### 3.2.5.5 Hyperparameter tuning

Since vsOED is trained with limited budgets in this chapter, our main strategy for hyperparameter tuning is to start with a relatively large hyperparameter value and gradually decrease it.

For the optimization of the model posterior predictor and the parameter posterior predictor with GMM and NF, we start with the initial learning rate $10^{-3}$ and an exponential learning rate decay

rate 0.9999.

For the optimization of the critic network, an initial learning rate of $10^{-3}$ and a learning rate decay rate of 0.9999 are used across all numerical cases. Both the variational approximation and the critic network are updated 5 steps (i.e. applying gradient ascent steps 5 times) within each outer iteration. It is important to note that updating the variational approximation and the critic network too many steps in each outer iteration may result in overestimation of the value function and adversely affect the policy search [66].

For the optimization of the actor network, a learning rate decay rate of 0.9999 is used. However, the choice of the initial learning rate is more problem-dependent. Typically, we start with an initial learning rate of $10^{-3}$ and gradually decrease it to $5\times10^{-4}$ or $2\times10^{-4}$ if divergence occurs. For IIG formulation, an initial learning rate of $10^{-3}$ works well. However, for TIG formulation, a smaller actor learning rate is required. This is because the learning of the critic in TIG is slower, and a large actor learning rate can more easily induce divergence in the early stages of training.

For other hyperparameters, including the number of updates $n_{\text{update}}$, the number of new MC episodes $n_{\text{episode}}$, the batch size $n_{\text{batch}}$, the replay buffer size $n_{\text{buffer}}$, a number of combinations are tested to select the optimal combination and their values are specified for each numerical case in Sec. 3.3.

## 3.3 Numerical results and discussions

### 3.3.1 Assessment setup

#### 3.3.1.1 Baseline algorithms for comparison

We validate and compare the performance of **vsOED** on a number of numerical experiments against baselines involving various *real-time* and *adaptive* algorithms, listed as follows:

- **Random.** For Random design, a design is sampled uniformly within $\mathcal{D}$.
- **DAD.** For DAD [55], we use the code available at: https://github.com/ae-foster/dad. The default setup is used as the full training; for testing under limited budgets, different combinations of hyperparameters are tested to select the optimal one.
- **iDAD.** For iDAD [76], we use the code available at: https://github.com/desi-ivanova/idad. The default setup is used as the full training but the learning rate is set as 0.0002; for testing under limited budgets, different combinations of hyperparameters are tested to select the optimal one.

- **RL.** For RL [14], we use the code available at: https://github.com/csiro-mlai/RL-BOED. The default setup is used as the full training; for testing under limited budgets, different combinations of hyperparameters are tested to select the optimal one.

The key properties of methods compared in this chapter are summarized in Table 3.10.

Table 3.10: Properties of different methods.

| | Real-time | Adaptive | Implicit | No model derivative | Multi-model |
|---|---|---|---|---|---|
| Random | ✓ | ✗ | ✓ | ✓ | ✓ |
| DAD | ✓ | ✓ | ✗ | ✗ | ✗ |
| iDAD | ✓ | ✓ | ✓ | ✗ | ✗ |
| RL | ✓ | ✓ | ✗ | ✓ | ✗ |
| **vsOED** | ✓ | ✓ | ✓ | ✓ | ✓ |

For **vsOED**, we employ GMMs and NFs for posterior approximation, and TIG and IIG reward formulations. We use the naming convention where, for example, **vsOED-G-I** stands for **G**MM with **I**IG, and **vsOED-N-T** for **N**F with **T**IG. Baseline methods include **Random** design, **DAD** [55], **iDAD** [76], and **RL** that employed advanced RL techniques [14]. **RL** can also be combined with both TIG and IIG, denoted by **RL-T** and **RL-I** respectively. **DAD** and **iDAD** require the derivative of the forward model, **vsOED** and **iDAD** can accommodate implicit likelihoods, and only **vsOED** can handle multi-model scenarios and model discrimination OED.

All experiments are implemented in Python using PyTorch. Truncated normal distribution is not naturally supported by PyTorch, we use the code from: https://github.com/toshas/torch_truncnorm. The solver of the SIR model is from: https://github.com/desi-ivanova/idad.

All experiments are run on the Great Lakes Slurm HPC Cluster nodes: https://arc.umich.edu/greatlakes/configuration/, each node is equipped with a single Nvidia Tesla A40 or V100 GPU.

#### 3.3.1.2 Prior contrastive estimation

In order to maintain a consistent comparison platform, we will always use prior contrastive estimation (PCE) [55, 14, 57] to estimate the expected utility of different trained policies when possible.

When nuisance parameters $\eta$ are not present, the reward corresponding to PoIs under the *one-*

*point estimate* formulation can also be estimated by

$$\ln \frac{p(\dot{\theta}_m|I_{k_2})}{p(\dot{\theta}_m|I_{k_1})} = \ln \frac{p(I_{k_2}|\dot{m},\dot{\theta}_m)p(I_{k_1})}{p(I_{k_1}|\dot{m},\dot{\theta}_m)p(I_{k_2})} \approx \ln \frac{p(I_{k_2}|\dot{m},\dot{\theta}_m)\frac{1}{L+1}\left[p(I_{k_1}|\dot{m},\dot{\theta}_m) + \sum_{l=1}^{L_m} p(I_{k_1}|\dot{m},\theta_{m,l})\right]}{p(I_{k_1}|\dot{m},\dot{\theta}_m)\frac{1}{L+1}\left[p(I_{k_2}|\dot{m},\dot{\theta}_m) + \sum_{l=1}^{L_m} p(I_{k_2}|\dot{m},\theta_{m,l})\right]},$$

where the first equality follows Bayes' rule, $\theta_{m,l} \sim p(\theta_m)$ are $L_m$ contrastive i.i.d. samples, which are shared in the numerator and denominator in the last equation, and $0 \le k_1 < k_2 \le N$. When $k_1 = 0$ and $k_2 = N$, it is the estimate for the terminal *one-point-IG*, and when $k_1 = k$ and $k_2 = k+1$, it is the estimate for the incremental *one-point-IG* at the $k$th stage. PCE forms a lower bound of the EIG, and the bound becomes tight as $L_m \to \infty$ [55]. Note that PCE cannot be practically applied when $\eta$ is present, since evaluating the likelihood $p(I_k|\dot{m},\theta_{m,l})$ would require yet another loop to marginalize out $\eta_m$ and is extremely expensive. Similarly, PCE also cannot be practically applied for estimating the reward of IG on the goal-oriented predictive QoIs, which requires another loop to marginalize out $\theta_m$ and $\eta_m$.

The reward of IG on model probability using the *one-point estimate* can also be rewritten as

$$\ln \frac{P(\dot{m}|I_{k_2})}{P(\dot{m}|I_{k_1})} = \ln \frac{p(I_{k_2}|\dot{m})p(I_{k_1})}{p(I_{k_1}|\dot{m})p(I_{k_2})} = \ln \frac{p(I_{k_2}|\dot{m})\sum_{m=1}^{\mathcal{M}} P(m)p(I_{k_1}|m)}{p(I_{k_1}|\dot{m})\sum_{m=1}^{\mathcal{M}} P(m)p(I_{k_2}|m)},$$

where each marginalized likelihood $p(I_k|m)$ may be estimated by

$$p(I_k|m) = \int_{\Theta,H} p(\theta_m,\eta_m|m)p(I_k|m,\theta_m,\eta_m)\,d\theta_m\,d\eta_m \approx \frac{1}{L_m}\sum_{l=1}^{L_m} p(I_k|m,\theta_{m,l},\eta_{m,l}),$$

with $\theta_{m,l}$ and $\eta_{m,l}$ being samples independently drawn from the joint prior $p(\theta_m,\eta_m|m)$. To ensure an accurate estimate during the evaluation stage to compare the policies found by different methods, we use a large sample size $L_m \approx \frac{10^6}{\mathcal{M}}$.

For cases having nuisance parameters or predictive QoIs (i.e. $\alpha_Z > 0$), other measures are reported.

### 3.3.2 Source location finding

We adapt the source location finding problem from [55]. In this numerical case, we enlist $\mathcal{M} = 3$ candidate models with uniform model prior (i.e. $P(m) = 1/3$). For the $m$th model ($m \in \{1, 2, 3\}$), there are $m$ sources randomly located in a 2D domain, each emitting a signal that decays inversely with the square of the distance. The PoIs are the source locations $\theta_m = \{\theta_{m,1}, \ldots, \theta_{m,m}\}$ where $\theta_{m,i} \in \mathbb{R}^2$ denotes the location of $i$th source. The total intensity at a given location $d$, aggregated from the $m$ sources, is then

$$\mu(m, \theta_m, d) = \epsilon_{bg} + \sum_{i=1}^{m} \frac{1}{\epsilon_{max} + \left\| \theta_{m,i} - d \right\|^2},$$

where $\epsilon_{bg} = 10^{-1}$ is the background signal, and $\epsilon_{max} = 10^{-4}$ is the maximum signal. The experimental observation is noise-corrupted signals, and the likelihood follows

$$\log y | m, \theta_m, d \sim \mathcal{N}(\log \mu(m, \theta_m, d), \sigma^2),$$

where the noise standard deviation is $\sigma = 0.5$. The prior is

$$\theta_{m,i} \sim \mathcal{N}(0, I).$$

The design is for finding the optimal sequence of observation locations to maximize the expected utility, and the design space is restricted to $\mathcal{D}_k = [-4, 4]^2$.

We are also interested in a goal-oriented OED situation that involves the flux along the $x$ spatial direction, which can be computed using Fick's law:

$$f(m, \theta_m, d) = -D \frac{\partial \mu(m, \theta_m, d)}{\partial x},$$

where $f$ represents the flux, and $D = 1$ is the diffusivity. More specifically, we consider the flux integrated over an infinite vertical wall located at $x = 6$, which yields

$$\begin{aligned} J(m, \theta_m) &= \int_{y=-\infty}^{+\infty} f(m, \theta_m, (6, y))\, dy \\ &= \sum_{i=1}^{m} -\frac{\pi(\theta_{m,i,x} - 6)}{\left(\epsilon_{max} + (\theta_{m,i,x} - 6)^2\right)^{3/2}}. \end{aligned}$$

The final goal-oriented QoI is the log flux magnitude $z_m = \log |J(m, \theta_m)|$. It is worth noting that the flux is only depending on the $x$-position of the source (i.e. $\theta_{m,i,x}$). Since the explicit form of the prior-predictive for the QoI is not analytically available, we elect to omit it from the computations

and it does not affect the optimal policy per Appendix B.5. Below we first show a uni-model case where $m$ is fixed at 2 and so $\alpha_{\mathcal{M}} = 0$, and then present the multi-model case.

#### 3.3.2.1 Uni-model example

In this uni-model example, $m$ is fixed at 2.

**Hyperparameters.** The hyperparameters for the uni-model source location finding problem are listed in Table 3.11, where vsOED-G-I stands for **G**MM with **I**IG, and vsOED-N-T for **N**F with **T**IG, etc. For the linear mapping in the output layer of the GMM net, we transform the output of the GMM mean net of the PoI posterior predictor to a range of $[-6, 6]$, the output of the GMM standard deviation net of the PoI posterior predictor to a range of $[10^{-5}, 1]$, the output of the GMM mean net of the QoI posterior predictor to a range of $[-6, 6]$, and the output of the GMM standard deviation net of the QoI posterior predictor to a range of $[10^{-5}, 2]$. The truncated normal distribution is not used in this example.

Table 3.11: Hyperparameters of the uni-model source location finding problem. In the table, "lr" means "learning rate".

| | vsOED-G-T | vsOED-G-I | vsOED-F-T | vsOED-F-I |
|---|---|---|---|---|
| #training iteration $n_{\text{update}}$ | 10001 | 10001 | 10001 | 10001 |
| #new episodes per iteration $n_{\text{episode}}$ | 1000 | 1000 | 1000 | 1000 |
| batch size $n_{\text{batch}}$ | 10000 | 10000 | 10000 | 10000 |
| parameter predictor initial lr | $10^{-3}$ | $10^{-3}$ | $10^{-3}$ | $10^{-3}$ |
| parameter predictor lr decay | 0.9999 | 0.9999 | 0.9999 | 0.9999 |
| #param predictor update per iteration | 5 | 5 | 5 | 5 |
| $n_{\text{mixture}}$ | 8 | 8 | N/A | N/A |
| $n_{\text{trans}}$ | N/A | N/A | 4 | 4 |
| initial actor lr | $5 \times 10^{-4}$ | $10^{-3}$ | $10^{-3}$ | $10^{-3}$ |
| actor lr decay | 0.9999 | 0.9999 | 0.9999 | 0.9999 |
| initial critic lr | $10^{-3}$ | $10^{-3}$ | $10^{-3}$ | $10^{-3}$ |
| critic lr decay | 0.9999 | 0.9999 | 0.9999 | 0.9999 |
| max buffer size | $10^6$ | $10^6$ | $10^6$ | $10^6$ |
| discount factor $\gamma$ | 1 | 0.9 | 1 | 0.9 |
| initial design noise scale | 0.5 | 0.5 | 0.5 | 0.5 |
| design noise scale decay | 0.9999 | 0.9999 | 0.9999 | 0.9999 |
| target network lr | 0.1 | 0.1 | 0.1 | 0.1 |

**Calculation of the goal-oriented posterior.** The true posterior-predictive PDF of the goal-oriented QoI $z$ shown in Fig. 3.4 are calculated on a discretized grid, where the prior $p(z)$ is obtained using kernel density estimation (KDE) [127] with $10^6$ prior $z$ samples, and the likelihood

$p(I_N|z,\pi)$ is obtained with approximate Bayesian computation (ABC) [39]:

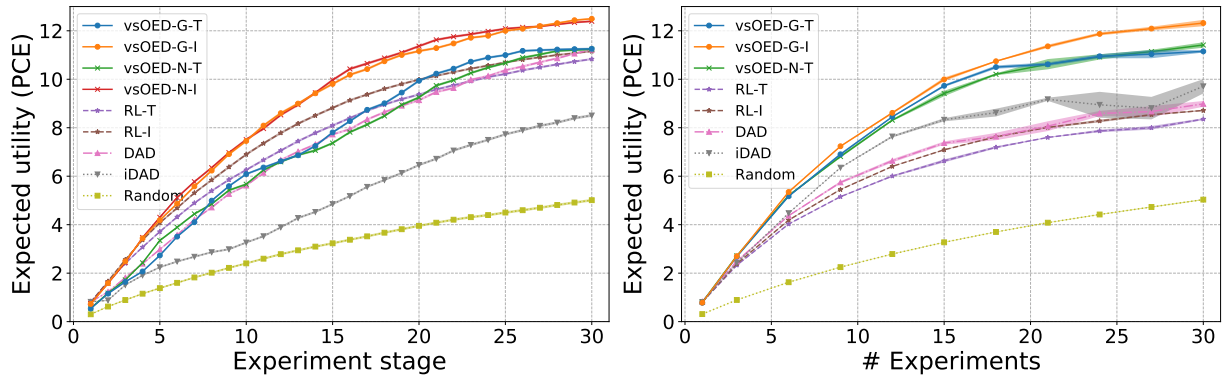
$$p(I_N|z,\pi) \approx \frac{1}{n_{\text{accept}}} \sum_{i=1}^{L} p(I_N|\theta^{(i)},\pi)\mathbf{1}_{|H(\theta^{(i)})-z|<=\epsilon},$$

where $\theta^{(i)} \sim p(\theta)$, $\mathbf{1}$ is an indicator function and $n_{\text{accept}}$ is the number of $\theta$ samples that satisfy the indicator function. We use $L = 10^6$ samples and acceptance tolerance $\epsilon = 10^{-3}$ to ensure accuracy.

**PoI inference OED.** The first case is a pure PoI (i.e. source locations) inference OED where $\alpha_\Theta = 1$ and $\alpha_Z = 0$; this is identical to the setup in previous literature [55, 76, 14]. Figure 3.1a presents the expected cumulative utilities at various experiment stages, where all policies are optimized for a design horizon of $N = 30$ experiments and then evaluated on the various intermediate experiment stages. We restrict vsOED training to a total budget of 10 million episode samples, while fully training RL, DAD and iDAD using their default publication settings (RL 8 trillion episodes, DAD 100 billion episodes, and iDAD 200 million episodes). In this plot, vsOED with IIG achieves noticeably better expected utilities compared to TIG, while TIG still reaches similar performance as other fully trained baselines as they all produce comparable expected utilities at $N = 30$. The lower values for vsOED with TIG in the earlier stages suggest that the policy sacrifices short-term rewards for a higher total expected utility.

Figure 3.1b presents the expected utility versus design horizon $N$, where at each $N$ the data point reflects a new policy optimized specifically for that horizon. We apply an equal computational budget of 10 million episodes for all methods. In this comparison, vsOED outperforms other baselines across all $N$. IIG again achieves better performance than TIG, especially for $N > 15$. The shaded regions in both Fig. 3.1a and 3.1b illustrate the robustness of vsOED, RL, and DAD training against random seeds, while iDAD exhibits some instability for longer horizons. Lastly, Fig. 3.2 provides a validation example showing that GMM can effectively approximate the posterior even when highly non-Gaussian.

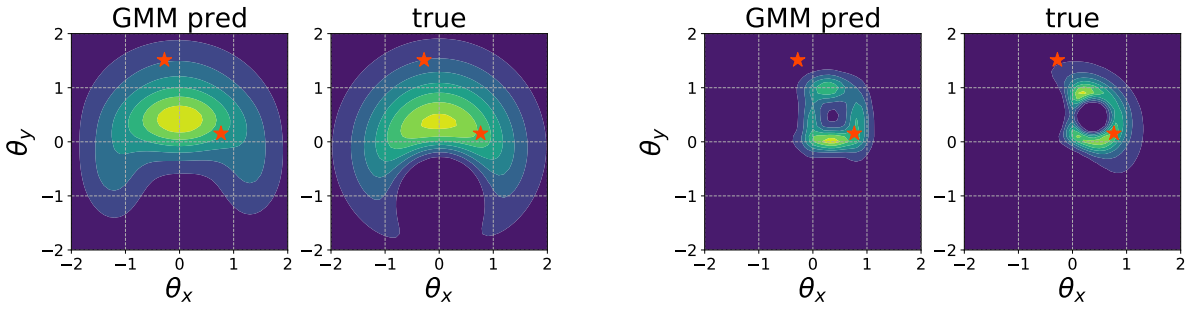
**QoI goal-oriented OED.** The second case is a pure goal-oriented OED where $\alpha_\Theta = 0$ and $\alpha_Z = 1$. The scenario is that the source is emitting a harmful contaminant posing a risk of populated area to the right of the domain. Our predictive QoI $z$ is the contaminant flux integrated on the infinite vertical wall located at $x = 6$. Figure 3.3 contrasts the behavior of the PoI OED policy and the goal-oriented OED policy, where the former adjusts toward the estimated source locations while the latter forms a roughly vertical design pattern. This behavior can be explained from physical principles: since the flux is integrated over the $y$-coordinate, it is solely dependent on the $x$-position of the source (i.e. $\theta_x$) (Sec. 3.3.2). Spreading measurements along a vertical is more sensitive at detecting changes in $\theta_x$ due to the isotropic nature of the source emission—this is supported by Fig. 3.4a showing the greater posterior shrinkage from a simple vertical sensor design




(a) Expected cumulative utility versus experiment stage

(b) Expected utility versus design horizon

Figure 3.1: Expected utilities of various OED methods, all estimated using PCE with $L = 10^6$. (a) Mean and standard error (shaded) from 2000 evaluation episodes. (b) Mean and standard error (shaded) of 4 replicates with different random seeds, each replicate evaluated with 2000 episodes.

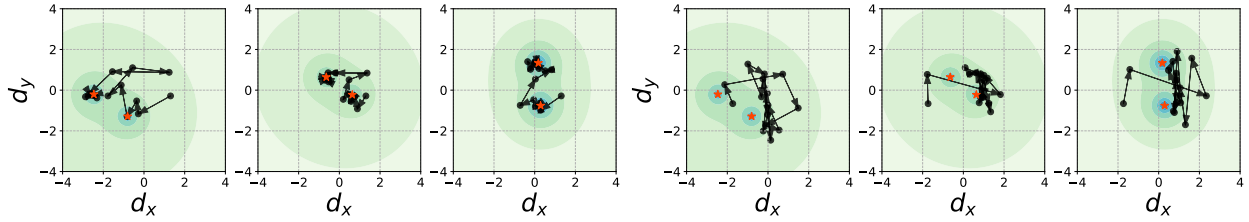



(a) $N = 1$

(b) $N = 3$

Figure 3.2: GMM posterior of PoIs versus their true posterior. Red stars are the true source locations.

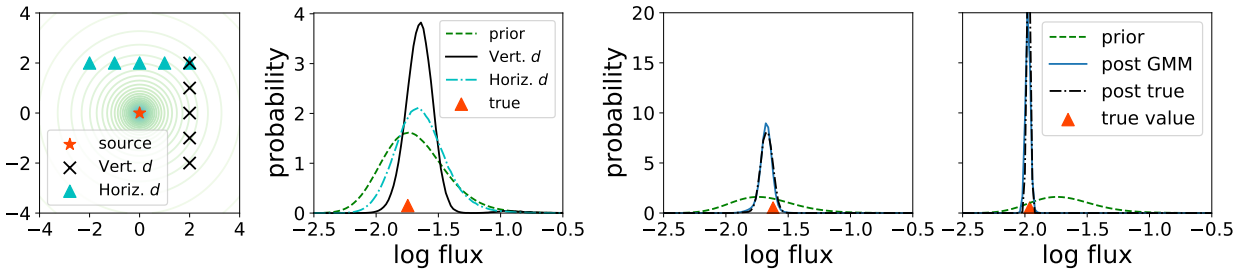
over a simple horizontal sensor design. Figure 3.4b demonstrates that the GMM again successfully approximates the posterior-predictive of $z$.




(a) PoI inference OED policy trajectories

(b) QoI goal-oriented OED policy trajectories

Figure 3.3: Policies for $N = 15$. The contour background illustrates the signal strength.

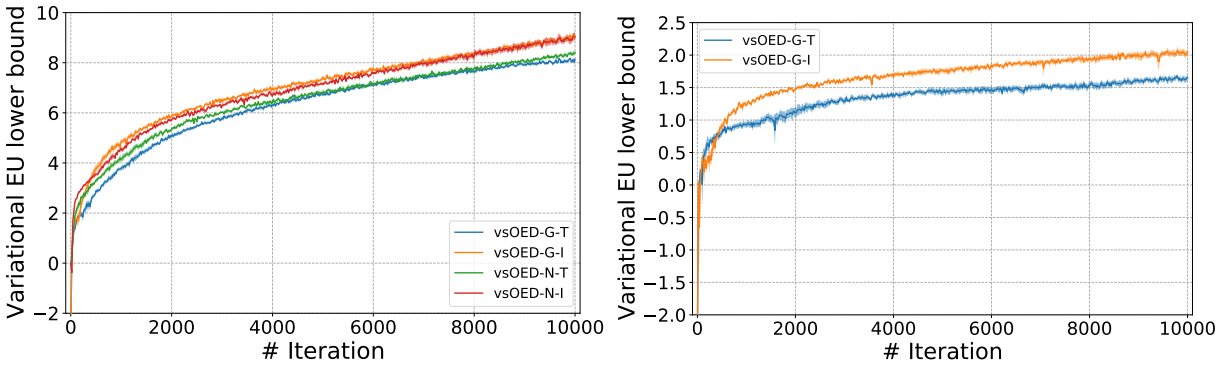



(a) Vertical $d$ versus horizontal $d$

(b) Posterior-predictive at $N = 15$

Figure 3.4: QoI posterior predictive comparisons for the goal-oriented OED.

**Training stability.** Figure 3.5 shows the training histories of the PoI inference OED ($\alpha_\Theta = 1$) and the QoI goal-oriented OED ($\alpha_Z = 1$), where the solid line and the shaded region are the mean and standard error of 4 replicates with different random seeds. The training of vsOED appears highly robust against randomization. Table 3.12 presents the PCE evaluation of optimal policies from 4 replicates of PoI inference OED, optimized for horizon $N = 30$. Each element in the table represents the mean and standard error computed from 2000 samples. Table 3.13 further provides the mean and standard error aggregated from the means of these 4 replicates. These tables further demonstrate the robustness of vsOED. Moreover, we observe that vsOED with NF achieves slightly better performance compared to using GMM. This can be attributed to the increased expressiveness of NFs. Figure 3.6 provides insights into the expressiveness of NFs, from which we can find that when the posterior is highly non-Gaussian, NF outperforms GMM in approximating the posterior. However, as the horizon $N$ increases, the posterior tends toward sharper multi-modal Gaussian mixtures, which explains the similar performance of using GMM and NFs at horizon $N = 30$. Figure 3.7 draws the variational expected utility lower bounds against the number of experiments for the goal-oriented OED. It also demonstrates the robustness of vsOED.




(a) Inference OED

(b) Goal-oriented OED

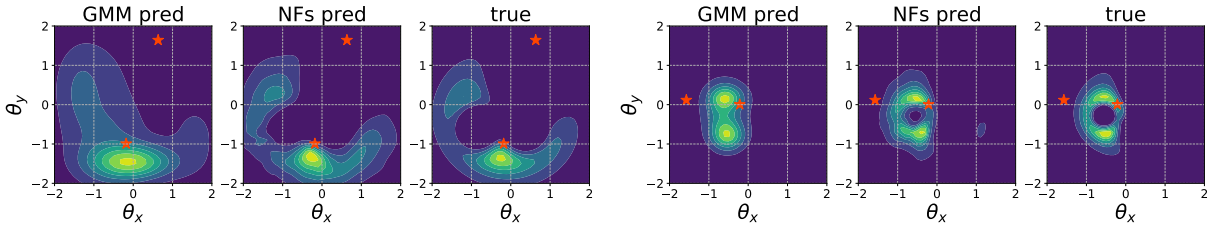
Figure 3.5: Training histories of PoI inference OED and QoI goal-oriented OED for the uni-model source location finding problem, optimized for horizon $N = 30$. The solid line and the shaded region are the mean and standard error of 4 replicates with different random seeds.

Table 3.12: PCE evaluation of optimal policies from 4 replicates of PoI inference OED for the uni-model source location finding problem, optimized for horizon $N = 30$.

| | Run 1 | Run 2 | Run 3 | Run 4 |
|---|---|---|---|---|
| vsOED-G-T | 11.257 ± 0.046 | 11.089 ± 0.046 | 11.237 ± 0.046 | 11.022 ± 0.044 |
| vsOED-G-I | 12.496 ± 0.040 | 12.306 ± 0.043 | 12.577 ± 0.040 | 11.895 ± 0.044 |
| vsOED-F-T | 11.239 ± 0.046 | 11.674 ± 0.045 | 11.149 ± 0.046 | 11.592 ± 0.045 |
| vsOED-F-I | 12.393 ± 0.042 | 12.342 ± 0.042 | 12.155 ± 0.042 | 12.536 ± 0.043 |

Table 3.13: Aggregated PCE evaluation results of optimal policies from 4 replicates of PoI inference OED for the uni-model source location finding problem, optimized for horizon $N = 30$.

| | Mean | SE |
|---|---|---|
| vsOED-G-T | 11.151 | 0.049 |
| vsOED-G-I | 12.319 | 0.132 |
| vsOED-F-T | 11.414 | 0.112 |
| vsOED-F-I | 12.357 | 0.068 |




(a) Example posterior 1

(b) Example posterior 2

Figure 3.6: Examples of GMM posterior, NF posterior, and true posterior at horizon $N = 3$. Red stars are the true source locations.

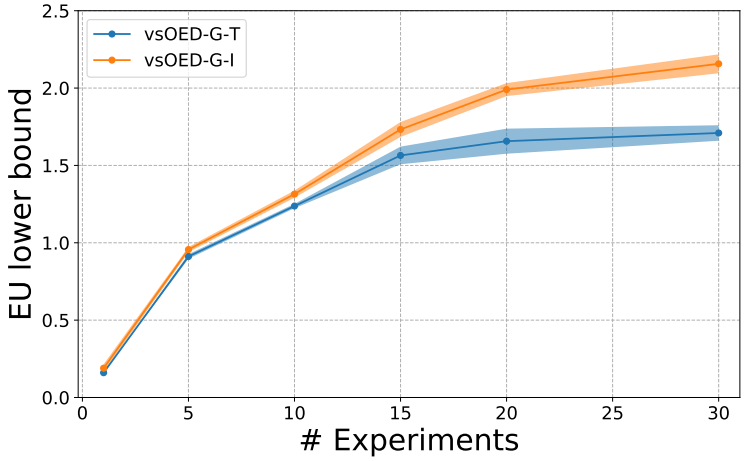



Figure 3.7: Variational expected utility lower bounds of goal-oriented OED for the uni-model source location finding problem. The solid line and the shaded region are the mean and standard error of 4 replicates with different random seeds.

#### 3.3.2.2 Multi-model example

For the multi-model source location finding problem, 5 scenarios will be considered: model discrimination OED ($\alpha_{\mathcal{M}} = 1$, $\alpha_{\Theta} = \alpha_Z = 0$), inference OED ($\alpha_{\Theta} = 1$, $\alpha_{\mathcal{M}} = \alpha_Z = 0$), goal-oriented OED ($\alpha_Z = 1$, $\alpha_{\mathcal{M}} = \alpha_{\Theta} = 0$), discrimination-inference OED ($\alpha_{\mathcal{M}} = \alpha_{\Theta} = 1$, $\alpha_Z = 0$) and discrimination-goal-oriented OED ($\alpha_{\mathcal{M}} = \alpha_Z = 1$, $\alpha_{\Theta} = 0$).

**Hyperparameters.** The hyperparameters are listed in Table 3.14. The linear mapping in the output layer of the GMM net is the same as the uni-model example in Sec. 3.3.2.1.

Table 3.14: Hyperparameters of the multi-model source location finding problem.

| | vsOED-G-T | vsOED-G-I |
|---|---|---|
| #training iteration $n_{\text{update}}$ | 10001 | 10001 |
| #new episodes per iteration $n_{\text{episode}}$ | 1000 | 1000 |
| batch size $n_{\text{batch}}$ | 10000 | 10000 |
| model predictor initial lr | $10^{-3}$ | $10^{-3}$ |
| model predictor lr decay | 0.9999 | 0.9999 |
| #model predictor update per iteration | 5 | 5 |
| parameter predictor initial lr | $10^{-3}$ | $10^{-3}$ |
| parameter predictor lr decay | 0.9999 | 0.9999 |
| #param predictor update per iteration | 5 | 5 |
| $n_{\text{mixture}}$ | 8 | 8 |
| initial actor lr | $2 \times 10^{-4}$ | $10^{-3}$ |
| actor lr decay | 0.9999 | 0.9999 |
| initial critic lr | $10^{-3}$ | $10^{-3}$ |
| critic lr decay | 0.9999 | 0.9999 |
| max buffer size | $10^6$ | $10^6$ |
| discount factor $\gamma$ | 1 | 0.9 |
| initial design noise scale | 0.5 | 0.5 |
| design noise scale decay | 0.9999 | 0.9999 |
| target network lr | 0.1 | 0.1 |

**Training stability.** For the multi-model source location finding, the investigation of training stability is illustrated here only on the inference OED scenario. Figure 3.8 shows the training histories of the inference OED, where the solid line and the shaded region are the mean and standard error of 4 replicates with different random seeds. Similar to the uni-model case, the training of vsOED appears highly robust against randomization. Table 3.15 presents the PCE evaluation of optimal policies from 4 replicates of inference OED, optimized for horizon $N = 30$. Each element in the table represents the mean and standard error computed from 2000 samples. Table 3.16 further provides the mean and standard error aggregated from the means of these 4 replicates. These results further support that vsOED is robust under different random seeds.

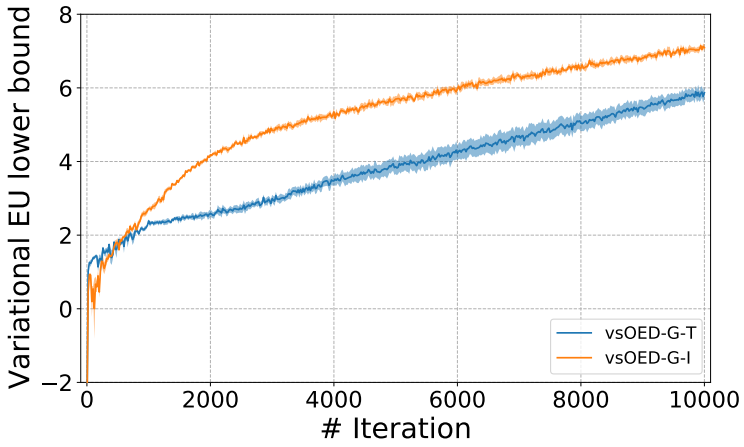



Figure 3.8: Training histories of PoI inference OED for the multi-model source location finding problem, optimized for horizon $N = 30$. The solid line and the shaded region are the mean and standard error of 4 replicates with different random seeds.

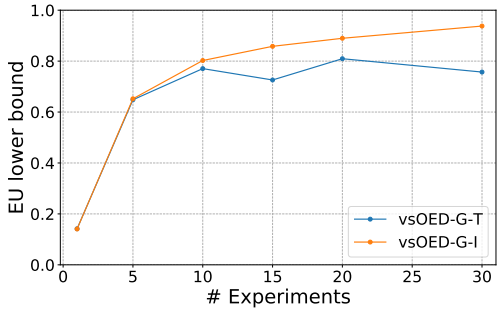
Table 3.15: PCE evaluation of optimal policies from 4 replicates of inference OED for the multi-model source location finding problem, optimized for horizon $N = 30$.

| | Run 1 | Run 2 | Run 3 | Run 4 |
|---|---|---|---|---|
| vsOED-G-T | 9.672 ± 0.055 | 9.263 ± 0.055 | 9.058 ± 0.057 | 8.833 ± 0.056 |
| vsOED-G-I | 10.567 ± 0.048 | 10.085 ± 0.049 | 10.464 ± 0.050 | 10.429 ± 0.048 |

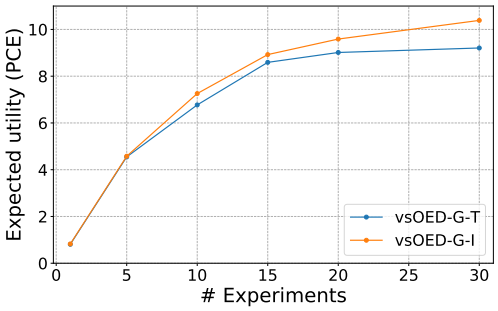
Table 3.16: Aggregated PCE evaluation results of optimal policies from 4 replicates of inference OED for the multi-model source location finding problem, optimized for horizon $N = 30$.

| | Mean | SE |
|---|---|---|
| vsOED-G-T | 9.206 | 0.154 |
| vsOED-G-I | 10.386 | 0.090 |

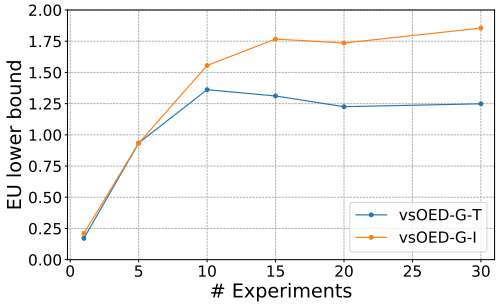
**Expected utilities.** Figure 3.9 plots the expected utilities of various OED scenarios, averaged over 2 replicates. The IIG formulation demonstrates greater stablility and higher performance, especially when the horizon $N$ is longer than 15.



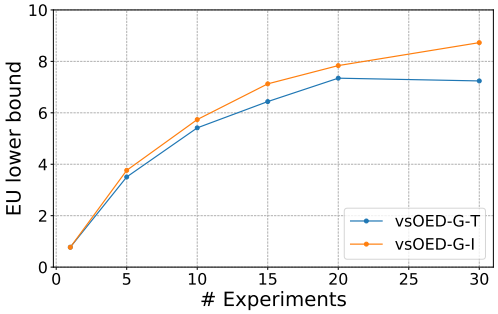

(a) Model discrimination OED



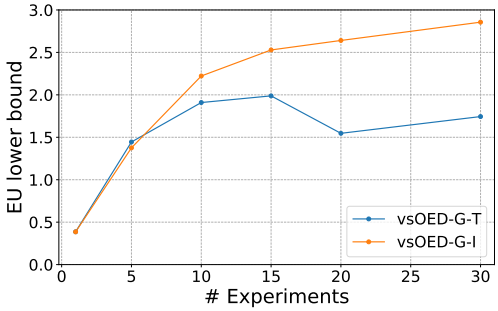

(b) Inference OED




(c) Goal-oriented OED




(d) Discrimination-inference OED




(e) Discrimination-goal-oriented OED

Figure 3.9: Expected utilities of various OED scenarios for the multi-model source location finding problem, averaged over 2 replicates. Variational lower bounds with $10^6$ samples are presented except for inference OED, where PCE with 2000 samples and $L = 10^6$ is used for evaluation.

**Policies.** Figure 3.10 plots example designs for various OED scenarios. The model discrimination OED tends to do more exploration, while the inference OED tends to exploit the knowledge about the source location. Similar to the uni-model example, the goal-oriented OED prefers to take vertical measurements. The discrimination-inference OED and the discrimination-goal-oriented

OED appear slightly more exploratory than the inference OED and the goal-oriented OED, respectively.

**Model discrimination.** Figure 3.11 illustrates the true model posteriors and the posteriors predicted by the model posterior predictor of the model discrimination OED, optimized for horizon $N = 30$. The policy learned by the model discrimination OED is effective in distinguishing between different models, and the predicted distributions from the model posterior predictor align well with the true posteriors.

To further illustrate that the model discrimination OED has found a good policy, we compare the EIG on model probability of various OED scenarios optimized for horizon $N = 30$ in Table 3.17. The EIG is calculated by PCE with 2000 samples and $L_m \approx \frac{10^6}{3}$ for $m = 1, 2, 3$. The model discrimination OED finds the optimal policy in terms of maximizing the EIG on model probability. The EIG on the model probability in the discrimination-inference OED and discrimination-goal-oriented OED are also higher than that of inference OED and goal-oriented OED, respectively.

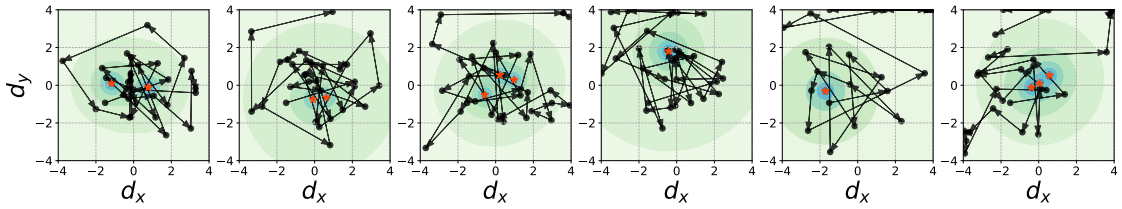
Table 3.17: EIG on model probability for various OED scenarios optimized for horizon $N = 30$.

| | Mean | SE |
|---|---|---|
| model discrimination OED | 1.020 | 0.003 |
| inference OED | 0.896 | 0.005 |
| goal-oriented OED | 0.815 | 0.005 |
| discrimination-inference OED | 0.950 | 0.005 |
| discrimination-goal-oriented OED | 0.967 | 0.004 |

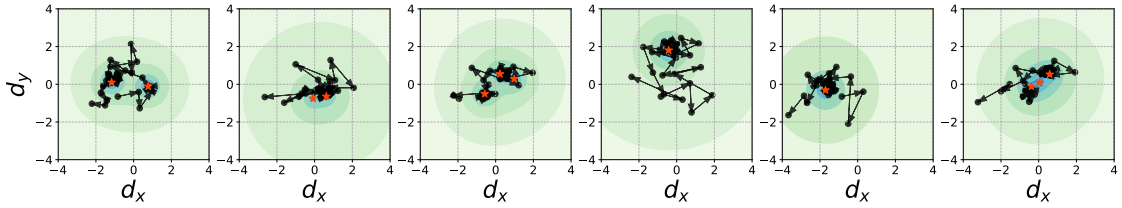
**PoI inference.** Table 3.18 lists the EIG on the PoI for various OED scenarios optimized for horizon $N = 30$. The EIG is calculated by PCE with 2000 samples and $L_m \approx \frac{10^6}{3}$ for $m = 1, 2, 3$. As expected, the inference OED finds the best policy in terms of maximizing the EIG on PoI inference.

Table 3.18: EIG on the PoI for various OED scenarios optimized for horizon $N = 30$.

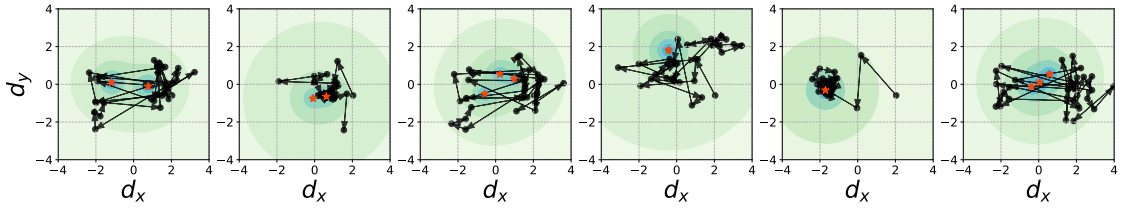
| | Mean | SE |
|---|---|---|
| model discrimination OED | 5.956 | 0.065 |
| inference OED | 10.567 | 0.048 |
| goal-oriented OED | 6.999 | 0.053 |
| discrimination-inference OED | 10.330 | 0.049 |
| discrimination-goal-oriented OED | 7.830 | 0.052 |



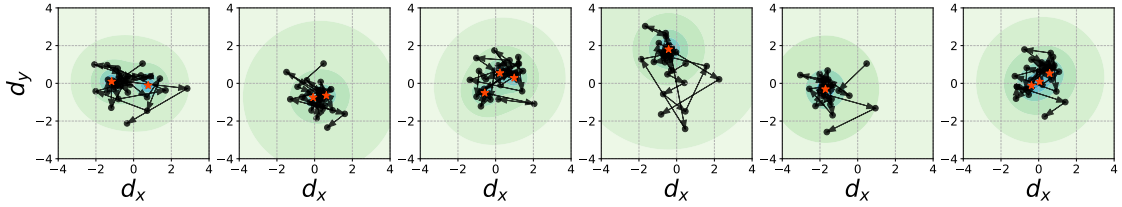

(a) Model discrimination OED



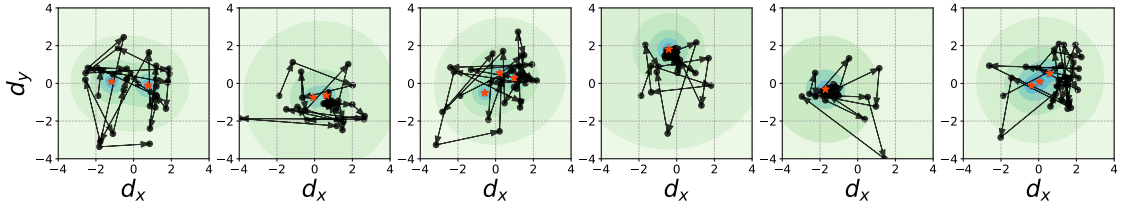

(b) Inference OED




(c) Goal-oriented OED




(d) Discrimination-inference OED




(e) Discrimination-goal-oriented OED

Figure 3.10: Example designs of various OED scenarios for the multi-model source location finding problem, optimized for horizon $N = 30$.

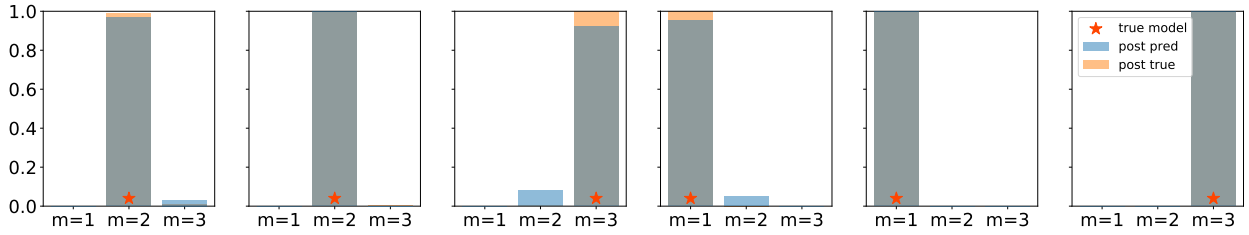



Figure 3.11: Example model posteriors from the model discrimination OED optimized for horizon $N = 30$.

### 3.3.3 Constant elasticity of substitution (CES)

This experiment involves a single model, and was previously studied in [56, 57, 14]. Constant elasticity of substitution (CES) falls under the domain of behavioral economics, where participants are presented with two baskets $x$ and $x'$ of goods and asked to assess the subjective difference in utility between the two baskets. The participants rate this difference on a sliding scale ranging from 0 to 1. The CES model [3] is then used to model the underlying utility function with latent PoIs $\theta = (\rho, \beta, \log u)$ with the following prior:

$$\begin{aligned} \rho &\sim Beta\,(1,1) \\ \beta &\sim Dirichlet\,([1,1,1]) \\ \log u &\sim \mathcal{N}\left(1, 3^2\right). \end{aligned}$$

It is worth noting that the degree of freedom of $\beta$ is 2 as the sum of $\beta_i$ for $i = 1, 2, 3$ is 1, therefore, only $\beta_1$ and $\beta_2$ are included in $\theta$. The design variable is $d = (x, x')$ where $x, x' \in [0, 100]^3$ represent the baskets of goods. The forward model is

$$\begin{aligned} U(x) &= \left(\sum_i x_i^\rho \beta_i\right)^{\frac{1}{\rho}} \\ \mu_\eta &= u \cdot (U(x) - U(x')) \\ \sigma_\eta &= \tau u \cdot (1 + \| x - x' \|) \\ \eta &\sim \mathcal{N}\left(\mu_\eta, \sigma_\eta^2\right) \\ y &= clip\,(sigmoid(\eta), \epsilon, 1-\epsilon)\,, \end{aligned} \tag{3.28}$$

where $\tau = 0.005$ and $\epsilon = 2^{-22}$.

**Hyperparameters.** The hyperparameters are listed in Table 3.19. We only use vsOED with TIG formulation since the horizon of this problem is at most 10. For the linear mapping in the

output layer of the GMM net, we transform the output of the GMM mean net of the PoI posterior predictor to a range of $[-1, 2]$ for $\rho$ and $\beta$, $[-17, 19]$ for $\log u$, and the output of the GMM standard deviation net of the PoI posterior predictor to a range of $[10^{-5}, 3]$ for all variables. The truncated normal distribution is used on $\rho$ and $\beta$ with support $[0, 1]$.

Table 3.19: Hyperparameters for the CES problem.

| | vsOED-G-T | vsOED-F-T |
|---|---|---|
| #training iteration $n_{\text{update}}$ | 10001 | 10001 |
| #new episodes per iteration $n_{\text{episode}}$ | 1000 | 1000 |
| batch size $n_{\text{batch}}$ | 10000 | 10000 |
| parameter predictor initial lr | $10^{-3}$ | $10^{-3}$ |
| parameter predictor lr decay | 0.9999 | 0.9999 |
| #param predictor update per iteration | 5 | 5 |
| $n_{\text{mixture}}$ | 8 | N/A |
| $n_{\text{trans}}$ | N/A | 4 |
| initial actor lr | $10^{-3}$ | $10^{-3}$ |
| actor lr decay | 0.9999 | 0.9999 |
| initial critic lr | $10^{-3}$ | $10^{-3}$ |
| critic lr decay | 0.9999 | 0.9999 |
| max buffer size | $10^6$ | $10^6$ |
| discount factor $\gamma$ | 1 | 1 |
| initial design noise scale | 5 | 5 |
| design noise scale decay | 0.9998 | 0.9998 |
| target network lr | 0.1 | 0.1 |

**Training stability.** Figure 3.12 shows the training histories of the CES problem, where the solid line and the shaded region are the mean and standard error of 4 replicates with different random seeds. The training process of the CES problem exhibits slightly more noise compared to the source location finding problem, but overall remains robust against randomization. Table 3.20 presents the PCE evaluation results of 4 replicates optimized for horizon $N = 10$. Each element in the table represents the mean and standard error computed from 2000 samples. Table 3.21 further provides the mean and standard error aggregated from the means of these 4 replicates.

Table 3.20: PCE evaluation of optimal policies from 4 replicates of PoI inference OED for the CES problem, optimized for horizon $N = 10$.

| | Run 1 | Run 2 | Run 3 | Run 4 |
|---|---|---|---|---|
| vsOED-G-T | 11.785 ± 0.068 | 12.340 ± 0.059 | 12.290 ± 0.058 | 11.125 ± 0.079 |
| vsOED-F-T | 8.401 ± 0.098 | 9.510 ± 0.086 | 8.908 ± 0.087 | 10.299 ± 0.081 |

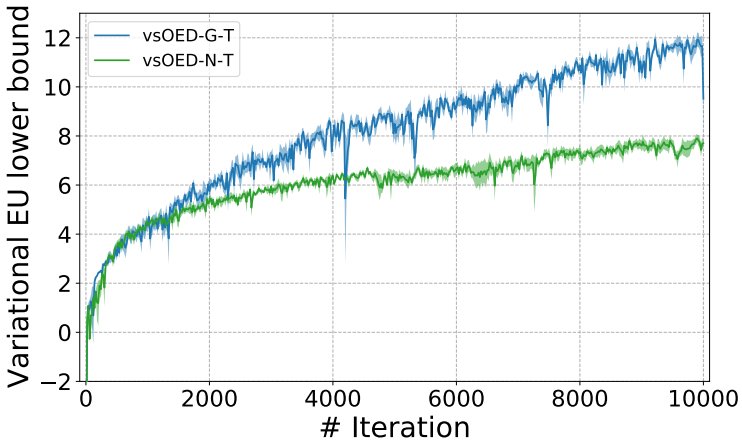



Figure 3.12: Training histories for the CES problem, optimized for horizon $N = 10$. The solid line and the shaded region are the mean and standard error of 4 replicates with different random seeds.

Table 3.21: Aggregated PCE evaluation of optimal policies from 4 replicates of PoI inference OED for the CES problem, optimized for horizon $N = 10$.

| | Mean | SE |
|---|---|---|
| vsOED-G-T | 11.885 | 0.245 |
| vsOED-F-T | 9.280 | 0.354 |

**Expected utilities and posteriors.** Figure 3.13a presents the expected cumulative utilities at various experiment stages, where all policies are optimized for a design horizon of $N = 10$ experiments and then evaluated on the various intermediate experiment stages. We restrict vsOED training to a total budget of 10 million episode samples, while fully training RL, DAD and iDAD using (RL 8 trillion episodes, DAD 100 billion episodes, and iDAD 200 million episodes). In Fig. 3.13b, we draw the expected utility plotted against the design horizon $N$. Each data point represents a new policy optimized for a specific design horizon. The computational budget of 10 million episodes is equally allocated to all methods for a fair comparison. Figure 3.13a demonstrates that the performance of vsOED with GMM and TIG is slightly inferior to the fully-trained RL but significantly better than fully-trained DAD and iDAD. The lower values for vsOED with GMM and TIG in the earlier stages indicate that the policy prioritizes long-term expected utility over short-term rewards. In Fig. 3.13b, it is evident that under a common budget, vsOED with GMM and TIG outperforms other baseline methods for all values of $N$. Furthermore, the shaded regions in Fig. 3.13b represent the robustness of vsOED-G-T and RL training against random seeds. On the other hand, vsOED-F-T and DAD exhibit higher noise.

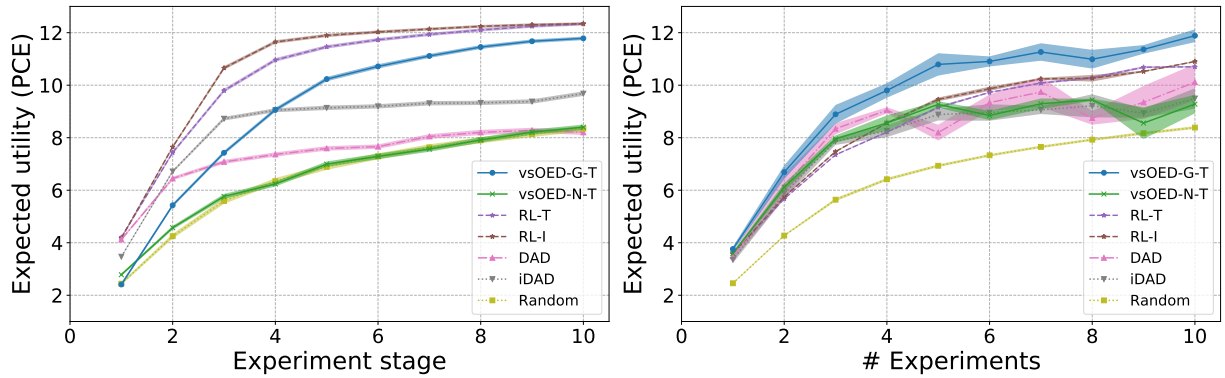
In this example, the performance of vsOED with NF is significantly worse compared to vsOED with GMM. This can be attributed to the limitation of NFs, as they are designed to handle random variables with infinite support, while in this case both $\rho$ and $\beta$ have finite support.

Figure 3.14 compares the posterior predicted by GMM and the true posterior. The GMM performs well in predicting $\log u$, but it tends to have wider posterior predictions for $\rho$ and $\beta$ compared to the true posterior. This is due to the nature of the CES problem, where many observations are clipped at the two ends as shown in Eqn. (3.28). Consequently, there are numerous observations with identical values, which makes it challenging for GMM to accurately learn the mapping from designs and observations to the posterior distribution. Nevertheless, despite the challenges in predicting the posterior accurately, vsOED with GMM is still able to find a good policy.

### 3.3.4 SIR model for disease spread

We demonstrate the ability of vsOED to handle implicit likelihood via the SIR example from [76]. This experiment only involves a single model.

SIR is a stochastic model [83, 37] describing the spread of infectious diseases in a population. Individuals in the population are divided into three categories: susceptible, infected, and recovered. Transitions among these categories are governed by the infection rate and recovery rate parameters. The SIR model problem aims to estimate the infection rate $\beta$ and the recovery rate $\rho$ by designing time points for measuring the number of infected individuals . Given a fixed population of size $N$,




(a) Expected cumulative utility versus experiment stage

(b) Expected utility versus design horizon

Figure 3.13: Expected utilities of various OED methods for th CES problem, all estimated using PCE with $L = 10^6$. (a) Mean and standard error (shaded) from 2000 evaluation episodes. (b) Mean and standard error (shaded) of 4 replicates with different random seeds, each replicate evaluated with 2000 episodes.

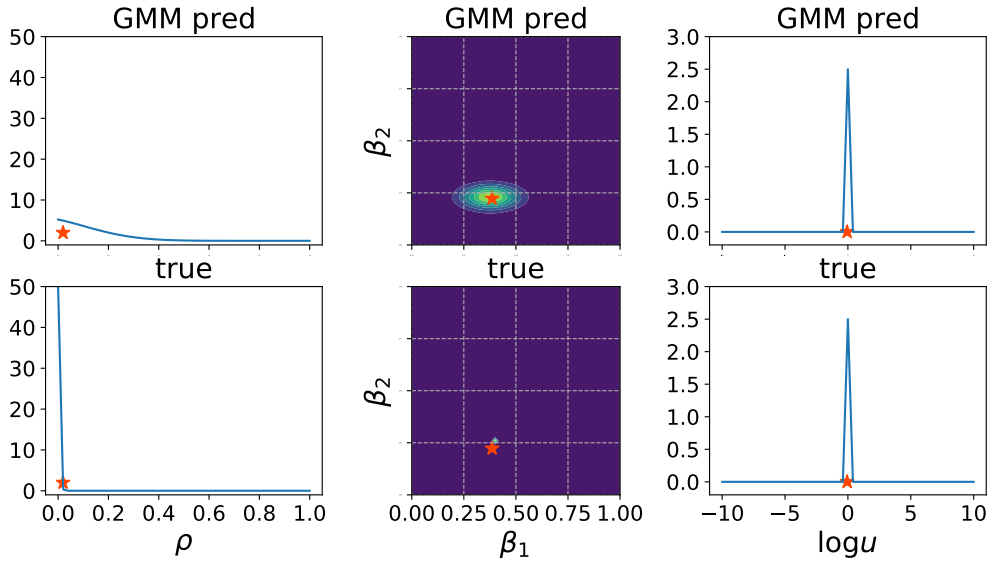



Figure 3.14: Examples of GMM posterior and true posterior for the CES problem at horizon $N = 10$. Red stars are the parameter values.

each individual starts from a susceptible state ($\tau$) ($\tau$ is time) to an infected state $I(\tau)$ with rate $\beta$, and then recovers back to the recovered state $R(\tau)$ with rate $\rho$.

The stochastic versions of SIR is usually defined by a continuous-time Markov chain (CTMC), which can be sampled via the Gillespie algorithm [2]. However, this generally yields discrete population states that have undefined gradients. We follow [76] to an alternative simulation algorithm that uses stochastic differential equations (SDEs), where yields continuous state populations and gradients can be approximated.

The population state vector is defined to be $\mathbf{X}(\tau) = (S(\tau), I(\tau))^T$, where $R(\tau)$ can be ignored as the population size is fixed. The system of Itô SDEs that defines the stochastic SIR model is given by:

$$d\mathbf{X}(\tau) = \mathbf{f}(\mathbf{X}(\tau))d\tau + \mathbf{G}(\mathbf{X}(\tau))d\mathbf{W}(\tau) \tag{3.29}$$

where $\mathbf{f}$ is the drift vector, $\mathbf{G}$ is the diffusion matrix, and $\mathbf{W}(\tau)$ is a vector of independent Wiener processes (also called Brownian motion). From [83], the drift vector and diffusion matrix are

$$\mathbf{f}(\mathbf{X}(\tau)) = \begin{pmatrix} -\beta\frac{S(\tau)I(\tau)}{N} \\ \beta\frac{S(\tau)I(\tau)}{N} - \rho I(\tau) \end{pmatrix} \text{ and } \mathbf{G}(\mathbf{X}(\tau)) = \begin{pmatrix} -\sqrt{\beta\frac{S(\tau)I(\tau)}{N}} & 0 \\ \sqrt{\beta\frac{S(\tau)I(\tau)}{N}} & -\sqrt{\rho I(\tau)} \end{pmatrix}. \tag{3.30}$$

Given Eqn. (3.29) and (3.30), we can simulate state populations $\mathbf{X}(\tau)$ by solving the SDE using finite-differencing methods, such as the Euler-Maruyama method. For a fair comparison, we follow [76] and just use the solutions of Eqn. (3.29) as data and do not consider an additional Poisson observational model that increases the noise in simulated data as suggested in [83].

The PoIs of this example are the logarithmic infection rate $\log\beta$ and the logarithmic recovery rate $\log\rho$ with the following prior:

$$\log\beta \sim \mathcal{N}\left(\log 0.5, 0.5^2\right)$$
$$\log\rho \sim \mathcal{N}\left(\log 0.1, 0.5^2\right).$$

The design variable $d$ is the time $\tau \in [0, 100]$ for taking measurements, where the observable is the number of infected people $I(\tau)$. $I(\tau)$ can be obtained by solving Eqn. (3.29). Solving the underlying SDE is expensive, we thus limit the computational budget to 1 million forward model simulations for both vsOED and iDAD. To accelerate the training process, we pre-generate and store 1 million simulations, and access the stored simulations during the training. A new set of $3 \times 10^5$ simulations are used as evaluation data. We emphasize that the likelihood of stochastic SIR model is implicit. This is because we can only sample from the likelihood, but evaluating the

likelihood PDF directly is not possible due to the stochastic nature of the process.

**Hyperparameters.** The hyperparameters are listed in Table 3.22. We only use vsOED with TIG formulation since the horizon of this problem is at most 10. For the linear mapping in the output layer of the GMM net, we transform the output of the GMM mean net of the PoI posterior predictor to a range of $[-6, 4]$, and the output of the GMM standard deviation net of the PoI posterior predictor to a range of $[10^{-5}, 0.5]$ for all variables. The truncated normal distribution is not used in this case.

Table 3.22: Hyperparameters for the SIR problem.

| | vsOED-G-T | vsOED-F-T |
|---|---|---|
| #training iteration $n_{\text{update}}$ | 10001 | 10001 |
| #new episodes per iteration $n_{\text{episode}}$ | 1000 | 1000 |
| batch size $n_{\text{batch}}$ | 10000 | 10000 |
| parameter predictor initial lr | $5 \times 10^{-4}$ | $10^{-3}$ |
| parameter predictor lr decay | 0.9999 | 0.9999 |
| #param predictor update per iteration | 5 | 5 |
| $n_{\text{mixture}}$ | 8 | N/A |
| $n_{\text{trans}}$ | N/A | 4 |
| initial actor lr | $5 \times 10^{-4}$ | $5 \times 10^{-4}$ |
| actor lr decay | 0.9999 | 0.9999 |
| initial critic lr | $10^{-3}$ | $10^{-3}$ |
| critic lr decay | 0.9999 | 0.9999 |
| max buffer size | $10^6$ | $10^6$ |
| discount factor $\gamma$ | 1 | 1 |
| initial design noise scale | 5 | 5 |
| design noise scale decay | 0.9999 | 0.9999 |
| target network lr | 0.1 | 0.1 |

**Training stability.** Figure 3.15 presents the training histories for the SIR problem, where the solid line and the shaded region are the mean and standard error of 4 replicates with different random seeds. Overall, the training of vsOED is highly stable, with consistent performance across different random seeds, except for a dip in the training of vsOED-G-T. Table 3.23 presents the lower bound evaluation of optimal policies from 4 replicates optimized for horizon $N = 10$. Each element in the table represents the mean and standard error computed from $3 \times 10^5$ samples. Table 3.24 further provides the mean and standard error aggregated from the means of these 4 replicates, supporting the robustness of vsOED in the SIR problem.

**Policies.** Fig. 3.16 shows the trajectories and designs of 3 realizaitons of $(\beta, \rho)$ with different ratios $R = \beta/\rho$. Smaller $R$ corresponds to a more spreading design, which aligns with the results in [76].

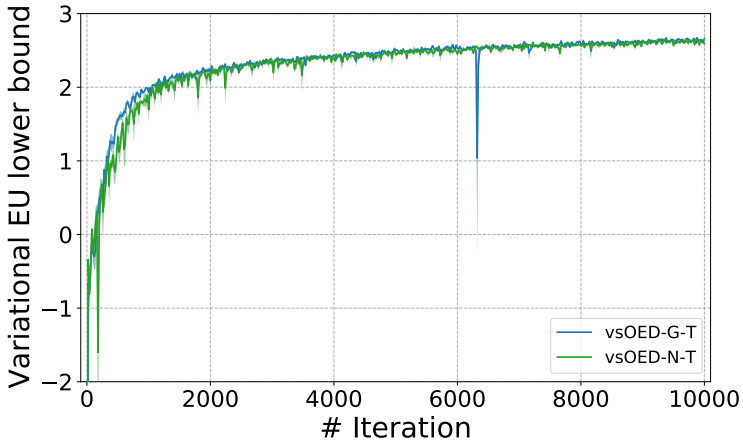


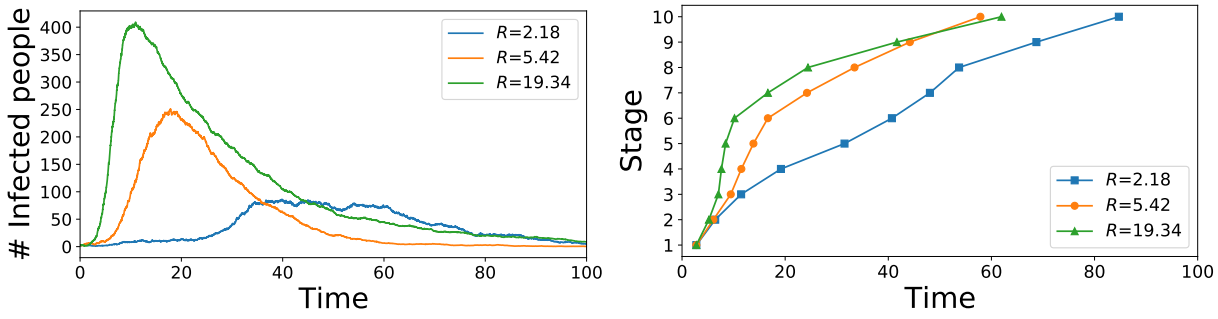

Figure 3.15: Training histories for the SIR problem, optimized for horizon $N = 10$. The solid line and the shaded region are the mean and standard error of 4 replicates with different random seeds.

Table 3.23: Variational expected utility lower bounds of optimal policies from 4 replicates for the SIR problem, optimized for horizon $N = 10$.

| | Run 1 | Run 2 | Run 3 | Run 4 |
|---|---|---|---|---|
| vsOED-G-T | $4.091 \pm 0.002$ | $4.093 \pm 0.002$ | $4.090 \pm 0.001$ | $4.092 \pm 0.001$ |
| vsOED-F-T | $4.097 \pm 0.002$ | $4.100 \pm 0.002$ | $4.091 \pm 0.002$ | $4.106 \pm 0.002$ |

Table 3.24: Aggregated variational expected utility lower bounds of optimal policies from 4 replicates for the SIR problem, optimized for horizon $N = 10$.

| | Mean | SE |
|---|---|---|
| vsOED-G-T | 4.092 | 0.001 |
| vsOED-F-T | 4.099 | 0.003 |

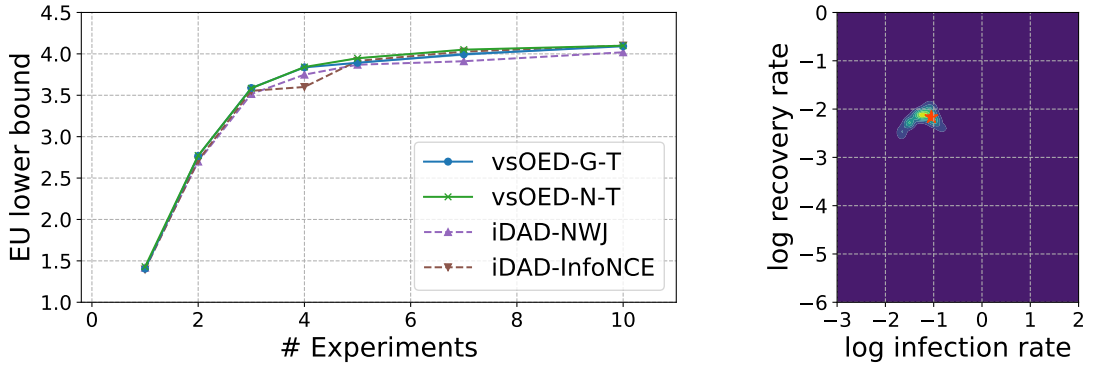



(a) SIR trajectories

(b) SIR policies

Figure 3.16: (a) SIR trajectories for 3 realizations of $(\beta, \rho)$ with different ratios $R = \beta/\rho$. (b) Corresponding designs.

**Expected utilities and the posterior.** Since the likelihood is implicit, PCE cannot be used to evaluate the expected utility. As a result, we directly present the variational lower bound as an alternative metric for measuring the performance of trained policies. Figure 3.17a illustrates the variational lower bound of the expected utility versus design horizon. vsOED and iDAD perform similarly, with vsOED slightly better in some cases. However, we note that the comparison is not entirely commensurable since the two methods use different variational lower bounds, and that iDAD additionally uses the forward model derivative. A potential benefit for vsOED is that it does not require forward model derivative, which can be valuable where model derivative is inaccessible. Figure 3.17b shows an example posterior generated from the GMM, which we see is consistent with the true data-generating values. Section 3.3.4 contains additional details on the SIR example.




(a) Expected utility versus design horizon

(b) GMM posterior at $N = 2$

Figure 3.17: (a) vsOED plot is the mean and standard error (shaded) from 4 replicates with different random seeds, each replicate evaluated with $3 \times 10^5$ episodes. Shaded regions are practically invisible, suggesting robustness. (b) An example posterior generated from the GMM.

### 3.3.5 Convection-diffusion

The last example entails finding the optimal sensor movement locations within a chemical contaminant plume governed by the 2D convection-diffusion PDE. This experiment is similar to the source location finding case in Sec. 3.3.2, and Sec. 2.3.2 for the uni-model example.

Here we consider $\mathcal{M} = 3$ candidate models with uniform prior model probability (i.e. $P(m) = 1/3$). For the $m$th model ($m \in \{1, 2, 3\}$), there are $m$ chemical contaminant sources randomly located in a 2D domain. Instead of an analytical function to describe the source signal, here the contaminant concentration at a given time $t$ and a location $\xi = [\xi_x, \xi_y] \in \mathbb{R}^2$ is determined by a

convection-diffusion partial differential equation (PDE). Specifically, for the $m$th model:

$$\frac{\partial G(\xi, t; m, \theta_m, \eta_m)}{\partial t} = \nabla^2 G - u(\eta_m) \cdot \nabla G + S(\xi, t; m, \theta_m), \qquad \xi \in [-1, 2]^2, \quad 0 \le t \le 0.2,$$

where $u = [v \cos\beta, v \sin\beta] \in \mathbb{R}^2$ is the convection velocity that is described by nuisance parameters $\eta_m$, which encompasses the convection speed (magnitude) $v \sim \mathcal{U}[0, 20]$ and the convection angle $\beta \sim \mathcal{U}[0, 2\pi]$. The source function is

$$S(\xi, t; m, \theta_m) = \sum_{i=1}^{m} \frac{s}{2\pi h^2} \exp\left(-\frac{\left\| \theta_{m,i} - \xi \right\|^2}{2h^2}\right),$$

where the PoIs $\theta_m = \{\theta_{m,1}, \ldots, \theta_{m,m}\}$ entails the source locations for $m$th model, $s = 2$ is the known source strength, and $h = 0.05$ is the known source width. The initial condition of the PDE is $G(\xi, 0; m, \theta_m, \eta_m) = 0$ and homogeneous Neumann boundary conditions are applied to all sides of the domain. The PoI prior is

$$\theta_{m,i,x}, \theta_{m,i,y} \sim \mathcal{U}[0, 1],$$

and the observation model is

$$y = G(d, t; m, \theta_m, \eta_m) + \epsilon,$$

with $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and $\sigma = 0.05$ being the observation noise standard deviation, and $d = [d_x, d_y] \in [0, 1]^2$ the design variable is the sensor locations for taking measurements.

Similar to Sec. 3.3.2, we are interested in the integrated flux at the right boundary and at a future time $t = 0.2$, with the formula

$$J(m, \theta_m, \eta_m) = \int_{\xi_y=0}^{1} -\frac{\partial G((\xi_x = 1, \xi_y), t = 0.2; m, \theta_m, \eta_m)}{\partial \xi_x} \, d\xi_y,$$

and the overall goal-oriented QoI is $z_m = \log\left(|J(m, \theta_m, \eta_m)| + 10^{-27}\right)$.

The OED problem involves designing a sequence of $N$ sensor locations over time (i.e. relocation movements), where the $k$th experiment is performed at time $t_k = 0.01(k+1)$. Moreover, we assume that the initial sensor location is at $\xi_0 = [0.5, 0.5]$, and we incorporate a sensor movement penalty to the immediate rewards to reflect the cost of moving. For stage $k = 0$, the penalty is $-0.1 \parallel d_0 - \xi_0 \parallel$, and for stage $k = 1, \ldots, N-1$, the penalty is $-0.1 \parallel d_k - d_{k-1} \parallel$; hence, a further movement would incur a higher cost.

**Surrogate model.** To solve the convection-diffusion PDE, we employ the second-order finite

volume method on a uniform grid. The grid has a size of $\Delta\xi_x = \Delta\xi_y = 0.01$, ensuring a consistent spatial discretization. For time integration, we utilize the second-order fractional step method with a time step of $\Delta t = 5.0 \times 10^{-4}$. The flux can be easily computed by applying the finite difference to the grids near the right boundary.

While using the numerical PDE solver as the forward model is possible, it can be computationally expensive and wasteful, since the numerical PDE solver solves for the concentration over the entire domain, while only a small subset of these values are used in the forward model. Therefore, to accelerate the computation, we pre-build NN surrogate forward models of $G(\xi, t_k; m, \theta_m, \eta_m)$ for each $m$th model and at each $t_k$ for $k = 0, \ldots, N-1$. We also build NN surrogate goal-oriented prediction models of $J(m, \theta_m, \eta_m)$ for each $m$th model. The architecture of the surrogate forward and prediction models are summarized in Table 3.25 and 3.26. 20,000 simulations are generated with random $\theta_m$ and $\eta_m$ for each $m$th model. From these, 18,000 are used for training, and 2000 for testing. The testing mean squared errors (MSE) are summarized in Table 3.27, showing good surrogate accuracy. Figure 3.18 presents comparisons of true and surrogate model predicted concentration fields, also indicating good agreement.

Table 3.25: Architecture of the surrogate forward model.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | $[\theta_m, \eta_m, \xi]$ | $2m + 4$ | - |
| H1 | Dense | 256 | ReLU |
| H2 | Dense | 256 | ReLU |
| H3 | Dense | 256 | ReLU |
| H4 | Dense | 256 | ReLU |
| Output | Dense | 1 | - |

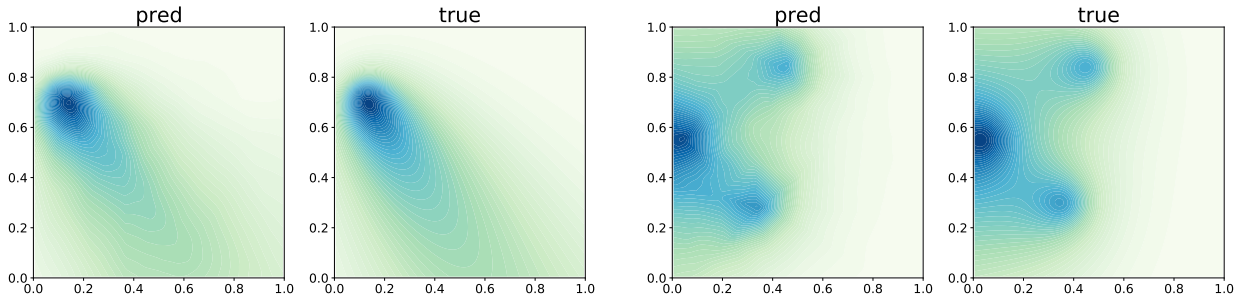
Table 3.26: Architecture of the surrogate prediction model.

| Layer | Description | Dimension | Activation |
|---|---|---|---|
| Input | $[\theta_m, \eta_m]$ | $2m + 2$ | - |
| H1 | Dense | 256 | ReLU |
| H2 | Dense | 256 | ReLU |
| H3 | Dense | 256 | ReLU |
| Output | Dense | 1 | - |

**Hyperparameters.** The hyperparameters are listed in Table 3.28. We only use vsOED with the TIG formulation. For the linear mapping in the output layer of the GMM net, we transform the output of the GMM mean net of the PoI posterior predictor to a range of $[-1, 2]$, and the output of

Table 3.27: Testing MSE of surrogate models.

| Model | surrogate forward model | surrogate prediction model |
|---|---|---|
| $m = 1$ | $3.094 \times 10^{-5}$ | $4.141 \times 10^{-5}$ |
| $m = 2$ | $3.284 \times 10^{-4}$ | $4.986 \times 10^{-4}$ |
| $m = 3$ | $1.650 \times 10^{-3}$ | $2.080 \times 10^{-3}$ |




(a) $m = 1$ (b) $m = 3$

Figure 3.18: Example comparisons between true and surrogate model predicted concentration fields.

the GMM standard deviation net of the PoI posterior predictor to a range of $[10^{-5}, 1]$ for all PoIs. For the goal-oriented QoI posterior predictor, we transform the output of the GMM mean net to a range of $[-15, 3]$, and the output of the GMM standard deviation net to a range of $[10^{-5}, 4]$. The truncated normal distribution is used on all PoIs with support $[0, 1]$.

Similar to the multi-model source location finding problem, 5 scenarios will be considered: model discrimination OED ($\alpha_{\mathcal{M}} = 1$, $\alpha_{\Theta} = \alpha_Z = 0$), inference OED ($\alpha_{\Theta} = 1$, $\alpha_{\mathcal{M}} = \alpha_Z = 0$), goal-oriented OED ($\alpha_Z = 1$, $\alpha_{\mathcal{M}} = \alpha_{\Theta} = 0$), discrimination-inference OED ($\alpha_{\mathcal{M}} = \alpha_{\Theta} = 1$, $\alpha_Z = 0$) and discrimination-goal-oriented OED ($\alpha_{\mathcal{M}} = \alpha_Z = 1$, $\alpha_{\Theta} = 0$).

**Training stability.** For brevity, the investigation of training stability is illustrated here only on the inference OED scenario, with similar observations found in other scenarios. Figure 3.8 shows the training histories of the inference OED, where the solid line and the shaded region are the mean and standard error of 4 replicates with different random seeds. Note that the prior term is omitted due to the presence of nuisance parameters. Table 3.29 presents the variational lower bounds evaluated for optimal policies from 4 replicates of inference OED, optimized for horizon $N = 10$. Each element in the table represents the mean and standard error computed from $10^6$ samples. Table 3.30 further provides the mean and standard error aggregated from the means of these 4 replicates. From these results, vsOED demonstrates excellent robustness under different random seeds.

Table 3.28: Hyperparameters for the convection-diffusion problem.

| | vsOED-G-T |
|---|---|
| #training iteration $n_{\text{update}}$ | 10001 |
| #new episodes per iteration $n_{\text{episode}}$ | 1000 |
| batch size $n_{\text{batch}}$ | 10000 |
| model predictor initial lr | $10^{-3}$ |
| model predictor lr decay | 0.9999 |
| #model predictor update per iteration | 5 |
| parameter predictor initial lr | $10^{-3}$ |
| parameter predictor lr decay | 0.9999 |
| #param predictor update per iteration | 5 |
| $n_{\text{mixture}}$ | 8 |
| initial actor lr | $5 \times 10^{-4}$ |
| actor lr decay | 0.9999 |
| initial critic lr | $10^{-3}$ |
| critic lr decay | 0.9999 |
| max buffer size | $10^6$ |
| discount factor $\gamma$ | 1 |
| initial design noise scale | 0.05 |
| design noise scale decay | 0.9999 |
| target network lr | 0.1 |

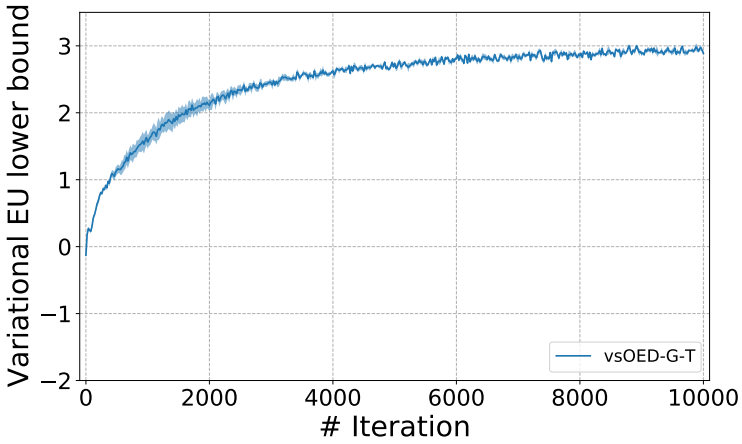



Figure 3.19: Training histories of inference OED for the convection-diffusion problem, optimized for horizon $N = 10$. The solid line and the shaded region are the mean and standard error of 4 replicates with different random seeds.

Table 3.29: Variational lower bounds evaluated for optimal policies from 4 replicates of inference OED for the convection-diffusion problem, optimized for horizon $N = 10$.

| | Run 1 | Run 2 | Run 3 | Run 4 |
|---|---|---|---|---|
| vsOED-G-T | 2.998 ± 0.002 | 3.057 ± 0.002 | 2.857 ± 0.002 | 3.039 ± 0.002 |

Table 3.30: Aggregated variational lower bounds evaluated for optimal policies from 4 replicates of inference OED for the convection-diffusion problem, optimized for horizon $N = 10$.

| | Mean | SE |
|---|---|---|
| vsOED-G-T | 2.988 | 0.039 |

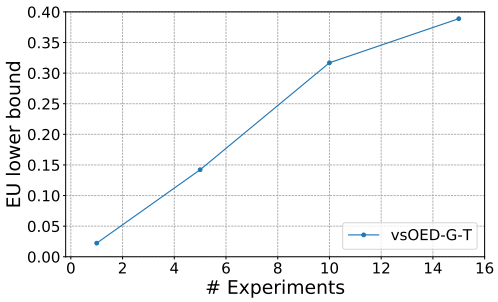
**Expected utilities.** Figure 3.20 plots the expected utilities of various OED scenarios, averaged over 2 replicates.

**Policies.** Figure 3.21 plots example designs of various OED scenarios optimized for horizon $N = 10$. The overall behavior is similar to the multi-model source location finding problem . The policy behavior of model discrimination OED is more exploratory often extending to the boundaries of the domain, leading to a high IG (narrow posterior) on model probability as shown in Fig. 3.22a. The inference OED policy appears to explore closer around the estimated sources while leveraging the background convection, and the goal-oriented policy exhibits a similar vertical design tendency as explained in the Sec. 3.3.2 example.

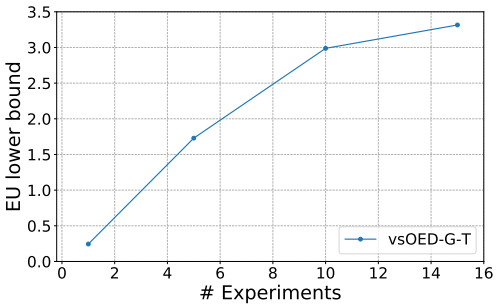
**Posteriors.** Figure 3.22 illustrates the model and PoI posteriors of the model discrimination OED and inference OED optimized for horizon $N = 10$. The model posterior predictor and GMM both effectively approximate their true posteriors.

## 3.4 Summary

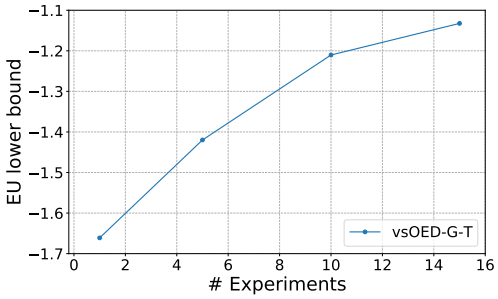
In this chapter, we propose a novel variational sequential optimal experimental design (**vsOED**) method to alleviate the expensive computational requirements for solving PG-sOED. The key idea of vsOED is to estimate and maximize an $O(n)$ lower bound formed through variational approximation of the Bayesian posteriors without needing explicit likelihood and prior. Notably, in contrast to existing sequential design algorithms that primarily focus on the EIG of model parameters within a single model, vsOED offers a *unified framework* that accommodates multi-model scenarios with diverse design objectives, including EIG for model probability, parameters of interest, and predictive quantities of interest, even in the presence of nuisance parameters. Furthermore, RL techniques are utilized to enhance the performance and efficiency of vsOED. We implement vsOED on benchmark



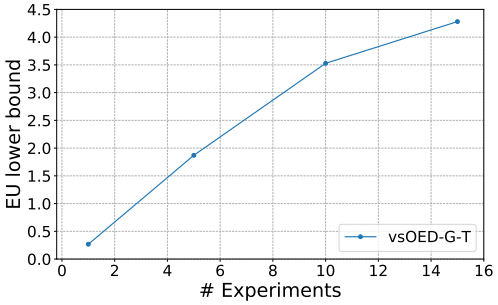

(a) Model discrimination OED



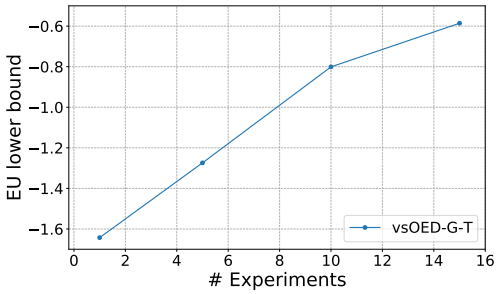

(b) Inference OED




(c) Goal-oriented OED

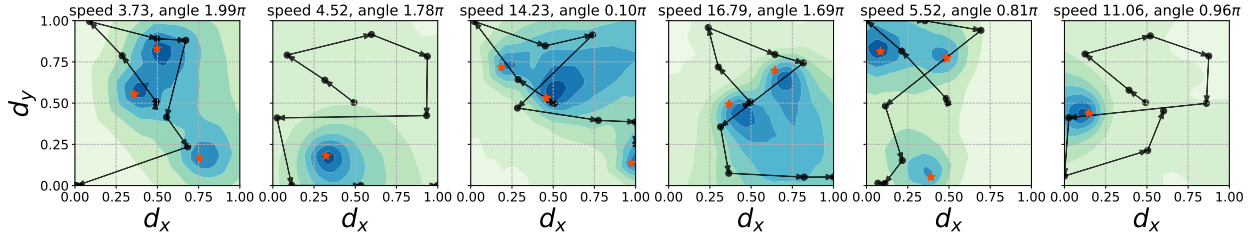



(d) Discrimination-inference OED

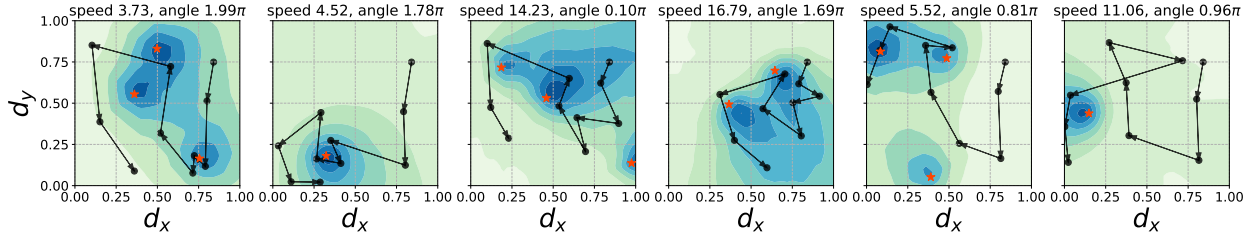



(e) Discrimination-goal-oriented OED

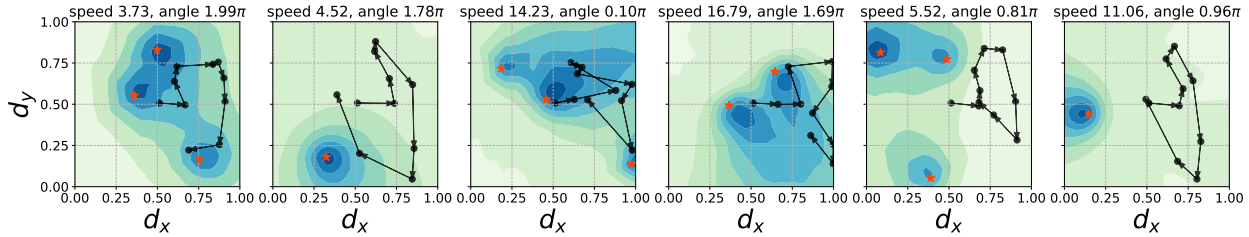
Figure 3.20: Expected utilities of various OED scenarios, averaged over 2 replicates. Variational lower bounds are evaluated using $10^6$ samples.



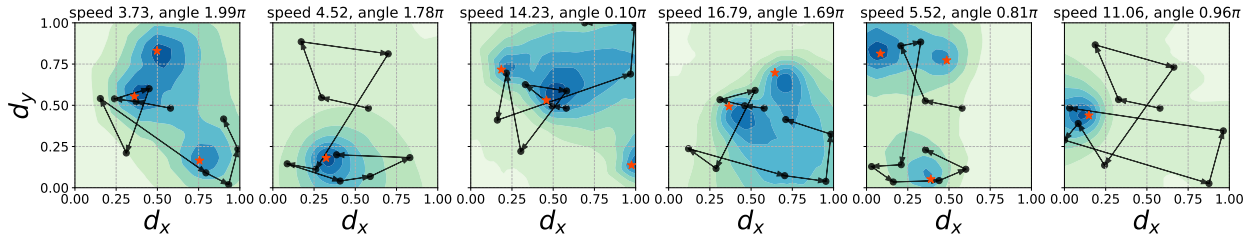

(a) Model discrimination OED



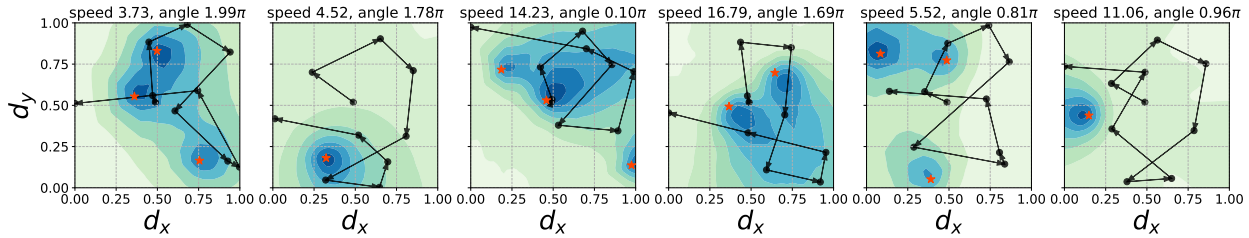

(b) Inference OED




(c) Goal-oriented OED

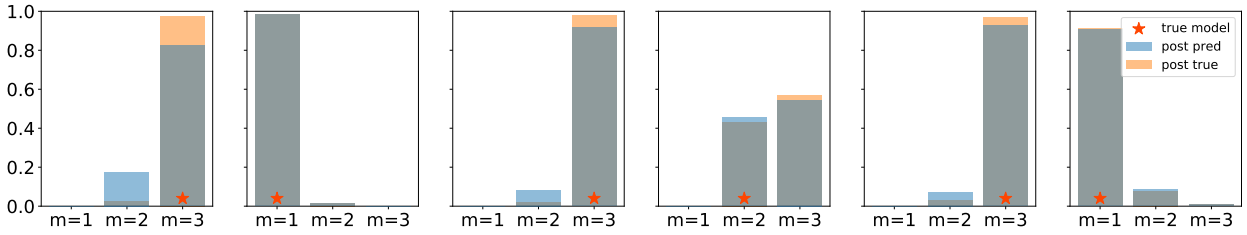



(d) Discrimination-inference OED

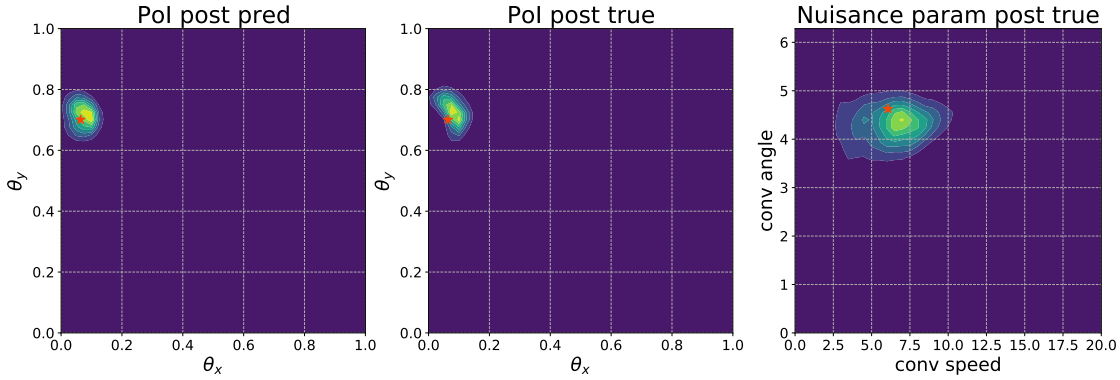



(e) Discrimination-goal-oriented OED

Figure 3.21: Example designs of various OED scenarios of the convection-diffusion problem, optimized for horizon $N = 10$.




(a) Model posteriors from the model discrimination OED scenario




(b) Parameter posteriors at $m = 1$ from the inference OED scenario

Figure 3.22: Example model and parameter posteriors from the model discrimination OED and inference OED for the convection-diffusion problem optimized for horizon $N = 10$.

problems and illustrate significantly improved sampling efficiency under fixed budgets of forward model runs, and also demonstrate with a physics-based model with PDE-governed dynamics.

The key contributions and novelty of our vsOED method are summarized as follows.

- We formulate the vsOED framework and generalize its usage to a wide range of OED scenarios.
- We provide a proof demonstrating the equivalence of the objective function when utilizing the full information gain and its one-point estiamte as the reward.
- We present the numerical techniques for solving vsOED problems, specifically the Monte Carlo estimator of policy gradient and variational gradient, and the DNN architectures for posterior approximation, policy and value functions.
- We validate vsOED on a number of cases, demonstrating its efficiency over other baseline methods, and its versatility in addressing diverse OED scenarios.
- We make available our vsoed code at https://github.com/wgshen/vsOED.

# CHAPTER 4

# Robust Optimal Experimental Design

The formulations in Chapter 2 and 3 focus on maximizing the *expected* (i.e., average) utility, they do not account for the risk of obtaining very low (or high) utility values. While one may quantify risk in different ways (e.g., mean-plus-variance, mean-plus-deviation/semi-deviation, conditional value-at-risk, entropic risk), we adopt the simple mean-plus-variance [100] approach and employ the *variance* to capture the dispersion of utility realizations. By incorporating both the expectation and variance of utility into a single optimization problem, one may find new designs that, for example, trades off some average utility in order to achieve a much lower risk.

In this chapter, we introduce the variance-penalized design criterion for achieving **robust optimal experimental design (rOED)** for batch (non-sequential) designs (robust sequential cases will be presented in the next chapter). The chapter begins with the formulation of variance-penalized rOED. We then propose a double-nested Monte Carlo (MC) estimator for the variance-penalized criterion and also derive its convergence rate. Numerical examples are presented, including a linear-Gaussian benchmark to validate the convergence of the proposed estimator, a synthetic nonlinear case to show the benefits of rOED, and a contaminant source inversion case with and without building obstacles to demonstrate its usage in more realistic physical problems.

The code for this chapter is available at: https://github.com/wgshen/rOED.

## 4.1 Problem formulation

### 4.1.1 Background

We begin by reviewing the notation and formulation for batch OED. Adopting the same notation as previous chapters, we let $d \in \mathbb{R}^{N_d}$ denote the controllable design variables of an experiment, $\theta \in \mathbb{R}^{N_\theta}$ the unknown model parameters, and $y \in \mathbb{R}^{N_y}$ the observations obtained from the experiments; $N_d$, $N_\theta$, and $N_y$ are the dimensions of design, parameter, and observation spaces, respectively. When an experiment is carried out under design $d$ and observation $y$ is obtained, the probability

density function (PDF) of the unknown parameters can be updated according to Bayes' rule:

$$p(\theta|y,d) = \frac{p(y|\theta,d)\,p(\theta|d)}{p(y|d)}, \tag{4.1}$$

where $p(\theta|d)$ is the prior PDF, $p(y|\theta,d)$ is the likelihood, $p(y|d)$ is the model evidence or marginal likelihood, and $p(\theta|y,d)$ is the posterior. The prior belief of $\theta$ should not be affected by the selected experiment design, thus $p(\theta|d)$ can be simplified to $p(\theta)$. The likelihood often results from an observation model such as

$$y = G(\theta,d) + \epsilon, \tag{4.2}$$

where $G$ is a forward model that governs the underlying experimental process (e.g., a system of partial differential equations (PDEs)), and $\epsilon \in \mathbb{R}^{N_y}$ represents the measurement noise. Often measurement noise consists of the superposition of a large number of small zero-mean random perturbations, and so by the Central Limit Theorem $\epsilon \sim \mathcal{N}(0,\Sigma_\epsilon)$ can be a reasonable representation. Thus, each likelihood evaluation $p(y|\theta,d) = p_\epsilon(y - G(\theta,d))$ involves a forward model solve, which is often the most computationally expensive part.

From a decision-theoretic view, the objective of OED can be stated as an *expected utility*:

$$U(d) = \int_{\mathcal{Y}} \int_{\Theta} p(\theta,y|d)u(d,y,\theta)\,d\theta\,dy, \tag{4.3}$$

where $u(d,y,\theta)$ is the *utility function* for a particular realization of $d$, $y$, and $\theta$, and the expectation is taken over $p(\theta,y|d)$ since $\theta$ and $y$ are random.

An information-theoretic utility can be adopted based on the Bayesian formulation [92], we choose the utility function to be the Kullback-Leibler (KL) divergence from the prior to the posterior:

$$u(d,y,\theta) = D_{\mathrm{KL}}(p_\theta(\cdot|y,d)\,||\,p_\theta(\cdot)) = \int_\Theta p(\tilde{\theta}|y,d)\ln\frac{p(\tilde{\theta}|y,d)}{p(\tilde{\theta})}\,d\tilde{\theta} = u(d,y), \tag{4.4}$$

where the KL divergence has a non-negative value which quantifies the difference between two distributions. Note that the last equality results from the utility itself involves taking an expectation over the parameter space, and hence not a function of $\theta$. We will then use this $u(d,y)$ as the utility function in the rest of this chapter. By substituting the utility function Eqn. (4.4) into Eqn. (4.3),

we obtain the expected utility:

$$\begin{aligned} U(d) &= \int_{\mathcal{Y}} \int_{\Theta} p(\theta, y|d) u(d, y)\, d\theta\, dy \\ &= \int_{\mathcal{Y}} p(y|d) u(d, y)\, dy \\ &= \int_{\mathcal{Y}} p(y|d) \int_{\Theta} p(\theta|y, d) \ln \frac{p(\theta|y, d)}{p(\theta)}\, d\theta\, dy, \end{aligned} \tag{4.5}$$

where the second equality is due to the marginalization of the outer integral of $\theta$. The expected utility is therefore the *expected information gain* (EIG) on parameters $\theta$ over observations $y$ given design variables $d$, and is also equivalent to the *mutual information* between $\theta$ and $y$ given $d$. We emphasize that the inner integral of Eqn. (4.5) reflects the update of knowledge on $\theta$ by observing $y$ (i.e., information gain), and the outer integral is considering all possible experimental outcomes because $y$ is not known when designing the experiment.

The OED problem then involves solving the following optimization problem:

$$d_U^* = \arg\max_{d \in \mathcal{D}} U(d), \tag{4.6}$$

where $d_U^*$ is the optimal design that maximizes the expected utility within the design space $\mathcal{D}$.

### 4.1.2 Utility variance

The variance of utility $u(d, y)$ is (recall $\theta$ is dropped per Eqn. (4.4)):

$$\begin{aligned} \tilde{U}(d) &= \mathbb{V}_{y|d} \left[ u(d, y) \right] \\ &= \mathbb{E}_{y|d} \left\{ \left[ u(d, y) - U(d) \right]^2 \right\} \\ &= \mathbb{E}_{y|d} \left[ u(d, y)^2 \right] - U(d)^2, \end{aligned} \tag{4.7}$$

where the second term in Eqn. (4.7) is the square of the expected utility. We introduce short-hand notation $\tilde{U}_{\mu_2}(d)$ to denote the first term, with the subscript $\mu_2$ indicating second moment. This

term can be further expanded as

$$
\begin{aligned}
\tilde{U}_{\mu_2}(d) &= \int_{\mathcal{Y}} p(y|d) \left[ \int_{\Theta} p(\theta|y,d) \ln \frac{p(\theta|y,d)}{p(\theta)} \, d\theta \right]^2 dy \\
&= \int_{\mathcal{Y}} p(y|d) \left[ \int_{\Theta} p(\theta|y,d) \ln \frac{p(y|\theta,d)}{p(y|d)} \, d\theta \right]^2 dy \qquad (4.8) \\
&= \int_{\mathcal{Y}} p(y|d) \left[ \ln p(y|d) \right]^2 dy \\
&\quad - 2 \int_{\mathcal{Y}} p(y|d) \ln p(y|d) \int_{\Theta} p(\theta|y,d) \ln p(y|\theta,d) \, d\theta \, dy \\
&\quad + \int_{\mathcal{Y}} p(y|d) \left[ \int_{\Theta} p(\theta|y,d) \ln p(y|\theta,d) \, d\theta \right]^2 dy \\
&= \int_{\mathcal{Y}} p(y|d) \left[ \ln p(y|d) \right]^2 dy \qquad (\tilde{U}_{\mu_2,1}(d)) \qquad (4.9) \\
&\quad - 2 \int_{\Theta} p(\theta) \int_{\mathcal{Y}} p(y|\theta,d) \ln p(y|d) \ln p(y|\theta,d) \, dy \, d\theta \qquad (\tilde{U}_{\mu_2,2}(d)) \\
&\quad + \int_{\mathcal{Y}} p(y|d) \left[ \int_{\Theta} p(\theta|y,d) \ln p(y|\theta,d) \, d\theta \right]^2 dy, \qquad (\tilde{U}_{\mu_2,3}(d))
\end{aligned}
$$

where the second equality is from applying Bayes' rule, the third equality results from expanding the square term, and the last equality is from applying Bayes' rule on the second term. The three terms in Eqn. (4.9) are denoted as $\tilde{U}_{\mu_2,1}$, $\tilde{U}_{\mu_2,2}$ and $\tilde{U}_{\mu_2,3}$ respectively.

By expanding the utility variance in this manner, we can approximate its value by estimating $U(d)$, $\tilde{U}_{\mu_2,1}(d)$, $\tilde{U}_{\mu_2,2}(d)$ and $\tilde{U}_{\mu_2,3}(d)$ individually. We will introduce the estimation techniques in Sec. 4.2.1.

### 4.1.3 Variance-penalized robust design criterion

We introduce the mean-plus-variance rOED objective that seeks to maximize the expected utility while penalizing a large utility variance:

$$
U_\lambda(d) = U(d) - \lambda \tilde{U}(d), \qquad (4.10)
$$

where $\lambda \in \mathcal{R}$ is a hyperparameter that reflects the relative importance between the expectation and variance terms, or user preference on how robust the design to be. For example, a larger positive $\lambda$ yields a more robust design and may be suitable for risk-averse situations, although likely at the cost of a lower expected utility. For a robust design, $\lambda$ should remain non-negative; however, one can choose negative $\lambda$ to achieve aggressive, risk-seeking designs. When $\lambda = 0$, Eqn. (4.10) simplifies

to the non-robust OED criterion. It is worth noting that Eqn. (4.10) can be rewritten as

$$\begin{aligned} U_\lambda(d) &= \mathbb{E}_{y|d}\left[u(d,y)\right] - \lambda \mathbb{E}_{y|d}\left\{\left[u(d,y) - U(d)\right]^2\right\} \\ &= \mathbb{E}_{y|d}\left\{u(d,y) - \lambda\left[u(d,y) - U(d)\right]^2\right\}. \end{aligned}$$

Therefore, if we treat $u(d,y) - \lambda\left[u(d,y) - U(d)\right]^2$ as a new utility function $u_\lambda(d,y)$, then $U_\lambda(d)$ still fits the general form of OED objective in Eqn. (4.3) (with the understanding that here $u(d,y,\theta) = u(d,y)$).

The rOED problem entails solving the following optimization problem:

$$d^*_{U_\lambda} = \arg\max_{d\in\mathcal{D}} U_\lambda(d). \tag{4.11}$$

For $\lambda > 0$, both the expected utility and utility variance at $d^*_{U_\lambda}$ are lower than at $d^*_U$.

## 4.2 Numerical methods for rOED

### 4.2.1 Monte Carlo estimator

The expected utility in Eqn. (4.5), as well as the variance-penalized form in Eqn. (4.10), generally cannot be evaluated in closed-form. We propose to estimate these quantities numerically using MC sampling. In the following sections, we will introduce the MC estimator for each term in the variance-penalized objective, and then combine them together to reach a complete MC estimator for the rOED criterion.

#### 4.2.1.1 Estimation of $U(d)$ and $U(d)^2$

The estimation of $U(d)$ is using the double-nested MC estimator proposed by [124]. Using Bayes' rule, Eqn. (4.5) can be rewritten as

$$U(d) = \int_\Theta p(\theta) \int_{\mathcal{Y}} p(y|\theta,d) \ln \frac{p(y|\theta,d)}{p(y|d)}\, dy\, d\theta, \tag{4.12}$$

and a nested MC estimator can be formed as

$$\hat{U}^{N,M_1}(d) = \frac{1}{N}\sum_{i=1}^{N}\left\{\ln p(y^{(i)}|\theta^{(i)},d) - \ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y^{(i)}|\theta^{(i,j)},d)\right]\right\}, \tag{4.13}$$

where $\theta^{(i)}$ are drawn from the prior $p(\theta)$, $y^{(i)}$ are drawn from the likelihood $p(y|\theta = \theta^{(i)}, d)$, $\theta^{(\cdot,j)}$ are again drawn from the prior $p(\theta)$, and $N$ and $M_1$ are the numbers of outer loop samples ($\theta^{(i)}$ and $y^{(i)}$) and inner loop samples ($\theta^{(\cdot,j)}$), respectively. $\hat{U}^{N,M_1}(d)$ is a biased estimator due to the inner loop MC estimator for $p(y|d) = \int_\Theta p(y|\theta, d)p(\theta)\, d\theta$, but it is asymptotically unbiased as $M_1 \to \infty$. The variance of this estimator is

$$\mathbb{V}\left[\hat{U}^{N,M_1}(d)\right] \approx \frac{A_1(d)}{N} + \frac{B_1(d)}{NM_1}, \tag{4.14}$$

and the bias is

$$\mathbb{E}\left[\hat{U}^{N,M_1}(d) - U(d)\right] \approx \frac{E_1(d)}{M_1}, \tag{4.15}$$

where $A_1$, $B_1$ and $E_1$ are problem-specific constants that depend on the design variables, as well as prior and likelihood distributions [124, 72]. The variance of this estimator is dominated by $N$, while the bias is by $M_1$. [72] also suggests reusing outer samples as the inner samples (i.e., $\theta^{(i,\cdot)} = \theta^{(\cdot)}$), which reduces the forward model evaluations from $O(NM_1)$ to $O(N)$ for a given $d$, with the implication that $N = M_1$. Reusing samples introduces additional bias but the effect is small, and it brings the additional benefits of avoiding arithmetic underflow for the estimation of log evidence when sample size is small. For the following sections, we will present the estimator with independent outer samples and inner samples, but reuse samples in the code implementation.

Building upon the estimator of $U(d)$, a MC estimator of $U(d)^2$ is simply $\hat{U}^{N,M_1}(d)^2$. Appendix C.1 provides a derivation that the variance of $\hat{U}^{N,M_1}(d)^2$ is

$$\mathbb{V}\left[\hat{U}^{N,M_1}(d)^2\right] \approx \frac{A_2(d)}{N} + \frac{B_2(d)}{NM_1}, \tag{4.16}$$

and the bias is

$$\mathbb{E}\left[\hat{U}^{N,M_1}(d)^2 - U(d)^2\right] \approx \frac{D_2(d)}{N} + \frac{E_2(d)}{M_1}, \tag{4.17}$$

Different from $\hat{U}^{N,M_1}(d)$, the bias of $\hat{U}^{N,M_1}(d)^2$ is also controlled by the number of outer samples $N$, in addition to the number of inner samples $M_1$.

#### 4.2.1.2 Estimation of $\tilde{U}_{\mu_2}(d)$

For the estimation of $\tilde{U}_{\mu_2}(d)$, we evaluate its three parts $\tilde{U}_{\mu_2,1}(d)$, $\tilde{U}_{\mu_2,2}(d)$ and $\tilde{U}_{\mu_2,3}(d)$ separately, and then add them up to form an estimator for $\tilde{U}_{\mu_2}(d)$.

The MC estimator of $\tilde{U}_{\mu_2,1}(d)$ is

$$\hat{\tilde{U}}^{N,M_1}_{\mu_2,1}(d) = \frac{1}{N}\sum_{i=1}^{N}\left\{\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y^{(i)}|\theta^{(i,j)},d)\right]\right\}^2, \tag{4.18}$$

where $\theta^{(\cdot,j)}$ are drawn from the prior $p(\theta)$, and $y^{(i)} \sim p(y|d)$ are drawn by sampling $\theta^{(i)}$ from the prior $p(\theta)$, and then sampling $y^{(i)}$ from the likelihood $p(y|\theta=\theta^{(i)},d)$. Having sample pairs $(\theta^{(i)}, y^{(i)})$ from the joint distribution $p(\theta, y|d)$, we can ignore $\theta^{(i)}$ samples, and the remaining $y^{(i)}$ become the samples from the marginal distribution $p(y|d)$. Appendix C.2 provides a derivation to show that the variance of $\hat{\tilde{U}}^{N,M_1}_{\mu_2,1}(d)$ is proportional to $\frac{A_3(d)}{N} + \frac{B_3(d)}{NM_1}$ while the bias is proportional to $\frac{E_3(d)}{M_1}$.

The MC estimator of $\tilde{U}_{\mu_2,2}(d)$ is

$$\hat{\tilde{U}}^{N,M_1}_{\mu_2,2}(d) = -\frac{2}{N}\sum_{i=1}^{N}\left\{\ln p(y^{(i)}|\theta^{(i)},d)\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y^{(i)}|\theta^{(i,j)},d)\right]\right\} \tag{4.19}$$

where the sampling of $\theta^{(i)}$, $y^{(i)}$ and $\theta^{(i,j)}$ is the same as $\hat{U}^{N,M_1}(d)$. Appendix C.3 provides a derivation to show that the variance of $\hat{\tilde{U}}^{N,M_1}_{\mu_2,2}(d)$ is proportional to $\frac{A_4(d)}{N} + \frac{B_4(d)}{NM_1}$ and the bias is proportional to $\frac{E_4(d)}{M_1}$.

The estimation of $\tilde{U}_{\mu_2,3}(d)$ is more difficult, because the inner part of $\tilde{U}_{\mu_2,3}(d)$ requires sampling from the posterior $p(\theta|y,d)$. Posterior sampling can be achieved by Markov chain Monte Carlo (MCMC) but may become very expensive since $N$ MCMC chains would be needed for each $d$. Instead, we can apply Bayes' rule to $p(\theta|y,d)$ in $\tilde{U}_{\mu_2,3}(d)$ and rewrite it as

$$\begin{aligned}\tilde{U}_{\mu_2,3}(d) &= \int_{\mathcal{Y}} p(y|d)\left[\int_{\Theta} p(\theta)\frac{p(y|\theta,d)}{p(y|d)}\ln p(y|\theta,d)\,d\theta\right]^2 dy \\ &= \int_{\mathcal{Y}} p(y|d)\left[\frac{1}{p(y|d)}\int_{\Theta} p(\theta)p(y|\theta,d)\ln p(y|\theta,d)\,d\theta\right]^2 dy,\end{aligned} \tag{4.20}$$

with the corresponding MC estimator

$$\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d) = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{M_1}{\sum_{j=1}^{M_1} p(y^{(i)}|\theta^{(i,j)},d)}\frac{1}{M_2}\sum_{k=1}^{M_2} p(y^{(i)}|\theta^{(i,k)},d)\ln p(y^{(i)}|\theta^{(i,k)},d)\right\}^2, \tag{4.21}$$

where $y^{(i)}$ are drawn from the marginal likelihood $p(y|d)$ in the same way as $\hat{\tilde{U}}^{N,M_1}_{\mu_2,1}(d)$, and $\theta^{(\cdot,j)}$ and $\theta^{(\cdot,k)}$ are independently drawn from the prior $p(\theta)$. Appendix C.4 provides a derivation to show that the variance of $\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d)$ is proportional to $\frac{A_5(d)}{N} + \frac{B_5(d)}{NM_1} + \frac{C_5(d)}{NM_2}$, and the bias is proportional

to $\frac{E_5(d)}{M_1} + \frac{F_5(d)}{M_2}$.

Combing the three parts together results in the overall MC estimator of $\tilde{U}_{\mu_2}(d)$:

$$\begin{aligned}\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2}(d) &= \hat{\tilde{U}}^{N,M_1}_{\mu_2,1}(d) + \hat{\tilde{U}}^{N,M_1}_{\mu_2,2}(d) + \hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d) \\ &= \frac{1}{N}\sum_{i=1}^{N}\left\{\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y^{(i)}|\theta^{(i,j)},d)\right]\right\}^2 \\ &\quad - \frac{2}{N}\sum_{i=1}^{N}\left\{\ln p(y^{(i)}|\theta^{(i)},d)\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y^{(i)}|\theta^{(i,j)},d)\right]\right\} \\ &\quad + \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{M_1}{\sum_{j=1}^{M_1} p(y^{(i)}|\theta^{(i,j)},d)}\frac{1}{M_2}\sum_{k=1}^{M_2} p(y^{(i)}|\theta^{(i,k)},d)\ln p(y^{(i)}|\theta^{(i,k)},d)\right\}^2,\end{aligned} \tag{4.22}$$

whose bias is the summation of the biases of three parts:

$$\mathbb{E}\left[\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2}(d) - \tilde{U}_{\mu_2}(d)\right] \approx \frac{E_3(d)+E_4(d)+E_5(d)}{M_1} + \frac{F_5(d)}{M_2} = \frac{E_6(d)}{M_1} + \frac{F_6(d)}{M_2}, \tag{4.23}$$

where $E_6(d) = E_3(d)+E_4(d)+E_5(d)$ and $F_6(d) = F_5(d)$. To approximate the variance of $\tilde{U}_{\mu_2}(d)$, we assume that $\hat{\tilde{U}}^{N,M_1}_{\mu_2,1}(d)$, $\hat{\tilde{U}}^{N,M_1}_{\mu_2,2}(d)$ and $\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d)$ are independent (this can be achieved by resampling for each estimator, although in practice we will use the same samples across different estimators), then the variance of $\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2}(d)$ is

$$\begin{aligned}\mathbb{V}\left[\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2}(d)\right] &= \mathbb{V}\left[\hat{\tilde{U}}^{N,M_1}_{\mu_2,1}(d)\right] + \mathbb{V}\left[\hat{\tilde{U}}^{N,M_1}_{\mu_2,2}(d)\right] + \mathbb{V}\left[\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d)\right] \\ &\approx \frac{A_3(d)+A_4(d)+A_5(d)}{N} + \frac{B_3(d)+B_4(d)+B_5(d)}{NM_1} + \frac{C_5(d)}{NM_2} \qquad (4.24) \\ &= \frac{A_6(d)}{N} + \frac{B_6(d)}{NM_1} + \frac{C_6(d)}{NM_2}, \qquad (4.25)\end{aligned}$$

where $A_6(d) = A_3(d)+A_4(d)+A_5(d)$, $B_6(d) = B_3(d)+B_4(d)+B_5(d)$ and $C_6(d) = C_5(d)$.

#### 4.2.1.3 Estimation of $\tilde{U}(d)$

Following Eqn. (4.7), the estimation of the utility variance $\tilde{U}(d)$ can be realized by combining the estimator of $\tilde{U}_{\mu_2}(d)$ and $U(d)^2$:

$$\hat{\tilde{U}}^{N,M_1,M_2}(d) = \hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2}(d) - \hat{U}^{N,M_1}(d)^2 \tag{4.26}$$

whose variance is (by assuming independence between theestimators)

$$\begin{aligned}\mathbb{V}\left[\hat{\tilde{U}}^{N,M_1,M_2}(d)\right] &= \mathbb{V}\left[\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2}(d)\right] + \mathbb{V}\left[\hat{U}^{N,M_1}(d)^2\right] \\ &\approx \frac{A_7(d)}{N} + \frac{B_7(d)}{NM_1} + \frac{C_7(d)}{NM_2},\end{aligned} \tag{4.27}$$

where $A_7(d) = A_2(d) + A_6(d)$, $B_7(d) = B_2(d) + B_6(d)$, $C_7(d) = C_6(d)$, and the bias is

$$\mathbb{E}\left[\hat{\tilde{U}}^{N,M_1,M_2}(d) - \tilde{U}(d)\right] \approx \frac{D_7(d)}{N} + \frac{E_7(d)}{M_1} + \frac{F_7(d)}{M_2}, \tag{4.28}$$

where $D_7(d) = -D_2(d)$, $E_7(d) = E_6(d) - E_2(d)$, and $F_7(d) = F_6(d)$.

#### 4.2.1.4 Estimation of $U_\lambda(d)$

Following Eqn. (4.10), the estimation of the variance-penalized objective $U_\lambda(d)$ can be achieved by

$$\begin{aligned}\hat{U}^{N,M_1,M_2}_\lambda(d) &= \hat{U}^{N,M_1}(d) - \lambda\hat{\tilde{U}}^{N,M_1,M_2}(d) \\ &= \hat{U}^{N,M_1}(d) - \lambda(\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2}(d) - \hat{U}^{N,M_1}(d)^2) \\ &= \hat{U}^{N,M_1}(d)\left[1 + \lambda\hat{U}^{N,M_1}(d)\right] - \lambda\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2}(d),\end{aligned} \tag{4.29}$$

where the estimator $\hat{U}^{N,M_1}(d)$ can be referred to Eqn. (4.13), and $\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2}(d)$ to Eqn. (4.22). The variance of this complete estimator is (by assuming independence between $\hat{U}^{N,M_1}(d)$ and $\hat{\tilde{U}}^{N,M_1,M_2}(d)$)

$$\begin{aligned}\mathbb{V}\left[\hat{U}^{N,M_1,M_2}_\lambda(d)\right] &= \mathbb{V}\left[\hat{U}^{N,M_1}(d)\right] + \lambda^2\mathbb{V}\left[\hat{\tilde{U}}^{N,M_1,M_2}(d)\right] \\ &\approx \frac{\lambda^2A_8(d) + A_9(d)}{N} + \frac{\lambda^2B_8(d) + B_9(d)}{NM_1} + \frac{\lambda^2C_8(d)}{NM_2},\end{aligned} \tag{4.30}$$

where $A_8(d) = A_7(d)$, $A_9(d) = A_2(d)$, $B_8(d) = B_7(d)$, $B_9(d) = B_2(d)$, $C_8(d) = C_7(d)$, and the bias is

$$\mathbb{E}\left[\hat{U}^{N,M_1,M_2}_\lambda(d) - U_\lambda(d)\right] \approx \frac{\lambda D_8(d)}{N} + \frac{\lambda E_8(d) + E_9(d)}{M_1} + \frac{\lambda F_8(d)}{M_2}, \tag{4.31}$$

where $D_8(d) = -D_7(d)$, $E_8(d) = -E_7(d)$, $E_9(d) = E_1(d)$, and $F_8(d) = -F_7(d)$. In practice, we will reuse outer $\theta^{(\cdot)}$ samples as inner $\theta^{(i,\cdot)}$ samples, which reduces the forward model evaluations from $O(N + NM_1 + NM_2)$ to $O(N)$; this requires setting $N = M_1 = M_2$. Both the bias and variance of variance-penalized objective estimator are approximately proportional to $O(\frac{1}{N})$ with

sample reuse. We will show the convergence of bias and variance of this estimator in Sec. 4.3.1. Notice that although $\hat{U}_\lambda^{N,M_1,M_2}(d)$ has the same order of bias and variance as $\hat{U}^{N,M_1}(d)$, the bias and variance of $\hat{U}_\lambda^{N,M_1,M_2}(d)$ will be higher than those of $\hat{U}^{N,M_1}(d)$ with the same number of samples, due to its larger constants. Moreover, we want to emphasize that estimating variance-penalized objective has the same order of time complexity as estimating the expected utility by double-nested MC since the former only uses quantities that have already been calculated in the latter.

### 4.2.2 Bayesian optimization

Equipped with the ability to estimate the objective function $U_\lambda(d)$ with MC sampling, we can now attempt to solve the optimization problem in Eqn. (4.11). Naïve optimization techniques such as grid search and random search would be highly expensive and do not scale well to multi-dimensional design spaces. More efficient and intelligent optimization algorithms are desired. One approach is to extract the gradient of $U_\lambda(d)$ either analytically or numerically, and then apply gradient-based or quasi-Newton optimization methods such as gradient ascent and L-BFGS-B. When gradient information is unavailable, one must adopt a derivative-free method such as simultaneous perturbation stochastic approximation (SPSA) and Nelder–Mead nonlinear simplex (NMNS) [72]. However, SPSA is sensitive to the MC noise in the objective estimation, while NMNS may converge slowly even for smooth functions. Moreover, all before-mentioned optimization methods are prone to getting stuck in a local optimum and have no guarantee to find the global optimum. Global optimization methods like simulated annealing and genetic algorithm also tend to be costly and require many iterations to converge.

We propose to use Bayesian optimization (BO) to solve the optimization problem in Eqn. (4.11). BO is a derivative-free global optimization method that is sample-efficient and also robust to noisy objective functions [23, 134, 128, 115]. It is particularly suitable for expensive objective evaluations (this is the case here since each MC estimate of $U_\lambda(d)$ requires many repeated forward model evaluations especially to estimate the utility variance accurately) and the optimization domain is recommended to be less than 20 dimensions [58, 107]. BO constructs and updates a surrogate Gaussian process (GP) of the objective function $U_\lambda(d)$ based on previous evaluations of the objective. The GP provides mean and variance *of the objective* (i.e. of $U_\lambda(d)$) resulting from not yet having evaluated the objective in the optimization process—notably, this is different from the mean and variance *of the utility* stemming from the uncertainty in the model. In other words, the uncertainty portrayed by the GP can be reduced by additional evaluations of the objective; the uncertainty of the utility can only be reduced by performing more experiments. From the objective GP, BO then chooses the next point to evaluate the objective by maximizing the acquisition function, for example by considering both GP mean and variance so as to balance the exploration

and exploitation to better identify the global optimum. Overall, we emphasize that BO itself is not OED, it is a tool for solving the optimization problem in Eqn. (4.11) and is agnostic to the OED utility formulations.

Each iteration of BO can be summarized by three steps: updating the GP by performing GP regression, forming the acquisition function, and finding the next objective evaluation location by maximizing the acquisition function. We will describe these steps in detail below.

#### 4.2.2.1 Gaussian process regression

A GP is a stochastic process (i.e., a collection of random variables) where any finite sub-collection of those random variables has a multivariate Gaussian distribution. Consider the task of inferring the variance-penalized objective function $U_\lambda(d) : \mathcal{R}^{N_d} \to \mathcal{R}$ whose evaluation is noisy (due to using a MC estimator):

$$\hat{U}_\lambda^{N,M_1,M_2}(d) = U_\lambda(d) + \eta, \tag{4.32}$$

where $\eta \sim \mathcal{N}(0, \sigma_\eta^2)$ models the error of the estimator. If we use a GP to describe $U_\lambda(d)$:

$$U_\lambda(\cdot) \sim \mathcal{GP}(\mu(\cdot), k(\cdot,\cdot)), \tag{4.33}$$

that means every collection of random variables $\{U_\lambda(d) : d \in \mathcal{D}\}$ follow a multivariate Gaussian distribution. More specifically, for a finite set of $d_1, \ldots, d_m \in \mathcal{D}$, the corresponding $U_\lambda(d_1), \ldots, U_\lambda(d_m)$ have the following Gaussian distribution:

$$\begin{bmatrix} U_\lambda(d_1) \\ \vdots \\ U_\lambda(d_m) \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \mu(d_1) \\ \vdots \\ \mu(d_m) \end{bmatrix}, \begin{bmatrix} k(d_1,d_1) & \cdots & k(d_1,d_m) \\ \vdots & \ddots & \vdots \\ k(d_m,d_1) & \cdots & k(d_m,d_m) \end{bmatrix} \right), \tag{4.34}$$

where the mean function $\mu(\cdot)$ is usually selected, without loss of generality, to be 0 to reach a zero-mean GP (this amounts to centering the data). A typical choice for the covariance kernel function $k(\cdot,\cdot)$ is the Radial basis function (i.e., square-exponential) kernel:

$$k(d,d') = \exp\left( -\frac{\| d - d' \|_2^2}{2l^2} \right), \tag{4.35}$$

where $l$ is a hyperparameter to control the width of the kernel.

When we have already collected noisy estimations $\overrightarrow{\hat{U}_\lambda^{N,M_1,M_2}}$ at seveal different design locations

$D$, then $\hat{U}_\lambda^{N,M_1,M_2}(d)$ at a new design $d$ follows a Gaussian distribution jointly with $\overrightarrow{\hat{U}_\lambda^{N,M_1,M_2}}$:

$$\begin{bmatrix} \overrightarrow{\hat{U}_\lambda^{N,M_1,M_2}} \\ \hat{U}_\lambda^{N,M_1,M_2}(d) \end{bmatrix} = \begin{bmatrix} \overrightarrow{U_\lambda} \\ U_\lambda(d) \end{bmatrix} + \begin{bmatrix} \overrightarrow{\eta} \\ \eta \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(D,D)+\sigma_\eta^2 I & K(D,d) \\ K(d,D) & k(d,d)+\sigma_\eta^2 \end{bmatrix}\right). \tag{4.36}$$

Since $\overrightarrow{\hat{U}_\lambda^{N,M_1,M_2}}$ is known, we can get the probability distribution of $U_\lambda(d)$ by conditioning on $\overrightarrow{\hat{U}_\lambda^{N,M_1,M_2}}$, which is also a Gaussian distribution, with the updated mean

$$\mu(d|D, \overrightarrow{\hat{U}_\lambda^{N,M_1,M_2}}) = K(d,D)\left(K(D,D)+\sigma_\eta^2 I\right)^{-1} \overrightarrow{\hat{U}_\lambda^{N,M_1,M_2}}, \tag{4.37}$$

and the standard deviation

$$\sigma(d|D, \overrightarrow{\hat{U}_\lambda^{N,M_1,M_2}}) = k(d,d) - K(d,D)\left(K(D,D)+\sigma_\eta^2 I\right)^{-1} K(D,d). \tag{4.38}$$

#### 4.2.2.2 Acquisition function

The acquisition function at a given location $d$ reflects the benefit of taking the next measurement at $d$ towards the task of solving the optimization problem. It is usually formulated by using accessible information from the GP (e.g., its mean and standard deviation) in order to balance the exploration (i.e., take measurements at highly uncertain regions in case the under-explored regions contain the global optimum) and exploitation (i.e., take measurements at good regions that are already known to refine the results). Many popular choices for the acquisition function are available, such as the probability of improvement (PI), expected improvement (EI) and upper confidence bound (UCB). For example, UCB is formulated as

$$a(d) = \mu(d|D) + \kappa\sigma(d|D), \tag{4.39}$$

where $\kappa$ is a hyperparameter that controls the degree of exploration. We adopt UCB in this work.

#### 4.2.2.3 Optimization of the acquisition function

The next objective evaluation location $d'$ is selected as the maximum point of the acquisition function:

$$d' = \arg\max_{d\in\mathcal{D}} a(d). \tag{4.40}$$

Since the acquisition function is very inexpensive to evaluate by design, this inner optimization problem can be solved easily using existing optimization packages. After obtaining the next measurement location, we then evaluate the objective function value $\hat{U}_\lambda^{N,M_1,M_2}(d')$, augment the data set of evaluated objectives $D$, and repeat the process, until we reach the stopping criterion (e.g., maximum number of iterations, change in objective below threshold).

In this work, we use an existing BO Python package [113], with some small modifications to make it applicable for more complex constraints.

### 4.2.3 Common random samples

Although BO tolerates noisy objective evaluations, its convergence may be slowed by high noise in estimating the variance-penalized objective. To mitigate this objective noise without simply adding more MC samples, we adopt the technique of common random samples: that is, reusing the same $\theta$ samples and noise samples across different designs $d$. This technique effectively introduces artificial correlation between the objective noise across different designs (a "synchronized randomization") so as to make the objective function smoother and easier to optimize. Note that common random samples is different from the before-mentioned reuse technique in Sec. 4.2.1.4, where the latter entails reusing outer loop $\theta$ samples as inner loop $\theta$ samples when estimating $U_\lambda(d)$ to reduce forward model evaluations and avoid arithmetic underflow. Using common random samples is equivalent to resetting the random seed of the random number generator every time when we start MC sampling to estimate the objective function; further discussions can be found in [73].

However, we want to emphasize that although using common random samples will introduce more variability to the arg-max under a finite sample size, it is still worth using since the variability becomes negligible with a decent amount of samples, and the resulting objective function is much smoother than not using common random samples. We will show the comparison between using and not using common random samples in Sec. 4.3.1 and Sec. 4.3.2.

## 4.3 Numerical results and discussions

We present three examples to illustrate the benefits of rOED. The first example is a linear Gaussian problem (Sec. 4.3.1) which has a closed-form solution and serves as a validation benchmark. We will compare the numerical results from our estimator with the analytical solution, to show the convergence and accuracy of the estimator. The second case is a synthetic nonlinear case (Sec. 4.3.2), in which we will show the benefits of considering robustness of design. We will then apply rOED to a 2D contaminant source inversion case (Sec. 4.3.3 and Sec. 4.3.4), to demonstrate its usage in a more realistic multi-dimensional physical problem.

### 4.3.1 Linear-Gaussian benchmark

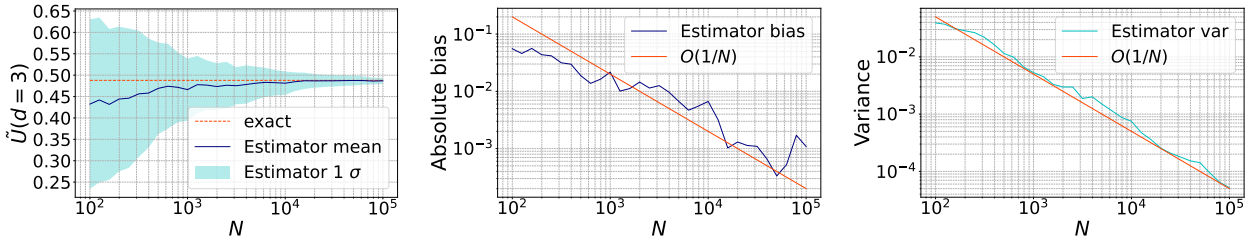
Consider an observation model with a forward model that is linear with respect to its parameter $\theta \in \mathcal{R}$:

$$y(\theta, d) = G(\theta, d) + \epsilon = \theta d + \epsilon. \tag{4.41}$$

The prior on $\theta$ is $\mathcal{N}(0, 3^2)$, and the design domain is $d \in [0, 3]$. Due to the linearity in the forward model and the Gaussian prior and noise, the posterior on $\theta$ is analytically Gaussian.

The variance-penalized objective estimator $\hat{U}_\lambda^{N,M_1,M_2}$ can be decomposed into two components: the expected utility estimator $\hat{U}^{N,M_1}$ and the utility variance estimator $\hat{\tilde{U}}^{N,M_1,M_2}$. The performance of $\hat{U}^{N,M_1}$ has been discussed in detail in [72], therefore in this example we focus on the utility variance estimator $\hat{\tilde{U}}^{N,M_1,M_2}$.

We first investigate the change of bias and variance of $\hat{\tilde{U}}^{N,M_1,M_2}(d)$ at a fixed $d$ as the sample number increases. We pick $d = 3$, which corresponds to the largest utility variance and also the largest estimation error. Note that $N = M_1 = M_2$ is always implied since we reuse outer samples as inner samples in our implementation. Fig. 4.1 shows the performance of our MC estimator as the sample number increases, including the comparison between the estimate and the exact solution (Fig. 4.1a), the absolute bias of the estimator against sample number (Fig. 4.1b), and the variance of the estimator against sample number (Fig. 4.1c). In order to evaluate the bias and the variance of the estimator, we repeat the computation 50 times for each $N$. Both the bias and the variance decrease approximately at the order of $O(\frac{1}{N})$, agreeing with the convergence analysis in Sec. 4.2. The absolute bias in Fig. 4.1b has more fluctuation than the variance in Fig. 4.1c. This is possibly a result of the sample reuse technique inducing more fluctuations to the additional terms in the bias estimator convergence rate (leading terms only) $O(\frac{A}{N} + \frac{B}{M_1} + \frac{C}{M_2})$ versus the variance estimator convergence rate $O(\frac{1}{N})$).

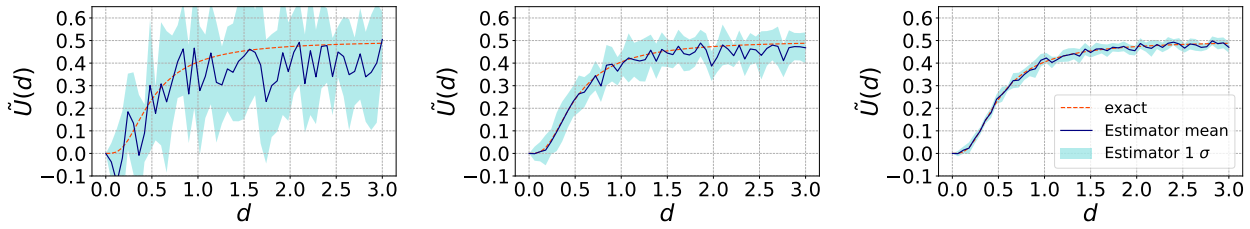



(a) Estimate versus exact (b) Estimator absolute bias (c) Estimator variance

Figure 4.1: The performance of MC estimator estimating the utility variance $\tilde{U}(d = 3)$ as the sample number increases for 1D linear Gaussian case.

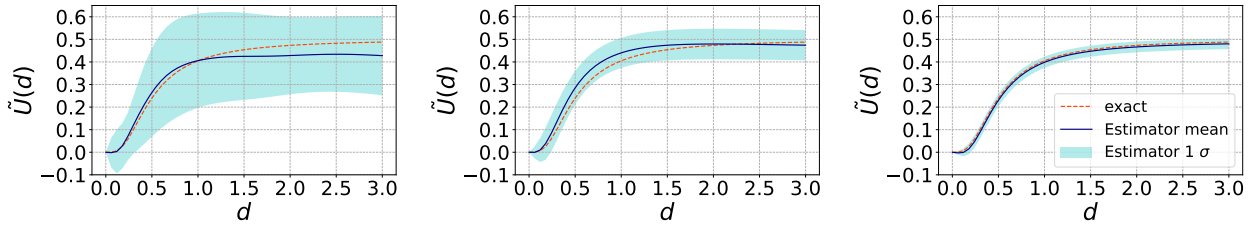
We then investigate the effect of using common random samples as mentioned in Sec. 4.2.3.

Fig. 4.2 and Fig. 4.3 show the comparison between not using and using common random samples. The estimator has been rerun 10 times to obtain the estimator mean and variance at each $d$. As expected, using common random samples results in a much smoother objective function compared to its counterpart. Although using common random samples will result in a slightly different objective function, the shift is negligible with a sufficient amount of samples, and the shift can be fully compensated by the benefits of additional smoothness. Therefore, we will always use common random samples in this chapter, and we will further show a 2D example in Sec. 4.3.2 to reinforce this conclusion.




(a) $N = 100$ (b) $N = 1000$ (c) $N = 10000$

Figure 4.2: The estimated utility variance $\tilde{U}(d)$ when **not** using common random samples for 1D linear Gaussian case.



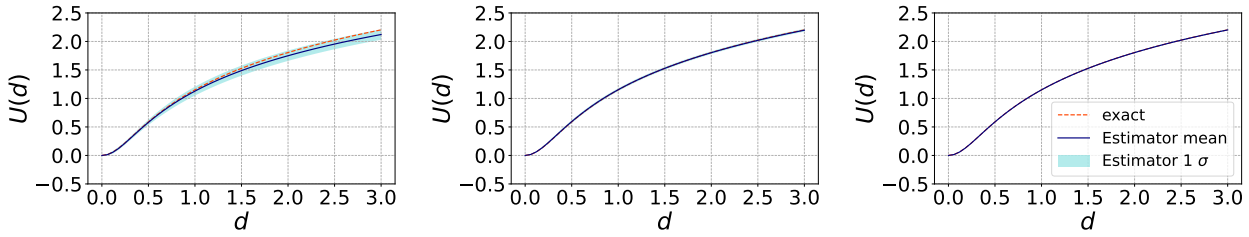

(a) $N = 100$ (b) $N = 1000$ (c) $N = 10000$

Figure 4.3: The estimated utility variance $\tilde{U}(d)$ when using common random samples for 1D linear Gaussian case.

Figure 4.4 shows the comparison between the estimated expected utility using common random samples and the exact expected utility under different sample sizes. The estimated expected utility quickly converges to the exact solution with high accuracy with just $N = 1000$ samples.

From Fig. 4.3 and 4.4, we can find that the expected utility in the range of $d \in [0, 3]$ has a log shape and a continuing increasing trend, while the utility variance starts to stabilize beyond $d = 2$. That means with higher $d$, we will get a higher expected utility $U(d)$, but with almost non-increasing utility variance $\tilde{U}(d)$, and so higher valued $d$ is preferred in rOED for 1D linear

Gaussian problem for as long as $\lambda$ is not too large to make the variance term dominating; this is the same optimal design location as the non-robust OED formulation.




(a) $N = 100$ (b) $N = 1000$ (c) $N = 10000$

Figure 4.4: The comparison between the estimated expected utility and the exact expected utility under different sample sizes for 1D linear Gaussian case.

### 4.3.2 Nonlinear model

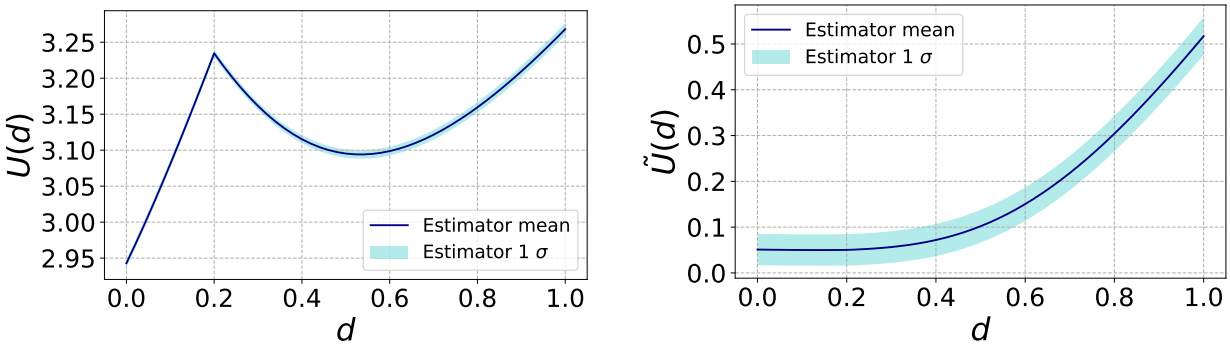
We adapt the nonlinear model used in [72] but with a slight modification:

$$\begin{aligned} y(\theta, d) &= G(\theta, d) + \epsilon \\ &= \theta^3 d^2 + \theta \exp(-1.3\,|0.2 - d|) + \epsilon, \end{aligned} \tag{4.42}$$

where $\theta \sim \mathcal{U}[0, 1]$ is the scalar unknown parameter, $\epsilon \sim \mathcal{N}(0, 10^{-4} I_{N_y})$ is an additive Gaussian noise, and we will consider 1D and 2D design spaces ($d \in [0, 1]$ and $d \in [0, 1]^2$, respectively). The observation space and additive Gaussian noise have the same dimension as the design space (i.e., $N_y = N_d$).

We first investigate the 1D design case. Figure 4.5 shows the estimated expected utility and estimated utility variance using $N = 10000$ samples and 10 reruns. Similar to the linear Gaussian case, the bias and variance of the variance estimator are larger than that of the expected utility estimator under the same sample size. Nevertheless, $N = 10000$ appears sufficient to solve the OED problem with utility variance. It is worth noting that if we only look at the expected utility (without variance), $U(d = 1.0)$ is slightly larger than $U(d = 0.2)$ and so $d = 1.0$ would be the non-robust OED optimal design. However, the variance at $d = 1.0$ is much higher than that at $d = 0.2$, which indicates that $d = 1.0$ is a risky design when considering the utility variance. We thus draw the $U_\lambda(d)$ in Fig. 4.6 with $\lambda$ equals 0.2 and 1 respectively. When $\lambda = 0.2$, the rOED objective at $d = 0.2$ is already better than $d = 1$; as $\lambda$ increases to 1, the advantage of $d = 0.2$ is even more prominent, and $d = 1$ in fact becomes the worst design in the entire design space.

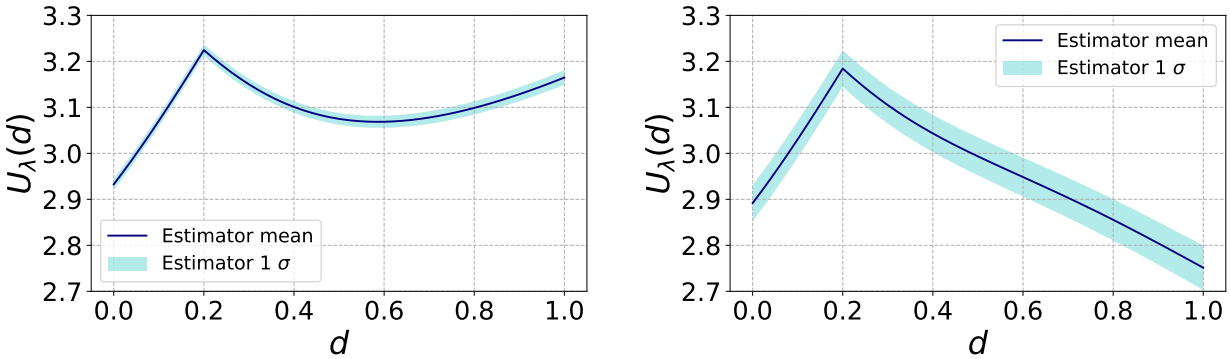
To further investigate the difference of utility variance between $d = 0.2$ and $d = 1$, we plot




(a) Expected utility

(b) Utility variance

Figure 4.5: Estimated expected utility and utility variance for 1D nonlinear case.



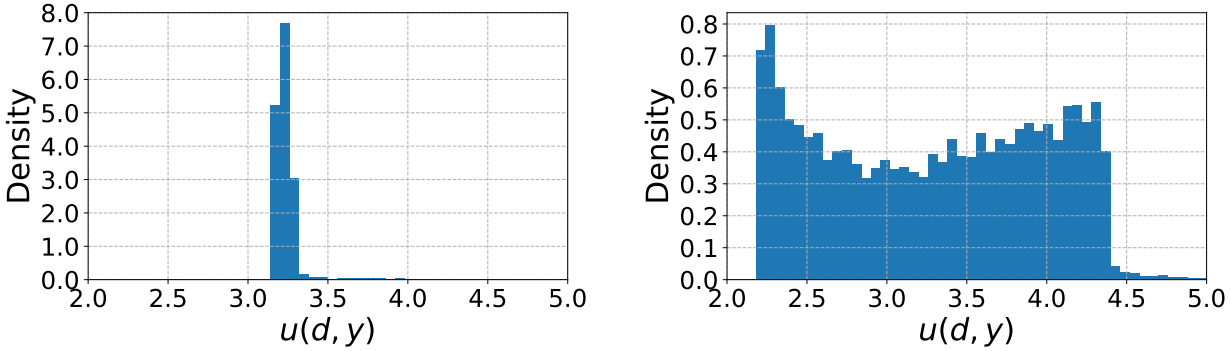

(a) $\lambda = 0.2$

(b) $\lambda = 1$

Figure 4.6: $U_\lambda(d)$ versus $d$ with $\lambda = 0.2$ and $\lambda = 1$ for 1D nonlinear case.

the histogram of $u(d, y)$ under $d = 0.2$ and $d = 1$ in Fig. 4.7, where $u(d, y)$ is computed by grid discretization on $\theta$ space. $u(d = 0.2, y)$ is quite stable (low spread), while $u(d = 1, y)$ varies almost uniformly between 2.2 and 4.4, with greater potential of getting much higher or lower utility values. We then draw scatter plot of $u(d, y)$ and $y$ in Fig. 4.8, where we see $d = 0.2$'s low variance is supported by $u(d, y)$ not changing much for large parts of the $y$ value, while $d = 1.0$'s high variance can be seen by its $u(d, y)$ being more sensitive to $y$. Sample posteriors conditioned on observing $y = 0.03$ (low utility for $d = 1$) or $y = 1$ (high utility for $d = 1$) for both $d = 0.2$ and $d = 1$ in Fig. 4.9, from which we can find that $y = 0.03$ and $y = 1$ results in similar posterior uncertainties for $d = 0.2$, but significant different posterior uncertainties for $d = 1.0$, further supporting the robustness of $d = 0.2$.

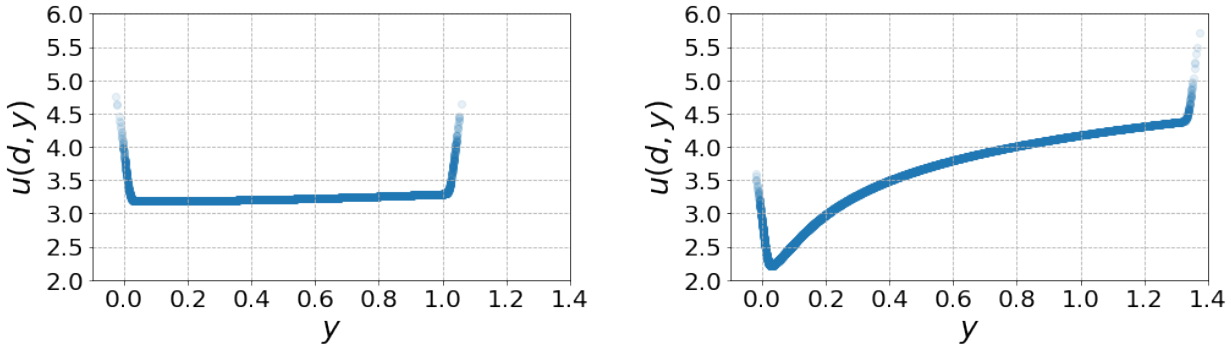
To fully explain why $d = 1$ has a higher variance compared to $d = 0.2$, we need to go back to the forward model in Eqn. (4.42). When $d = 0.2$, the forward model is dominated by $\theta$; when $d = 1$, the dominating term is $\theta^3$. Plotting $G(\theta, d)$ as a function of $\theta$ at $d = 0.2$ and $d = 1$ respectively in Fig. 4.10, $G(\theta, d = 0.2)$ exhibits a linear shape and $G(\theta, d = 1)$ a cubic curve shape. As a heuristic, a higher slope of $G$ tend to suggest more information since the output of $G$ is more sensitive to the input. From these plots, the slope of $G(\theta, d = 0.2)$ is almost invariant at different $\theta$, while the slope of $G(\theta, d = 1)$ changes significantly. This difference helps explain why the utility $u(d = 0.2, y)$ is stable, while the utility $u(d = 1, y)$ has a large variation depending on whether the underlying $\theta$ is small (low slope region) or big (high slope region).




(a) $d = 0.2$ (b) $d = 1$

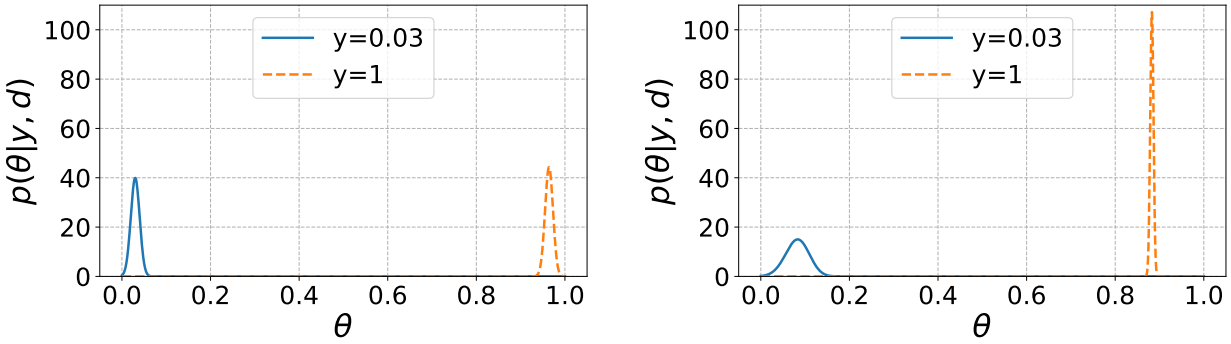
Figure 4.7: Histograms of $u(d, y)$ for $d = 0.2$ and $d = 1$ for 1D nonlinear case.

We then investigate the performance of BO. Figure 4.11 presents the updating history of Bayesian optimization when using common random samples, where Fig. 4.11a superimpose BO points upon the objective function and Fig. 4.11b plots the updating history of BO against the update step. We observe that BO quickly converges to the optimal design, while continue exploring the design space in order to find potentially better designs as exhibited by the dips in the BO updating history after




(a) $d = 0.2$ (b) $d = 1$

Figure 4.8: The scatter plots of $u(d, y)$ against $y$ at $d = 0.2$ and $d = 1$ for 1D nonlinear case.




(a) $d = 0.2$ (b) $d = 1$

Figure 4.9: The posterior $p(\theta|y, d)$ when $y = 0.03$ and $y = 1$, at $d = 0.2$ and $d = 1$ for 1D nonlinear case.

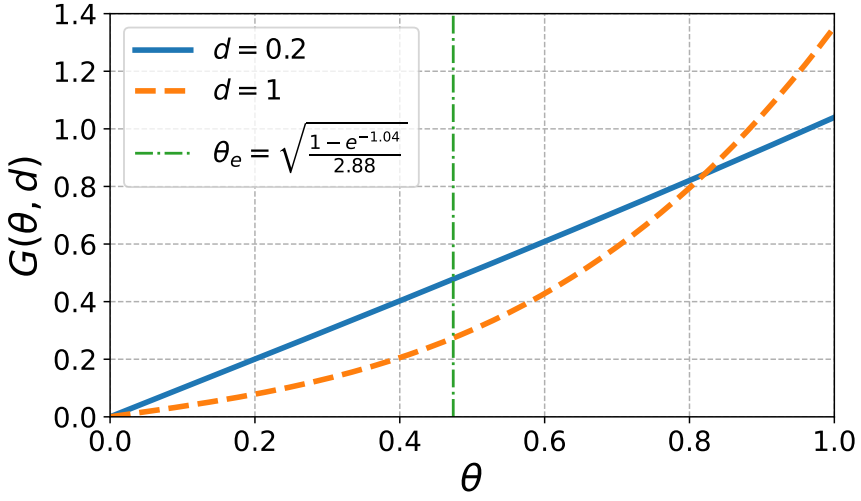
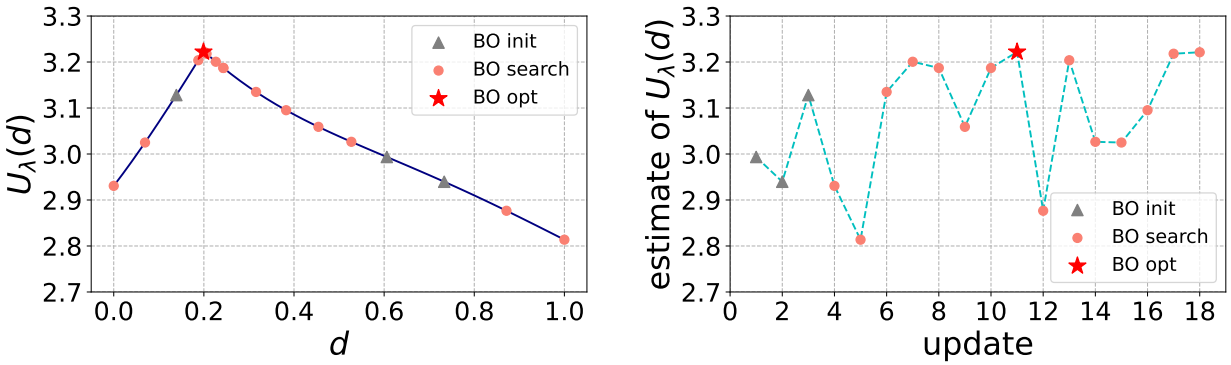



Figure 4.10: $G(\theta, d)$ versus $\theta$, at $d = 0.2$ and $d = 1$ for 1D nonlinear case.

reaching the optimal design.




(a) BO locations superimposed on the objective function

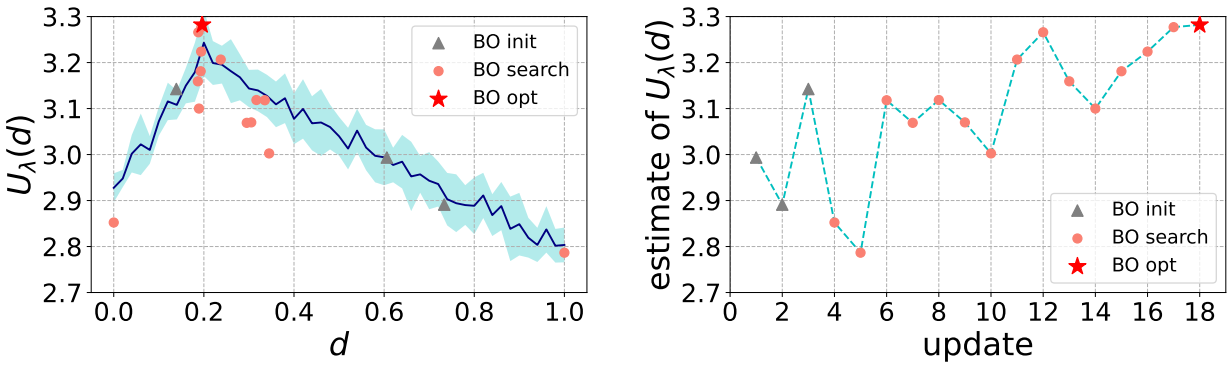
(b) BO history

Figure 4.11: Updating history of BO when using common random samples for the 1D nonlinear case, where dark blue curve is the estimated $U_\lambda(d)$ (i.e., objective function), grey triangles are the initial points of BO, orange circle are the search points of BO, and the red star is the optimal point of BO.

To demonstrate BO's ability to tolerate noisy objective evaluations, we apply it to the same problem as Fig. 4.11, but without using common random samples, plotting the results in Fig. 4.12. In order to make BO insensitive to noise, we set the standard deviation of the BO noise $\sigma_\eta$ (mentioned in Eqn. (4.32)) as 0.05. In this case, BO still finds the optimal design, but it has more exploitation searches around $d = 0.2$ because BO is not confident about this optimal point due to the high noise. As a consequence, BO has fewer exploration searches and would need more iterations to explore the under-explored region between $d = 0.4$ and $d = 1$, if there was a better design in this region.

Fig. 4.13 shows the contours of estimated expected utility, utility variance and variance-penalized objective when $\lambda = 1$. Apparently, $d = [0.2, 1]$ and $d = [1, 0.2]$ correspond to the highest expected utility, while $d = [0.2, 0.2]$ is the optimal robust design when $\lambda = 1$. Notice that using common random samples does introduce some small shift, which can be found by the asymmetry at the bottom left corner of Fig. 4.13b. Comparing Fig. 4.13 to Fig. 4.14, it is obvious that using common random samples does smoothen the objective function a lot.

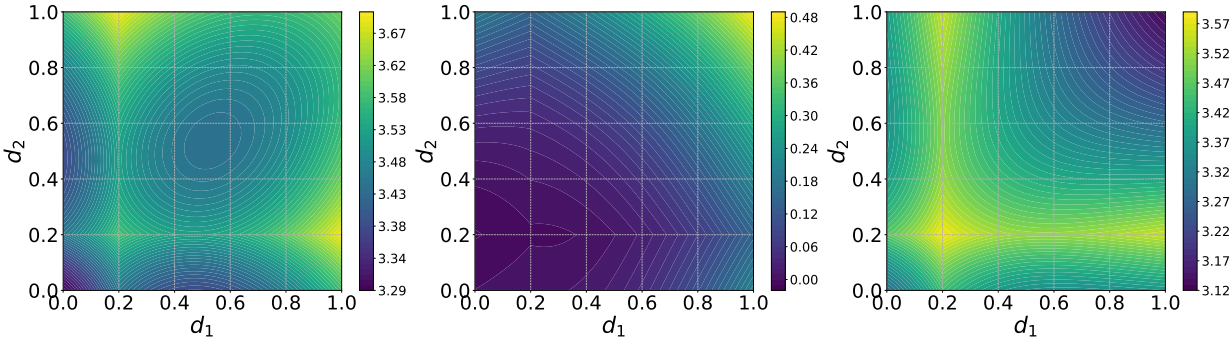
Fig. 4.15 presents the updating history of BO when using common random samples, from which we can find that BO successfully finds the global optimum $d^*_{U_\lambda} = [0.2, 0.2]$ against the other two local optimal points $d = [0.2, 1]$ and $d = [1.0.2]$. For this case, we are using $N = 10000$ samples and rerunning it for 10 times to get a more accurate estimate. It can be difficult to find the global optimum even for BO when the objective function is too noisy, as shown in Fig. 4.16. Although BO does realize that there is a cross pattern having higher objective values and places all BO points alongside this cross pattern, it struggles to pinpoint the true global optimum. This difficulty can be




(a) BO on objective function

(b) BO history

Figure 4.12: Updating history of BO when **not** using common random samples for 1D nonlinear case, where the dark blue curve and blue shaded area are the estimated mean and standard deviation of $U_\lambda(d)$ (i.e., objective function), grey triangles are the initial points of BO, orange circle are the searching points of BO, and the red star is the optimal point of BO.

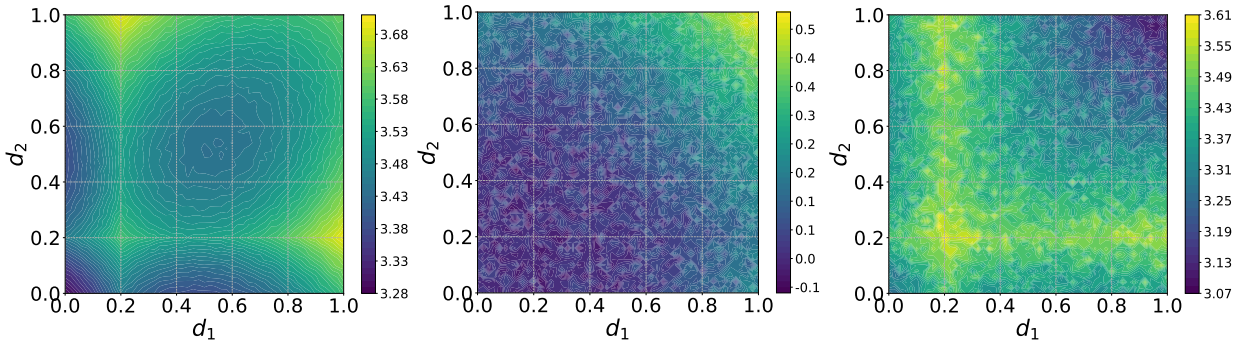



(a) $U(d)$

(b) $\tilde{U}(d)$

(c) $U_\lambda(d)$ when $\lambda = 1$

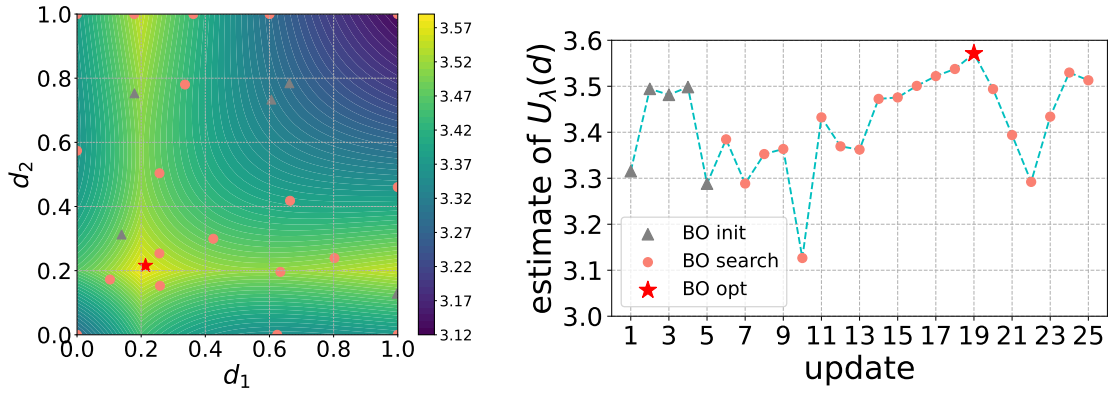
Figure 4.13: Contours of estimated expected utility, utility variance and variance-penalized objective when using common random samples for 2D nonlinear case.




(a) $U(d)$ (b) $\tilde{U}(d)$ (c) $U_\lambda(d)$ when $\lambda = 1$

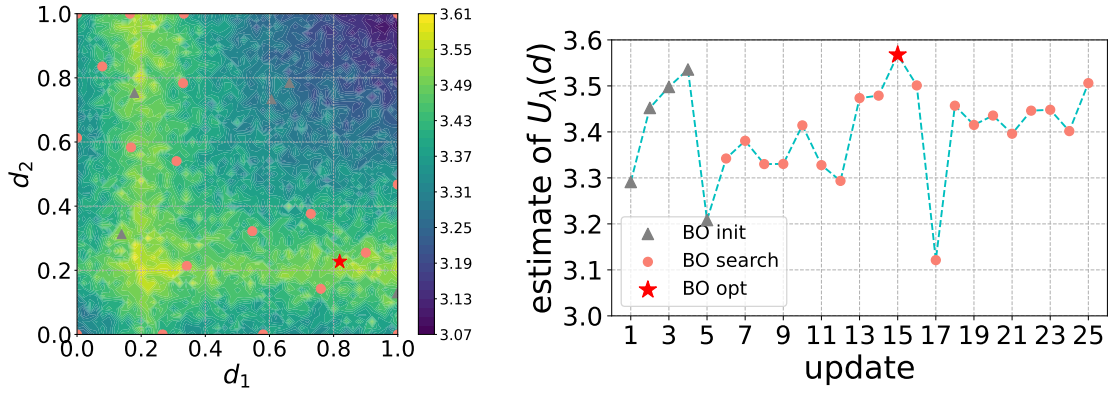
Figure 4.14: Contours of estimated expected utility, utility variance and variance-penalized objective when **not** using common random samples for 2D nonlinear case.

mitigated by increasing MC samples in estimating the objective, or running BO for more iterations.




(a) BO on objective contour (b) BO history

Figure 4.15: Updating history of BO when using common random samples for 2D nonlinear case, where the background is the estimate of $U_\lambda(d)$ (i.e., objective function), grey triangles are the initial points of BO, orange circles are the searching points of BO, and the red star is the optimal point of BO.




(a) BO on objective contour

(b) BO history

Figure 4.16: Updating history of BO when **not** using common random samples for 2D nonlinear case, where the background is the estimate of $U_\lambda(d)$ (i.e., objective function), grey triangles are the initial points of BO, orange circles are the searching points of BO, and red star is the optimal point of BO.

### 4.3.3 Contaminant source inversion in a diffusion field

#### 4.3.3.1 Problem setup

The progression of a contaminant's concentration in a 2D square domain $[0,1]^2$ may be described by the scalar diffusion PDE:

$$\frac{\partial G(z,t;\theta)}{\partial t} = \nabla^2 G + S(z,t;\theta), \qquad z \in [0,1]^2, \quad t > 0, \tag{4.43}$$

where $\theta = [\theta_x, \theta_y] \in \mathcal{R}^2$ represents the source location, which is also the unknown parameter to be inferred and endowed with a uniform prior $\mathcal{U}[0,1]^2$. The source term has the form

$$S(z,t;\theta) = \frac{s}{2\pi h^2} \exp\left(-\frac{\|\theta - z\|^2}{2h^2}\right), \tag{4.44}$$

where $s = 2$ and $h = 0.05$ indicate the source strength and source width respectively. The initial condition is $G(z,0;\theta) = 0$ and we apply Neumann boundary condition on all sides of the square domain. The PDE is solved by the second-order finite volume method on a uniform grid with $\Delta z_x = \Delta z_y = 0.01$, and the time marching is second order fractional step method with $\Delta t = 5\times10^{-4}$.

The design variable is selected as the location of the sensor to measure the contaminant concentration. We only do one experiment at $t = 0.16$, thus the dimension of the design variable only depends on how many sensors we want to place to take measurements (i.e., if we have $m$ sensors in the domain, then the dimension of design variables will be $N_d = 2m$, and the dimension of

observations will be $N_y = m$, while the dimension of parameters $N_\theta$ is always 2). We also assume that there is an additive Gaussian noise on the measurements, such that the measurement at location $z$ is modelled by

$$y(z) = G(z, t = 0.16; \theta) + \epsilon, \tag{4.45}$$

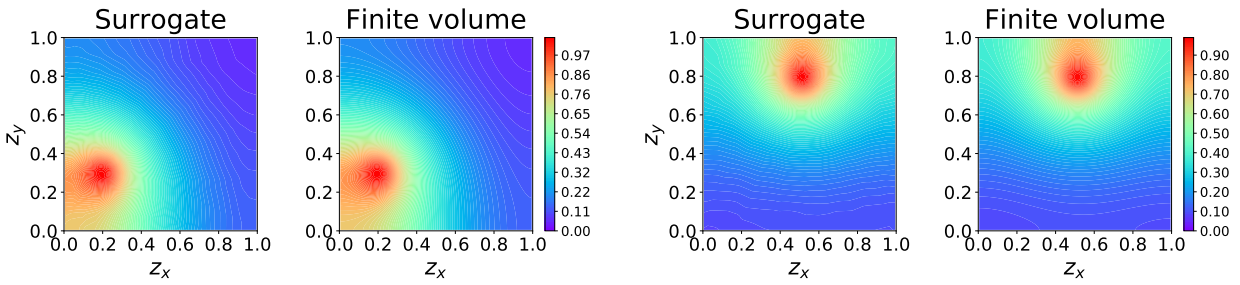
where $\epsilon \sim \mathcal{N}(0, 0.05^2)$ represents the additive noise. The design variable $d$ is then just a set of sensor locations (i.e., $d = [z^{(1)}, \ldots, z^{(m)}]$ for $m$ sensors), and the observation $y$ is simply a batch of observations at those sensor locations (i.e., $y = [y(z^{(1)}), \ldots, y(z^{(m)})]$). For multiple sensor problems, we also assume that the measurement noise to be independent and identically distributed across different sensors.

#### 4.3.3.2 Surrogate modeling

Using the full PDE solver as the forward model is doable but computationally expensive. For example, using a single 2.6 GHz CPU on a MacBook Pro laptop requires approximately 1.2 seconds per PDE forward model solve, and so estimating $U_\lambda(d)$ with 10000 MC samples takes about 3.3 hours at each $d$. To accelerate the computations, we use deep neural networks (DNNs) to construct a surrogate model of $G(z, t = 0.16; \theta)$. The DNN takes a four-dimensional input including $\theta$ and $z$. It has 5 hidden layers, each with 100, 200, 100, 50, and 20 nodes, and ReLU activation function. The output of the DNN is a scalar representing the value of $G$. To train the DNN, we first generate 1000 $\theta$ uniformly sampled over the parameter space, then obtain the corresponding $G$ on uniform grid points in $z$ by running the full PDE solver. These results are then split into 80% training set and 20% testing set. After training, the testing mean squared error (MSE) can reach down to the order of $10^{-6}$. Figure 4.17 shows the comparison of the contaminant concentration $G$ computed by DNN surrogate and finite volume, they appear almost identical. More importantly, DNN surrogate model provides a $10^5$ speedup at the cost of 40 minutes on generating the data and training the DNN.

#### 4.3.3.3 Results

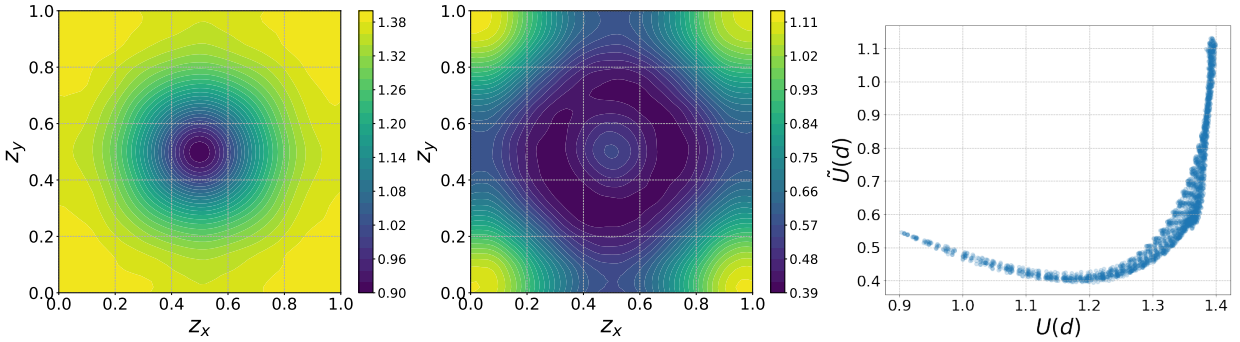
We first consider a case with design one sensor. Objective value is estimated using $N = 30000$ MC samples. Common random samples are employed across different designs. Figure 4.18a shows the contours of estimated expected utility, from which we observe the four corners to have the highest expected utility while the center of the domain has the lowest, which has already been explained in Case 1 in Sec. 2.3.2. However, the four corners also have a much higher utility variance as shown in Fig. 4.18b. The high variance at the corners results from the utility of a corner design can vary significantly depending on the distance between the sensor location and the source location.




(a) example 1 (b) example 2

Figure 4.17: Sample comparison of the concentration field $G$ using the DNN surrogates (left column) and finite volume (right column). They appear nearly identical.

Figure 4.18c further presents the scatter plot of estimated utility variance against the expected utility. From this figure, we observe a steep cliff at the high $U(d)$ region, which means that many designs have similar expected utility but quite different utility variances. Hence, through the rOED formulation, we can identify a design with low utility variance but still achieves high expected utility.




(a) Estimated $U(d)$ (b) Estimated $\tilde{U}(d)$ (c) Estimated $\tilde{U}(d)$ versus $U(d)$

Figure 4.18: Contours of estimated expected utility, utility variance and the scatter plot of utility variance against expected utility when using common random samples for 2D source inversion case with 1 sensor.

Figure 4.19 presents the contours of estimated variance-penalized objective with different $\lambda$ values, and Fig. 4.20 shows the histograms of $u(d_U^*, y)$ and $u(d_{U_\lambda}^*, y)$, where the $d_U^*$ and $d_{U_\lambda}^*$ are selected from the grid points with maximal $U(d)$ and $U_\lambda(d)$ in Fig. 4.19. As $\lambda$ increases, the optimal sensor location first moves from the corner to the middle of the boundary and then moves towards the domain center, and the utility variance shrinks significantly with a small sacrifice on the expected utility, which can be seen by observing the values before and after the $\pm$ sign in Fig. 4.20.

To illustrate the BO results, we focus on a specific case with $\lambda = 0.5$. Figure 4.21 presents the

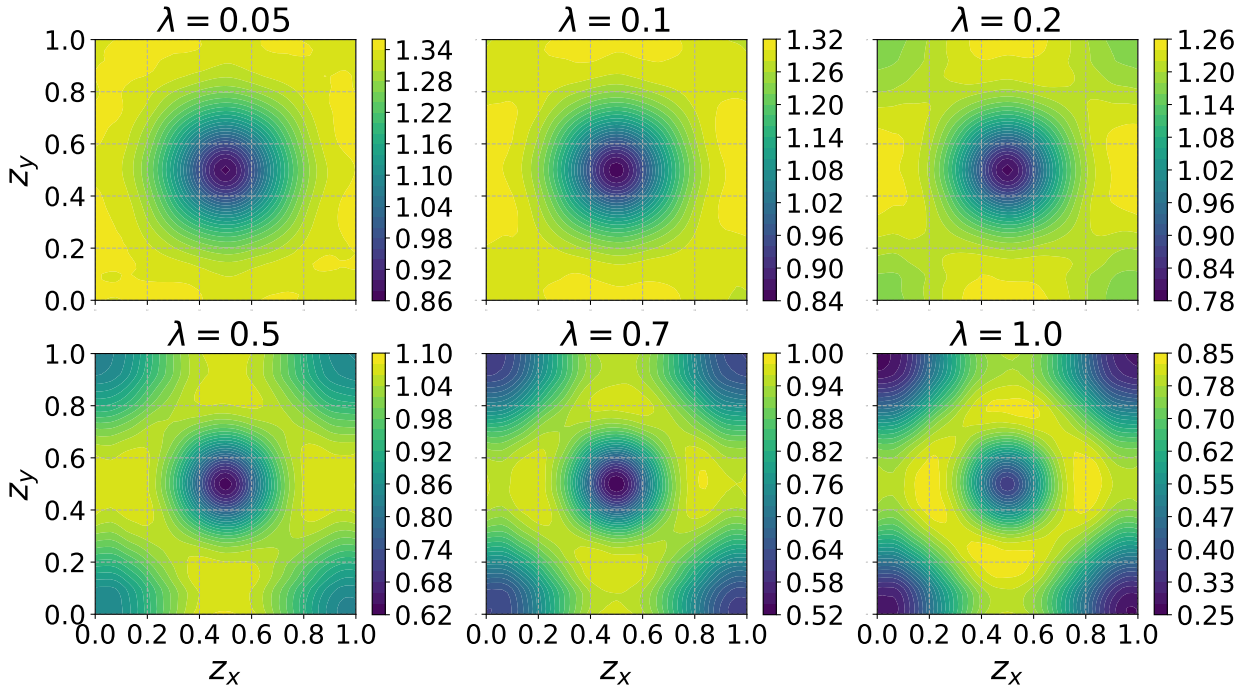



Figure 4.19: Contours of estimated variance-penalized objective with different $\lambda$ values for 2D source inversion case with 1 sensor.

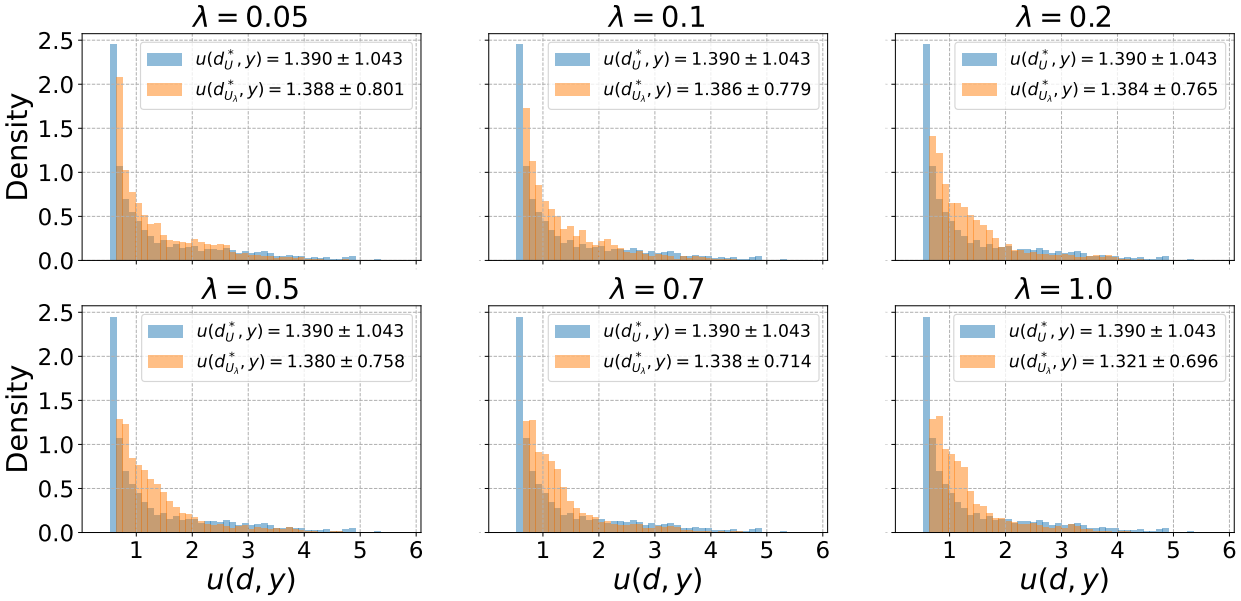
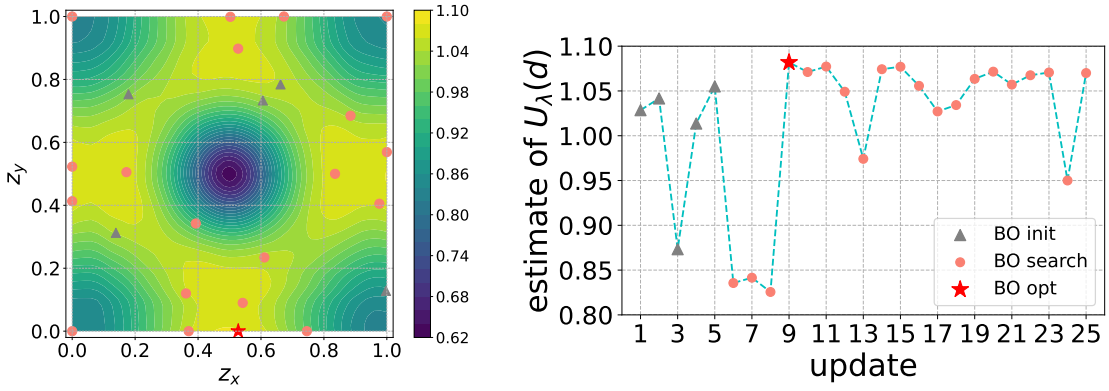



Figure 4.20: Histograms of $u(d_U^*, y)$ and $u(d_{U_\lambda}^*, y)$ with different $\lambda$ values for 2D source inversion case with 1 sensor, where the value before ± sign is the MC mean (expected utility), and the value after ± sign is the MC standard deviation (square root of the utility variance).

updating history of BO, showing rapid convergence to the optimal design located near the middle of each of the domain's boundaries. In fact, BO has searched all four regions (from symmetry) with high objective function value, showing good performance in finding the global optimum. To further explain why the middle of the domain boundary has lower variance than the corners, we draw 3000 $\theta$ samples from the prior, generate corresponding observations $y$ through the surrogate model for both $d_U^*$ and $d_{U_\lambda}^*$ that are found by BO, and then compute the KL divergence from prior to posterior for these $y$ observations. After that, we pick 5 $\theta$ samples with low KL divergence and draw their posteriors in Fig. 4.22. The worst cases of $d_U^*$ have a lower KL divergence than the worst cases of $d_{U_\lambda}^*$, with a more flat posterior distribution. This is because when the source is located somewhere near the diagonal, the posterior of $d_U^*$ will also have a full diagonal shape across the square domain, hence a wider distribution coverage and a lower KL divergence.




(a) BO on objective contour

(b) BO history

Figure 4.21: Updating history of Bayesian optimization when using common random samples for 2D source inversion case with 1 sensor, where the background is the estimate of $U_\lambda(d)$ when $\lambda = 0.5$ (i.e., objective function), grey triangles are the initial points of BO, orange circles are the searching points of BO, and the red star is the optimal point of BO.

Next we design two sensor locations for a $N_d = 4$ setting. For illustration, we randomly sample 1000 combinations of two sensor locations and use $N = 30000$ MC samples for each estimate. Figure 4.23a shows the estimated expected utility of those location combinations, with the value represented by the coloring and the highest combination is marked by a thick line. The results suggest taking measurements at the two adjacent corners yields the highest expected utility. Figure 4.23b similarly plots the estimated utility variance of location combinations, and the combination with the lowest estimated utility variance is marked by a thick line. Combinations that are closer to the domain center have smaller variances. Figure 4.23c further shows the scatter plot between estimated utility variance and expected utility, from which we can find a similar pattern as the 1 sensor case with a steep cliff at the high $U(d)$ region.

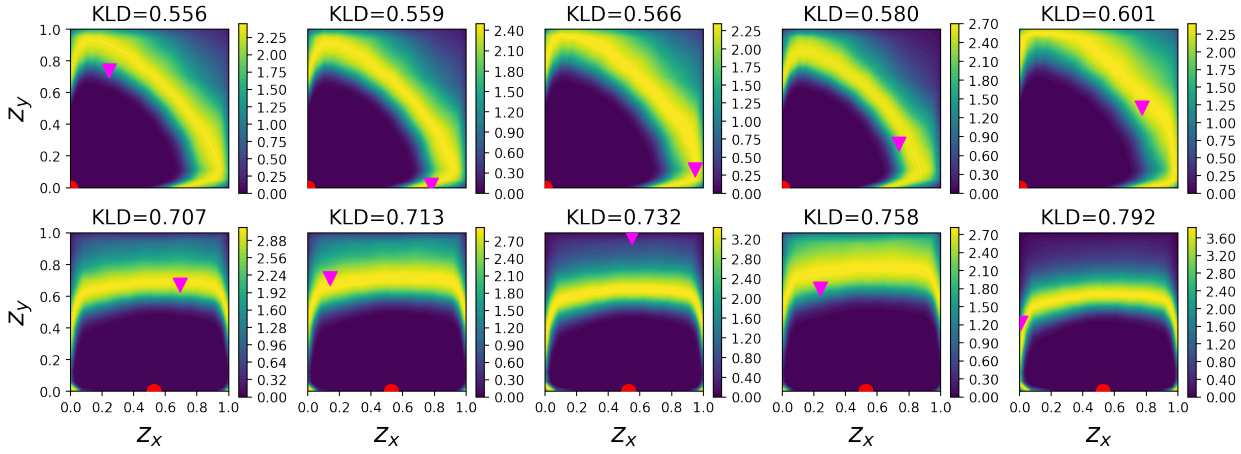
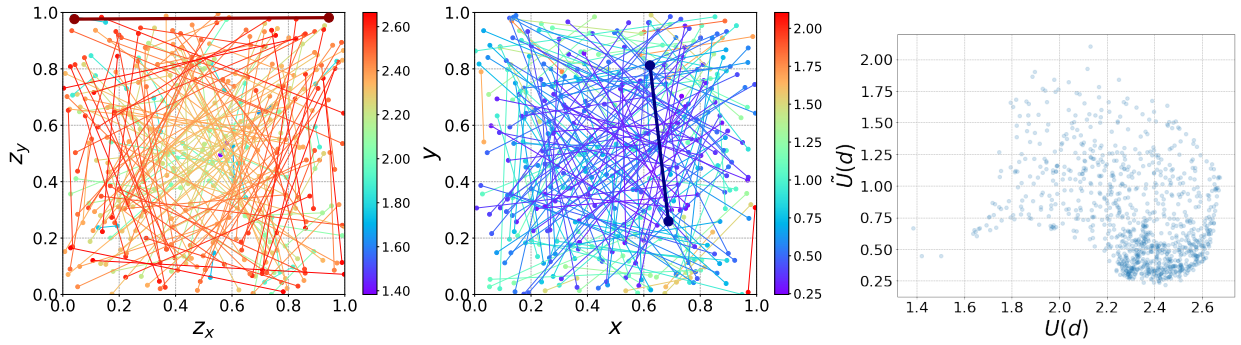



Figure 4.22: Example posteriors with low KL-divergence for 2D source inversion case with 1 sensor, where the first row corresponds to $d^*_U$ and the second row $d^*_{U_\lambda}$, the red star denotes the sensor location, and the magenta inverted triangle denotes the true source location.




(a) Estimated $U(d)$ (b) Estimated $\tilde{U}(d)$ (c) Estimated $\tilde{U}(d)$ v.s. $U(d)$

Figure 4.23: Random combinations of sensor locations and their estimated expected utility, utility variance, and the scatter plot of utility variance against expected utility when using common random samples for 2D source inversion case with 2 sensors.

Figure 4.24 presents the random location combinations and their estimated $U_\lambda(d)$ with different $\lambda$ values, and Fig. 4.25 shows the histograms of $u(d_U^*, y)$ and $u(d_{U_\lambda}^*, y)$, where the $d_U^*$ and $d_{U_\lambda}^*$ are selected from the random combinations with maximal $U(d)$ and $U_\lambda(d)$ in Fig. 4.24. As $\lambda$ increases, the optimal sensor location first moves from the domain boundary to the domain center, and the utility variance shrinks significantly with a small sacrifice on the expected utility, which is similar to the 1 sensor case. The histogram of $u(d_U^*, y)$ has a multimodal distribution, whose highest peak lies in the low utility region; while the histogram of $u(d_{U_\lambda}^*, y)$ is unimodal, whose peak lies in the middle of the two peaks of $u(d_U^*, y)$. The rOED framework effectively merges these two peaks together and find a design that is more robust, or "more unimodal" in terms of its histogram.

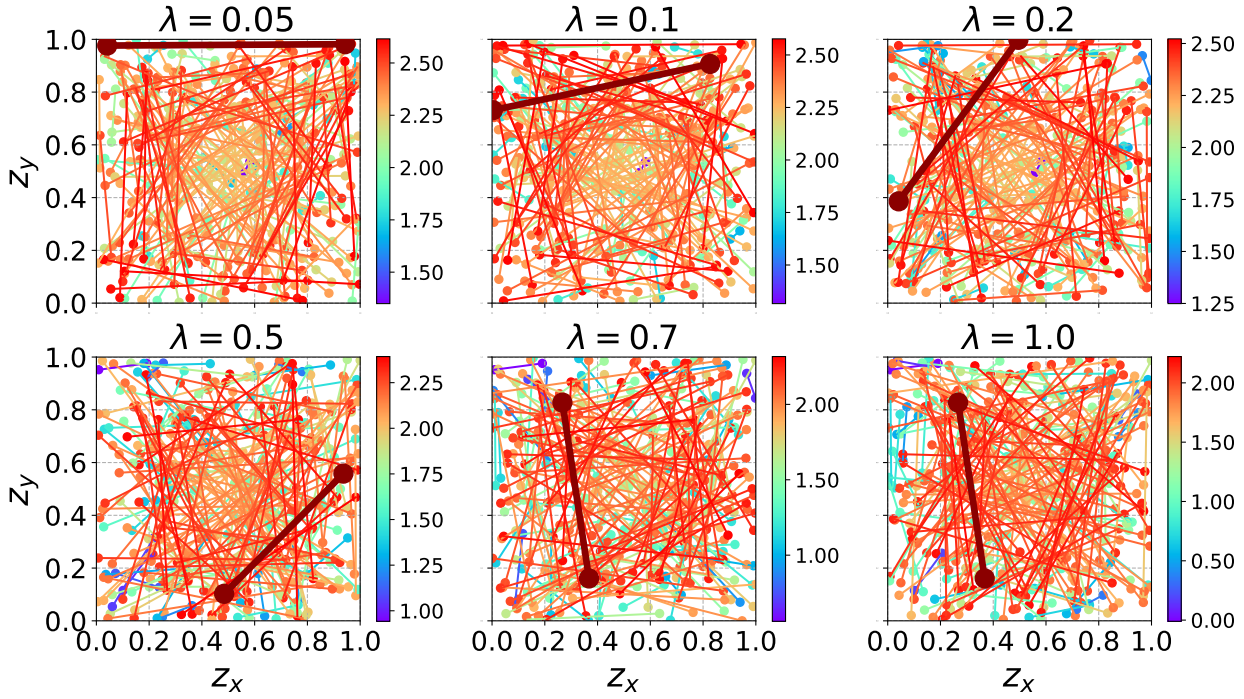



Figure 4.24: Contours of estimated variance-penalized objective with different $\lambda$ values for 2D source inversion case with 2 sensors.

The BO results with $\lambda = 0$ and $\lambda = 0.5$ are shown in Fig. 4.26. For $\lambda = 0$, BO optimum after 50 updates has an objective function value of 2.669, which is very close to the maximum of 1000 random combinations 2.674. For $\lambda = 0.5$, BO optimum has a value of 2.382, which is also close to the maximum of 1000 random combinations 2.398. BO, however, found these high values in significantly fewer objective evaluations (less than 20).

To further illustrate the difference between $d_U^*$ and $d_{U_\lambda}^*$, we draw 3000 $\theta$ samples from the prior, generate corresponding observations $y$ through the surrogate model for both $d_U^*$ and $d_{U_\lambda}^*$ that are found by BO, and then compute the KL divergence from prior to posterior values for these $y$ observations. After that, we pick 5 $\theta$ samples with low KL divergence (including the one with the

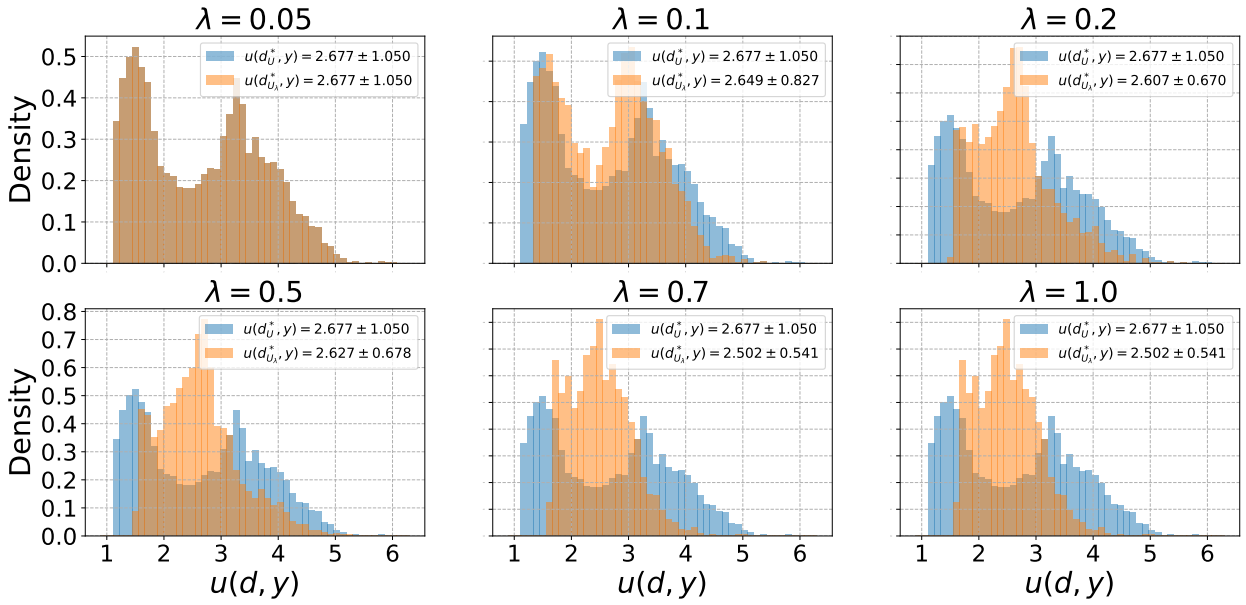
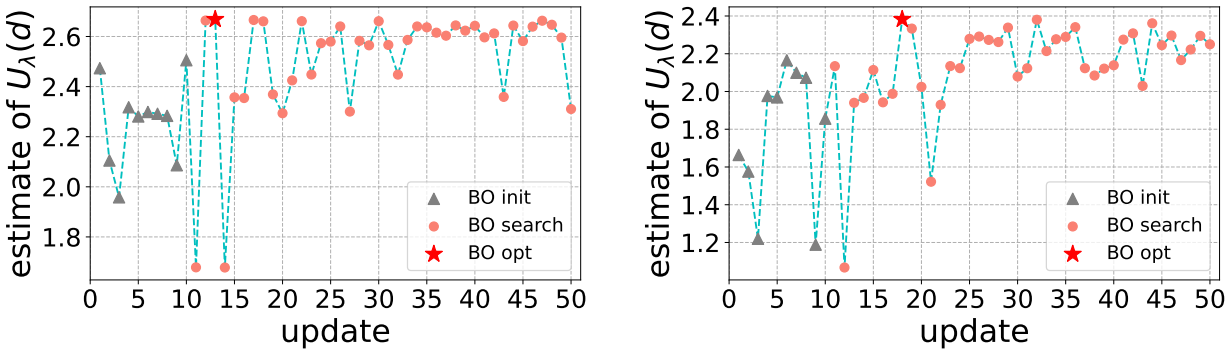



Figure 4.25: Histograms of $u(d_U^*, y)$ and $u(d_{U_\lambda}^*, y)$ with different $\lambda$ values for 2D source inversion case with 2 sensors, where the value before ± sign is the MC mean (expected utility), and the value after ± sign is the MC standard deviation (square root of the utility variance).




(a) BO history when $\lambda = 0$

(b) BO history when $\lambda = 0.5$

Figure 4.26: The updating history of Bayesian optimization when using common random samples for 2D source inversion case with 2 sensors, where grey triangles are the initial points of BO, orange circles are the searching points of BO, and the red star is the optimal point of BO.

lowest KL divergence) as the worst cases and draw the posteriors of them in Fig. 4.27. Similar to the 1 sensor case, the worst cases of $d_U^*$ display a much lower utility than the worst cases of $d_{U_\lambda}^*$ with a more flat posterior distribution.

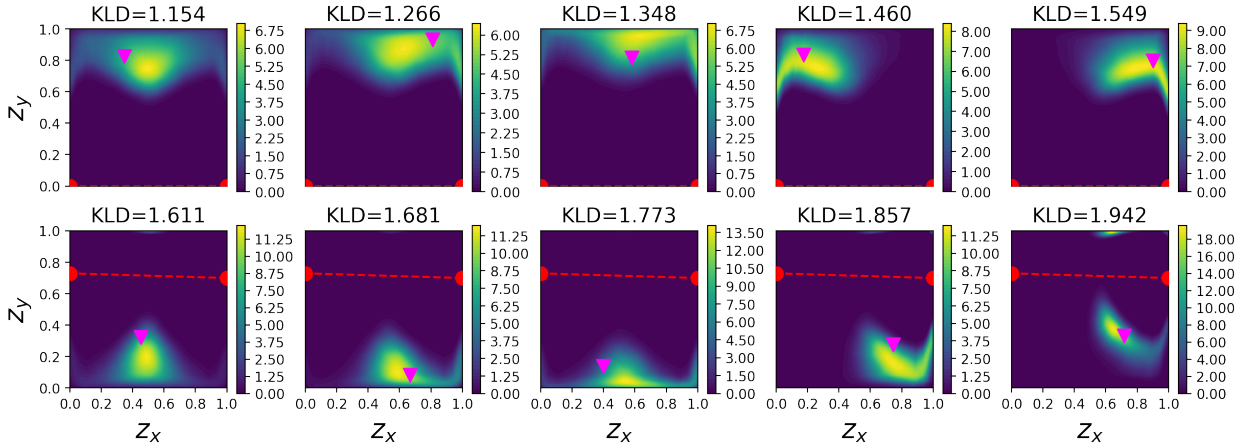



Figure 4.27: Example posteriors with low KL-divergence for 2D source inversion case with 2 sensors, where the first row corresponds to $d_U^*$ and the second row $d_{U_\lambda}^*$, the red stars denote the sensor locations, and the magenta inverted triangle denotes the true source location.

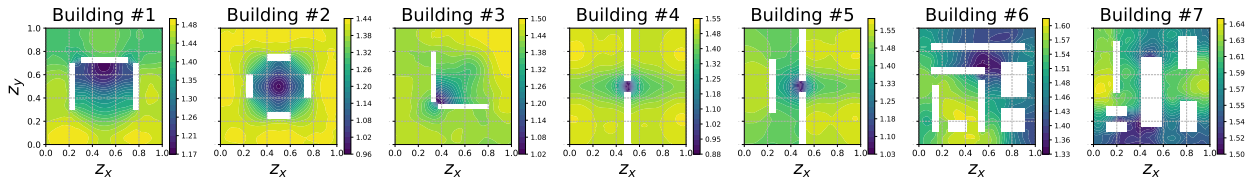
### 4.3.4 Contaminant source inversion with building obstacles

We now add additional building obstacles to the source inversion domain in order to make it more realistic. The prior of source location is still a uniform distribution except for the area of building obstacles, in which the prior density is 0.

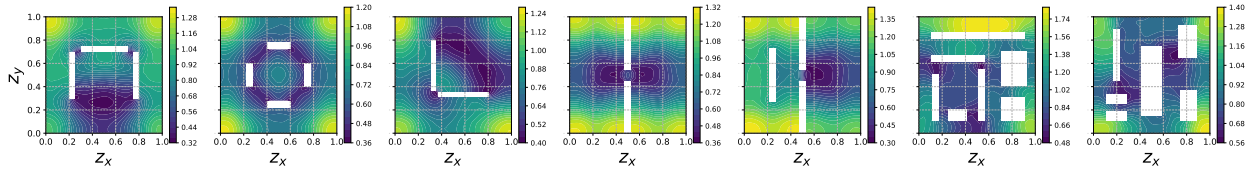
Figure 4.28 presents the contours of estimated expected utility, utility variance and the scatter plot of utility variance against the expected utility with 7 different building obstacles, where each column corresponds to the same building. The 'steep cliff' also exists, which means that we can still find a robust design that has much lower utility variance but slightly lower expected utility when there are building obstacles in the domain.

We further pick two representative cases, which are building #4 and #5 and designing 1 and 2 sensors respectively. We first focus on placing 1 sensor on building #4, with Fig. 4.29 presenting the contours of estimated variance-penalized objective with different $\lambda$ values and Fig. 4.30 showing the histograms of $u(d_U^*, y)$ and $u(d_{U_\lambda}^*, y)$. Here the $d_U^*$ and $d_{U_\lambda}^*$ are found from grid search. As $\lambda$ increases, the optimal sensor location moves towards the domain center.

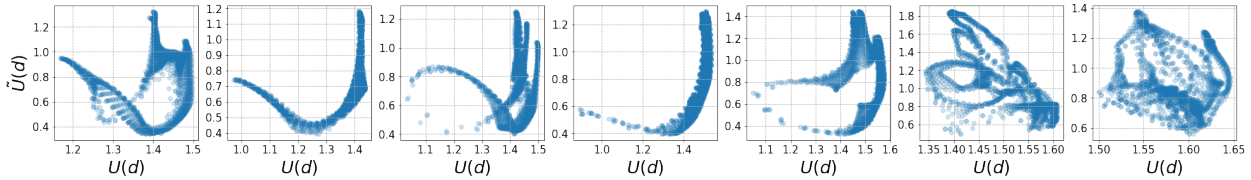
We then apply BO to find the optimal design $d_{U_\lambda}^*$ for both $\lambda = 0$ and $\lambda = 0.5$. Figure 4.31 shows the updating history of BO, where it finds 3 out of 4 local optimums within tens of updates. Note that the design constraints are active to prevent placing sensors inside the building obstacles during optimization of the acquisition function.




(a) Estimated $U(d)$




(b) Estimated $\tilde{U}(d)$




(c) Estimated $\tilde{U}(d)$ v.s. $U(d)$

Figure 4.28: Contours of estimated expected utility, utility variance and the scatter plot of utility variance against expected utility with 7 different building obstacles for 2D source inversion case with 1 sensor, where each column corresponds to the same building.

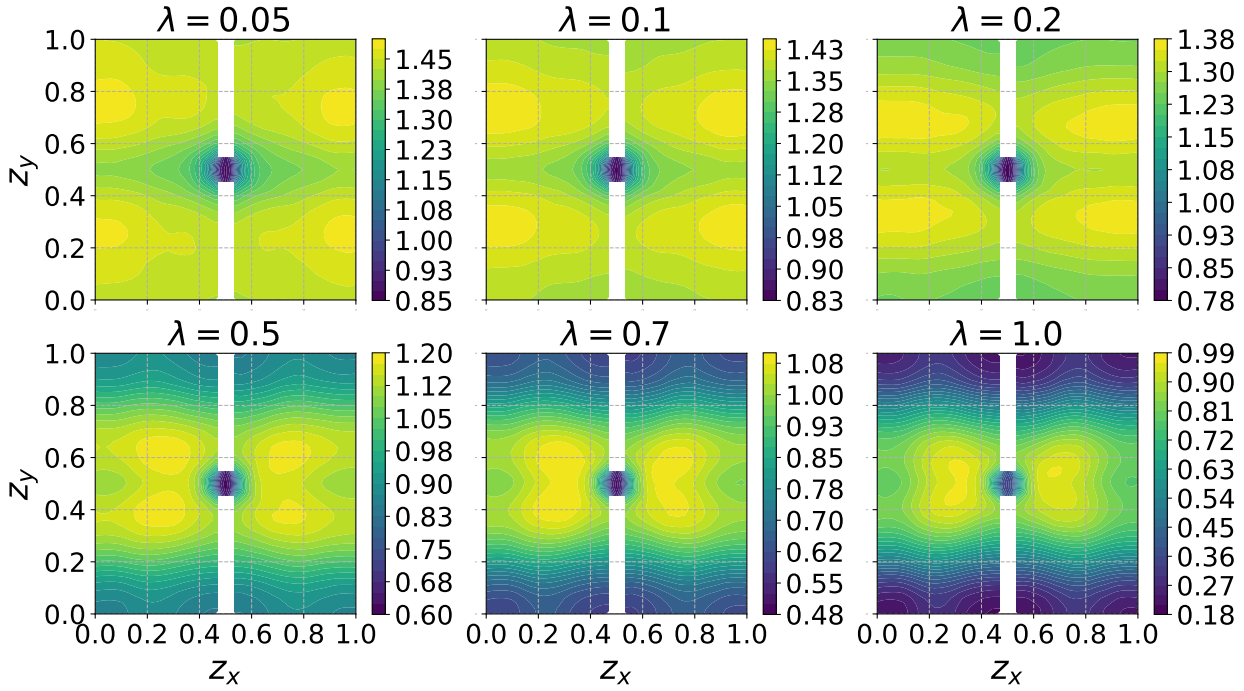



Figure 4.29: Contours of estimated variance-penalized objective with different $\lambda$ values for 2D source inversion case with 1 sensor and building #4.

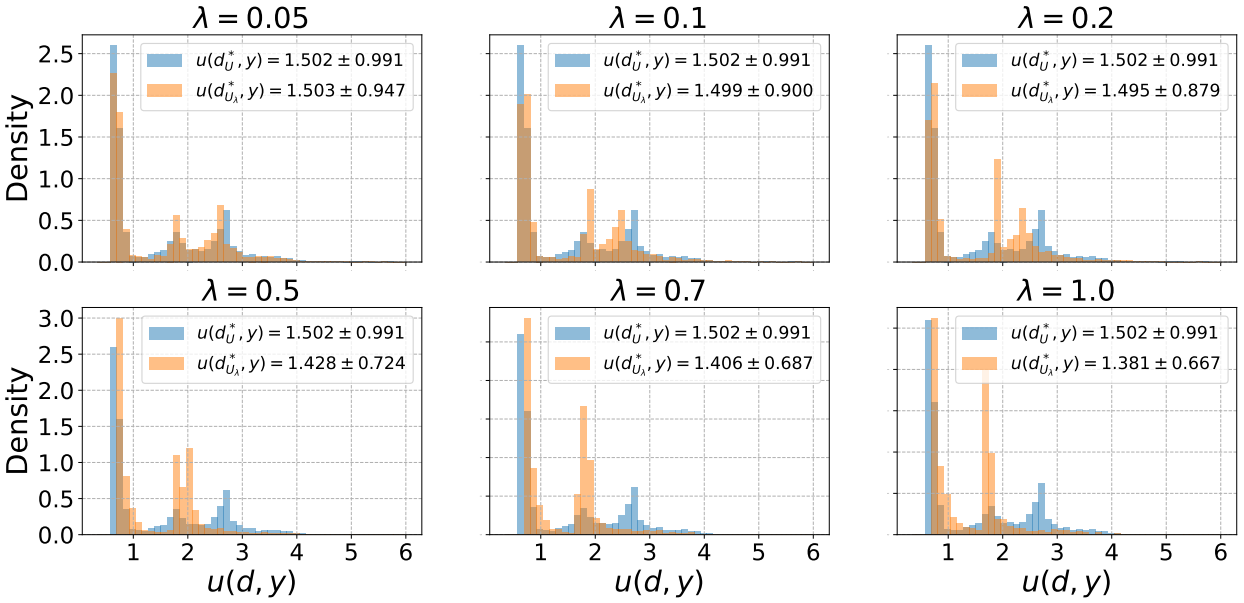
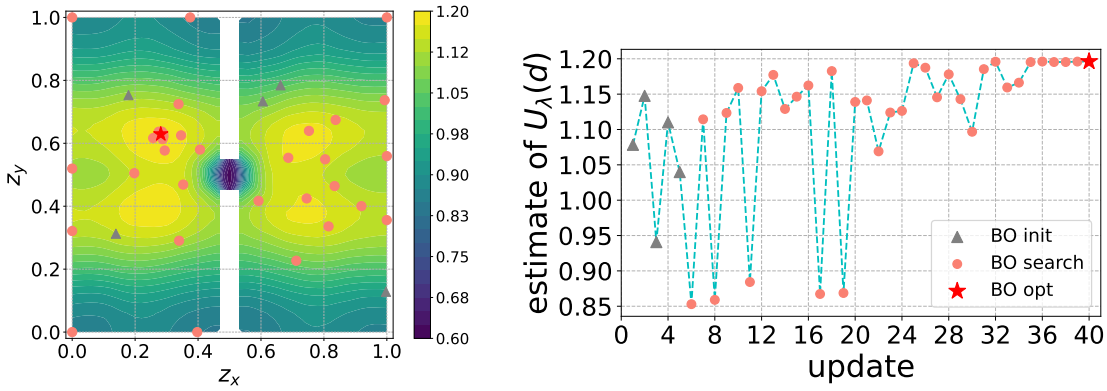



Figure 4.30: Histograms of $u(d_U^*, y)$ and $u(d_{U_\lambda}^*, y)$ with different $\lambda$ values for 2D source inversion case with 1 sensor and building #4, where the value before ± sign is the MC mean (expected utility), and the value after ± sign is the MC standard deviation (square root of the utility variance).




(a) BO on objective contour

(b) BO history

Figure 4.31: Updating history of BO when using common random samples for 2D source inversion case with 1 sensor and building #4, where the background is the estimate of $U_\lambda(d)$ when $\lambda = 0.5$ (i.e., objective function), grey triangles are the initial points of BO, orange circles are the searching points of BO, and the red star is the optimal point of BO.

We then draw posteriors of 5 worst cases of $d_U^*$ and $d_{U_\lambda}^*$ that are obtained from BO in Fig. 4.32. For the worst cases of $d_U^*$, it is difficult to tell apart whether the source is on the left side or the right side (see the probability mass at the bottom of the left side), and sometimes will even put vast probability mass to the wrong side (see the second and the fourth cases); while for the worst cases of $d_{U_\lambda}^*$, at least it can discriminate the side of the source correctly, thus results in a higher utility for the worst cases.

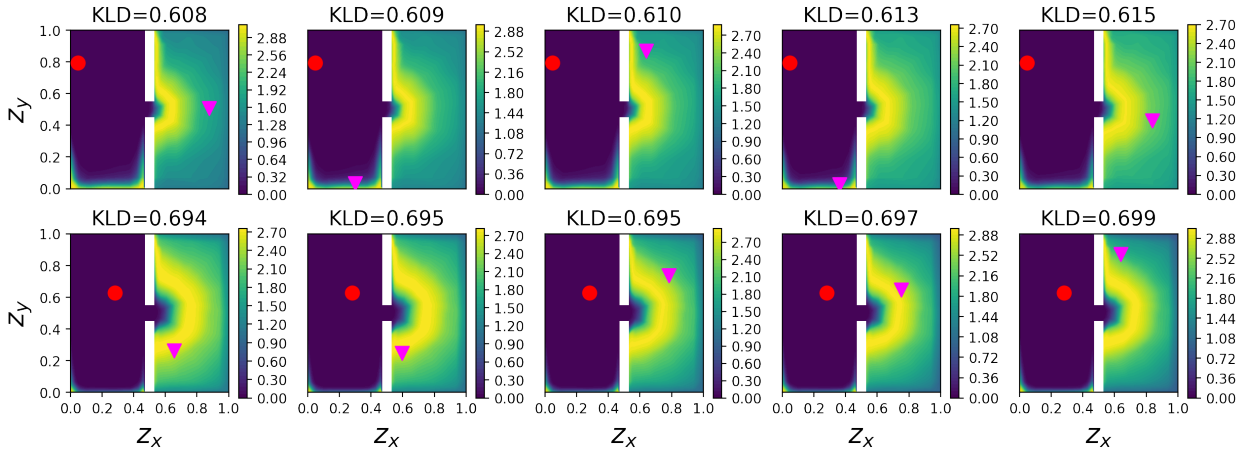



Figure 4.32: Example posteriors with low KL-divergence for 2D source inversion case with 1 sensor and building #4, where the first row corresponds to $d_U^*$ and the second row $d_{U_\lambda}^*$, the red star denotes the sensor location, and the magenta inverted triangle denotes the true source location.

Now focusing on the second example for designing two sensors on building #5, Fig. 4.33 shows

the histograms of $u(d_U^*, y)$ and $u(d_{U_\lambda}^*, y)$ and Fig. 4.34 plots the example posteriors of 5 worst cases of $d_U^*$ and $d_{U_\lambda}^*$. Overall it makes sense to have two sensors on both the left side and right side, and the robust design $d_{U_\lambda}^*$ further places two sensors one at the bottom and the other at the top. Such a design can be understood to be more space-filling and likely to mitigate some of the worst cases encountered by $d_U^*$ where the source location happens to be far away from both two sensors at the top.

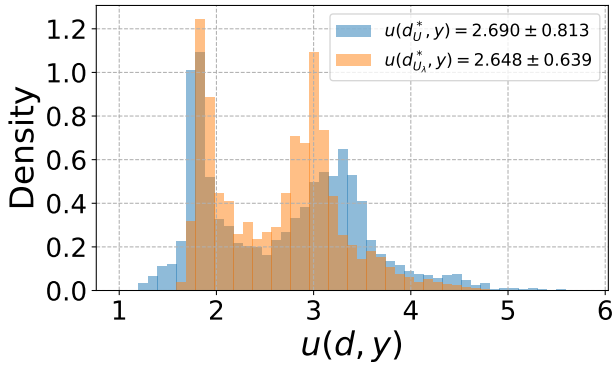



Figure 4.33: Histograms of $u(d_U^*, y)$ and $u(d_{U_\lambda}^*, y)$ for 2D source inversion case with 1 sensor and building #5, where the value before ± sign is the MC mean (expected utility), and the value after ± sign is the MC standard deviation (square root of the utility variance).

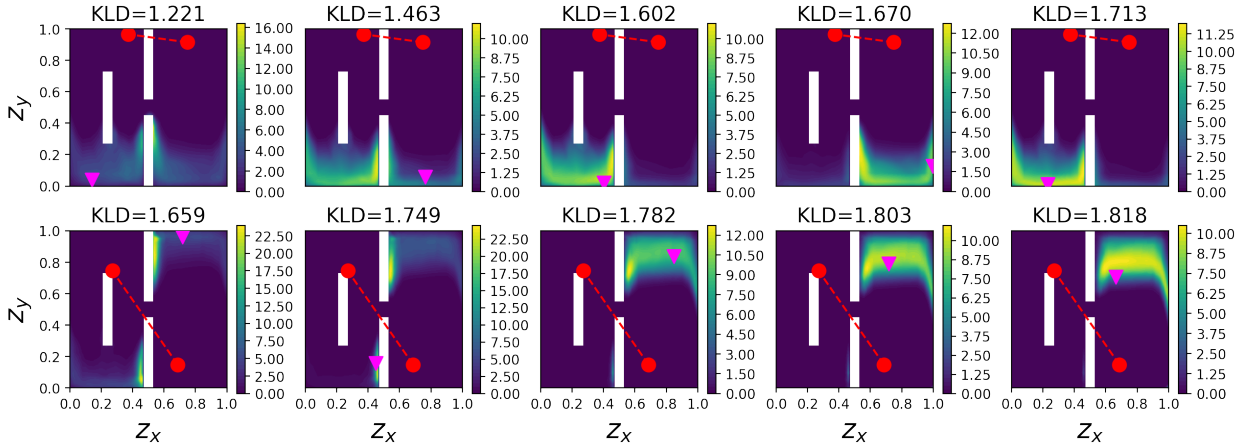



Figure 4.34: Example posteriors with low KL-divergence for 2D source inversion case with 1 sensor and building #5, where the first row corresponds to $d_U^*$ and the second row $d_{U_\lambda}^*$, the red star denotes the sensor location, and the magenta inverted triangle denotes the true source location.

## 4.4 Summary

In this chapter, we introduce a variance-penalized robust criterion for achieving robust Bayesian optimal experimental design (**rOED**). This criterion is applicable to a wide range of utility functions that adhere to the general form suggested by [93], and the variance-penalized criterion itself also conforms to this general form. Adopting a Bayesian perspective with a focus on parameter inference, we employ the information gain proposed by [92] as the utility function. The robust criterion favors a design with higher expected information gain, but also lower variance. To efficiently estimate the variance-penalized objective, we propose a Monte Carlo sampling technique that incorporates sample reuse. Our numerical examples demonstrate the convergence rate of estimation accuracy as the sample number increases and highlight the value of considering utility variance. To obtain the globally optimal design in an efficient manner, we propose employing Bayesian optimization (BO). Moreover, common random samples are used to introduce artificial correlation among different design points, to smoothen the objective function and expedite optimization convergence.

The key contributions and novelty of our rOED method are summarized as follows.

- We formulate the variance-penalized rOED framework.
- We present the Monte Carlo estimator for estimating the objective of rOED, and analyze the variance and bias of this estimator.
- We validate rOED on a benchmark example and showcase its effectiveness in tackling complex physical problems.
- We make available our roed code at `https://github.com/wgshen/rOED`.

# CHAPTER 5

# Robust Sequential Optimal Experimental Design

Having introduce sequential optimal experimental design (sOED) in Chapter 2 and robust optimal experimental design (rOED) for batch design in Chapter 4, we now combine these principles together to introduce the variance-penalized **robust sequential optimal experimental design (rsOED)**. This short chapter presents the formulation of rsOED, numerical methods for solving the rsOED problem centering around the idea of estimating policy gradient (PG) for the the variance-penalized expected total utility using Monte Carlo (MC) sampling and variational approximation. Lastly, a numerical example of nonlinear model is presented to demonstrate the effectiveness of rsOED.

## 5.1 Problem formulation

The rsOED framework is identical to that of sOED from Chapter 2 (see Sec. 2.1.2), except that the objective function in Eqn. (2.3) is replaced with the variance-penalized objective. The rsOED problem statement is then, from a given initial state $x_0$, find the optimal policy

$$\begin{aligned} \pi^* = \arg\max_{\pi} \quad & U_\lambda(\pi) \\ \text{s.t.} \quad & d_k = \mu_k(x_k) \in \mathcal{D}_k, \\ & x_{k+1} = \mathcal{F}_k(x_k, d_k, y_k), \qquad \text{for} \quad k = 0, \ldots, N-1, \end{aligned} \tag{5.1}$$

where the objective function is now

$$U_\lambda(\pi) = U(\pi) - \lambda\tilde{U}(\pi) \tag{5.2}$$

$$= \mathbb{E}_{y_0,\ldots,y_{N-1}|\pi,x_0}\left[\sum_{k=0}^{N-1} g_k(x_k, d_k, y_k) + g_N(x_N)\right] - \lambda\mathbb{V}_{y_0,\ldots,y_{N-1}|\pi,x_0}\left[\sum_{k=0}^{N-1} g_k(x_k, d_k, y_k) + g_N(x_N)\right] \tag{5.3}$$

and $\lambda$ is a scalar penalty coefficient. The purpose of the new objective function Eqn. (5.3) is to enable a tradeoff between maximizing the expected utility and minimizing the utility variance, thereby achieving more robust policy that is less affected by variability in the experimental outcomes.

It is worth noting that, unlike the incremental-terminal equivalence of the expected total utility $U(\pi)$ shown in Theorem 1, the variance of the total utility $\tilde{U}(\pi)$differs between the incremental and terminal formulations (i.e., $\tilde{U}_T(\pi) \neq \tilde{U}_I(\pi)$). We only focus on the terminal formulation (Eqn. (2.7) and (2.8)) for rsOED in this chapter.

Lastly, our rsOED formulation in this chapter will only present the setup involving a single forward model with OED for parameter inference. However, it can be extended to accommodate the generalized scenarios involving multi-model, goal-oriented Quantities of Interest (QoIs), and nuisance parameters, akin to the vsOED formulation in Chapter 3.

## 5.2 Numerical methods for rsOED

Similar to the numerical methods for sOED (see Sec. 2.2), we approach the rsOED problem by explicitly parameterizing the policy function. This allows us to leverage gradient-based optimization techniques to optimize the policy parameters and find the policy that maximizes the variance-penalized expected total utility. In the following, we first present the policy gradient of rsOED in Sec. 5.2.1 with its numerical estimation in Sec. 5.2.2, and then discuss how to estimate the KL divergence using the variational approximation in Sec. 5.2.3.

### 5.2.1 Derivation of the policy gradient

The strategy for establishing the policy gradient based rsOED (PG-rsOED) is similar to that of the PG-sOED. Each policy function $\mu_k$ is parameterized by parameters $w_k$ ($k = 0, \ldots, N-1$), denoted as $\mu_{k,w_k}$. The overall policy $\pi$ is then parameterized by the set $w = \{w_k, \forall k\} \in \mathbb{R}^{N_w}$ and denoted as $\pi_w$, where $N_w$ is the dimension of the overall policy parameter vector. The rsOED problem

statement, with a parameterized policy, becomes:

$$
\begin{aligned}
w^* = \arg\max_{w} \quad & U_\lambda(w) \\
\text{s.t.} \quad & d_k = \mu_{k,w_k}(x_k) \in \mathcal{D}_k, \\
& x_{k+1} = \mathcal{F}_k(x_k, d_k, y_k), \qquad \text{for} \quad k = 0, \ldots, N-1,
\end{aligned} \tag{5.4}
$$

from a given initial state $x_0$, where

$$
\begin{aligned}
U_\lambda(w) &= U(w) - \lambda \tilde{U}(w) && (5.5) \\
&= \mathbb{E}_{y_0,\ldots,y_{N-1}|\pi_w,x_0}\left[\sum_{k=0}^{N-1} g_k(x_k, d_k, y_k) + g_N(x_N)\right] \\
&\quad - \lambda \mathbb{V}_{y_0,\ldots,y_{N-1}|\pi_w,x_0}\left[\sum_{k=0}^{N-1} g_k(x_k, d_k, y_k) + g_N(x_N)\right]. && (5.6)
\end{aligned}
$$

The next step involves deriving the gradient $\nabla_w U_\lambda(w)$ so it can be utilized with gradient-based optimization for solving the rsOED problem.

In order to present the gradient expression for the PG-rsOED method, it is necessary to first introduce the value functions. The *state-value function* (or *V-function*) and the *action-value function* (or *Q-function*) corresponding to the expected utility have been introduced in Appendix A.2 and Sec. 2.2.1, respectively. Here we introduce the *variance state-value function* (*$\tilde{V}$-function*) and the *variance action-value function* (*$\tilde{Q}$-function*) corresponding to the variance of the total utility. The *variance state-value function* following policy $\pi_w$ and at the $k$th experiment is defined as

$$
\tilde{V}_k^{\pi_w}(x_k) = \mathbb{V}_{y_k,\ldots,y_{N-1}|\pi_w,x_k}\left[\sum_{t=k}^{N-1} g_t(x_t, \mu_{t,w_t}(x_t), y_t) + g_N(x_N)\right] \tag{5.7}
$$

$$
\tilde{V}_N^{\pi_w}(x_N) = 0, \tag{5.8}
$$

for $k = 0, \ldots, N-1$, where $x_{k+1} = \mathcal{F}_k(x_k, \mu_{k,w_k}(x_k), y_k)$. The variance state-value function is the variance of cumulative remaining reward starting from a given state $x_k$ and following policy $\pi_w$ for all remaining experiments. The *varaince action-value function* following policy $\pi_w$ and at the $k$th experiment is defined as

$$
\tilde{Q}_k^{\pi_w}(x_k, d_k) = \mathbb{V}_{y_k,\ldots,y_{N-1}|\pi_w,x_k,d_k}\left[g_k(x_k, d_k, y_k) + \sum_{t=k+1}^{N-1} g_t(x_t, \mu_{t,w_t}(x_t), y_t) + g_N(x_N)\right] \tag{5.9}
$$

$$
\tilde{Q}_N^{\pi_w}(x_N, \cdot) = 0, \tag{5.10}
$$

for $k = 0, \ldots, N-1$, where $x_{k+1} = \mathcal{F}_k(x_k, d_k, y_k)$. The variance action-value function is the variance of cumulative remaining reward for performing the $k$th experiment at the given design $d_k$ from a given state $x_k$ and thereafter following policy $\pi_w$. The two functions are related through

$$\tilde{V}_k^{\pi_w}(x_k) = \tilde{Q}_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k)). \tag{5.11}$$

Moreover, the variance action-value function can also be expressed using a recursive relationship as follows:

$$\tilde{Q}_k^{\pi_w}(x_k, d_k) = \mathbb{E}_{y_k|x_k,d_k}\left[\hat{V}_{k+1}^{\pi_w}(x_{k+1}) + \tilde{V}_{k+1}^{\pi_w}(x_{k+1})\right], \tag{5.12}$$

where $\hat{V}_{k+1}^{\pi_w}(x_{k+1}) = \left[g_k(x_k, d_k, y_k) + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, d_k)\right]^2$ for $k = 0, \ldots, N-2$, $\hat{V}_N^{\pi_w}(x_N) = \left[g_{N-1}(x_{N-1}, d_{N-1}, y_{N-1}) + V_N^{\pi_w}(x_N) - Q_{N-1}^{\pi_w}(x_{N-1}, d_{N-1})\right]^2$, and $\tilde{V}_N^{\pi_w}(x_N) = 0$. The proof is provided in Appendix D.1.

The gradient of the variance-penalized objective function in Eqn. (5.5) can be decomposed into the gradient of the expected utility, which is ready in Eqn. (2.16), and the gradient of the variance of the total utility.

**Theorem 6** (Policy gradient for rsOED)**.** *The gradient of the variance of the total utility $\tilde{U}(w)$ with respect to the policy parameters is*

$$\begin{aligned}
&\nabla_w \tilde{U}(w) \qquad (5.13)\\
&= \sum_{k=0}^{N-1} \mathbb{E}_{x_k|\pi_w,x_0}\left[\nabla_w \mu_{k,w_k}(x_k) \nabla_{d_k} \tilde{Q}_k^{\pi_w}(x_k, d_k)\Big|_{d_k=\mu_{k,w_k}(x_k)}\right]\\
&\quad + \sum_{k=0}^{N-2} \mathbb{E}_{x_{k+1}|\pi_w,x_0}\Bigg\{2\left[g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k))\right] \times\\
&\qquad\qquad \sum_{l=k+1}^{N-1} \mathbb{E}_{x_l|\pi_w,x_{k+1}}\left[\nabla_w \mu_{l,w_l}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)|_{d_l=\mu_{l,w_l}(x_l)}\right]\Bigg\}.
\end{aligned}$$

A proof is provided in Appendix D.2. By combining the gradient of the expected utility in Eqn. (2.16) and the gradient of the variance of the total utility in Eqn. (5.13), we can obtain the gradient of the variance-penalized objective as

$$\nabla_w U_\lambda(w) = \nabla_w U(w) - \lambda \nabla_w \tilde{U}(w). \tag{5.14}$$

### 5.2.2 Numerical estimation of the policy gradient

In general, the policy gradient in Eqn. (5.14) does not have a closed-form and numerical approximation is required. A MC estimator for $\nabla_w U(w)$ has been provided in Eqn. (2.17), and the MC estimator for the gradient of variance of the total utility is

$$
\begin{aligned}
&\nabla_w \tilde{U}(w) \\
&\approx \frac{1}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1} \nabla_w \mu_{k,w_k}(x_k^{(i)}) \nabla_{d_k^{(i)}} \tilde{Q}_k^{\pi_w}(x_k^{(i)}, d_k^{(i)})\Big|_{d_k^{(i)}=\mu_{k,w_k}(x_k^{(i)})} \\
&\quad + \frac{1}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-2} 2\left[g_k^{(i)} + V_{k+1}^{\pi_w}(x_{k+1}^{(i)}) - Q_k^{\pi_w}(x_k^{(i)}, \mu_{k,w_k}(x_k^{(i)}))\right] \times \\
&\qquad\qquad \left[\sum_{l=k+1}^{N-1} \nabla_\theta \mu_{l,w_l}(x_l^{(i)}) \nabla_{d_l^{(i)}} Q_l^{\pi_w}(x_l^{(i)}, d_l^{(i)})\Big|_{d_l^{(i)}=\mu_{l,w_l}(x_l^{(i)})}\right] \\
&= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1} \nabla_w \mu_{k,w_k}(x_k^{(i)}) \nabla_{d_k^{(i)}} \tilde{Q}_k^{\pi_w}(x_k^{(i)}, d_k^{(i)})\Big|_{d_k^{(i)}=\mu_{k,w_k}(x_k^{(i)})} \\
&\quad + \frac{2}{M}\sum_{i=1}^{M}\sum_{k=1}^{N-1} \left\{\sum_{l=0}^{k-1}\left[g_l^{(i)} + V_{l+1}^{\pi_w}(x_{l+1}^{(i)}) - Q_l^{\pi_w}(x_l^{(i)}, \mu_{l,w_l}(x_l^{(i)}))\right]\right\} \times \\
&\qquad\qquad \nabla_w \mu_{k,w_k}(x_k^{(i)}) \nabla_{d_k^{(i)}} Q_k^{\pi_w}(x_k^{(i)}, d_k^{(i)})\Big|_{d_k^{(i)}=\mu_{k,w_k}(x_k^{(i)})}
\end{aligned}
$$

where the superscript indicates the $i$th episode (i.e., trajectory instance) generated from MC sampling and $M$ is the number of MC samples. Note that we have switched the order of summation in the second part to facilitate easier computation. The sampling technique is the same as that discussed in Sec. 2.2.2. By combining with the MC estimator for the PG of expected utility in Eqn. (2.17), we arrive at the overall MC estimator for the variance-penalized objective function:

$$
\begin{aligned}
&\nabla_w U_\lambda(w) \qquad (5.15)\\
&\approx \frac{1}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1} \nabla_w \mu_{k,w_k}(x_k^{(i)}) \nabla_{d_k^{(i)}} \left(Q_k^{\pi_w}(x_k^{(i)}, d_k^{(i)}) - \lambda \tilde{Q}_k^{\pi_w}(x_k^{(i)}, d_k^{(i)})\right)\Big|_{d_k^{(i)}=\mu_{k,w_k}(x_k^{(i)})} \\
&\quad - \frac{2\lambda}{M}\sum_{i=1}^{M}\sum_{k=1}^{N-1} \left\{\sum_{l=0}^{k-1}\left[g_l^{(i)} + V_{l+1}^{\pi_w}(x_{l+1}^{(i)}) - Q_l^{\pi_w}(x_l^{(i)}, \mu_{l,w_l}(x_l^{(i)}))\right]\right\} \times \\
&\qquad\qquad \nabla_w \mu_{k,w_k}(x_k^{(i)}) \nabla_{d_k^{(i)}} Q_k^{\pi_w}(x_k^{(i)}, d_k^{(i)})\Big|_{d_k^{(i)}=\mu_{k,w_k}(x_k^{(i)})}
\end{aligned}
$$

Compared with PG-sOED, the MC estimator for rsOED further entails computing the gradient of the variance action-value function $\nabla_{d_k^{(i)}} \tilde{Q}_k^{\pi_w}(x_k^{(i)}, d_k^{(i)})$. Therefore, we also use a DNN (*$\tilde{Q}$-network*)

with parameters $\tilde{\nu}$ to represent the variance action-value function. The training of $\tilde{Q}$-network is similar to the training of Q-network described in Sec. 2.2.2.2. Note that $\tilde{Q}_k^{\pi_w}(x_k, d_k)$ can be written in a recursive form (see Eqn. (5.12)), the analogous loss function for optimizing $\tilde{\nu}$ is

$$\mathcal{L}(\tilde{\nu}) = \frac{1}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\left[\tilde{Q}_{\tilde{\nu}}^{\pi_w}(k, x_k^{(i)}, d_k^{(i)}) - \left(\hat{V}_{k+1}^{\pi_w}(x_{k+1}^{(i)}) + \tilde{V}_{k+1}^{\pi_w}(x_{k+1}^{(i)})\right)\right]^2, \tag{5.16}$$

where $\hat{V}_{k+1}^{\pi_w}(x_{k+1}^{(i)}) = \left[g_k(x_k^{(i)}, d_k^{(i)}, y_k^{(i)}) + Q_{k+1}^{\pi_w}(x_{k+1}^{(i)}, d_{k+1}^{(i)}) - Q_k^{\pi_w}(x_k^{(i)}, d_k^{(i)})\right]^2$ for $k = 0, \ldots, N-2$, $\hat{V}_N^{\pi_w}(x_N^{(i)}) = \left[g_{N-1}(x_{N-1}^{(i)}, d_{N-1}^{(i)}, y_{N-1}^{(i)}) + V_N^{\pi_w}(x_N^{(i)}) - Q_{N-1}^{\pi_w}(x_{N-1}^{(i)}, d_{N-1}^{(i)})\right]^2$, $\tilde{V}_{k+1}^{\pi_w}(x_{k+1}^{(i)}) = \tilde{Q}_{k+1}^{\pi_w}(x_{k+1}^{(i)}, d_{k+1}^{(i)})$ for $k = 0, \ldots, N-2$, $\tilde{V}_N^{\pi_w}(x_N^{(i)}) = 0$, and $d_k^{(i)} = \mu_w(k, x_k^{(i)})$ for $k = 0, \ldots, N-1$.

### 5.2.3 Evaluation of Kullback-Leibler rewards

The last remaining task involves estimating the terminal reward $g_N(x_N)$, as specified in Eqn. (2.8), representing the KL divergence from the prior $p(\theta|I_0)$ to the final posterior $p(\theta|I_N)$. In Chapter 2, we discretize the $\theta$-space and estimate the posterior PDF pointwise, however, it could be impractical when $N_\theta > 4$ as the number of grid points increase exponentially as the dimension of $\theta$-space. The Prior Contrastive Estimator (PCE) can be employed for estimating the KL divergence as well:

$$\begin{aligned} g_N(x_N) &= \int_\Theta p(\theta|I_N) \ln\left[\frac{p(\theta|I_N)}{p(\theta|I_0)}\right] d\theta \\ &= \int_\Theta p(\theta|I_0)\frac{p(I_N|\theta, I_0)}{p(I_N|I_0)} \ln\left[\frac{p(I_N|\theta, I_0)}{p(I_N|I_0)}\right] d\theta \\ &\approx \frac{1}{M_{\text{PCE}}}\sum_{j=1}^{M_{\text{PCE}}}\frac{p(I_N|\theta^{(j)}, I_0)}{p(I_N|I_0)} \ln\left[\frac{p(I_N|\theta^{(j)}, I_0)}{p(I_N|I_0)}\right], \end{aligned} \tag{5.17}$$

where $\theta^{(j)} \sim p(\theta|I_0)$, and the marginal likelihood $p(I_N|I_0)$ is estimated by

$$p(I_N|I_0) \approx \frac{1}{M_{\text{PCE}}}\sum_{j=1}^{M_{\text{PCE}}} p(I_N|\theta^{(j)}, I_0). \tag{5.18}$$

However, obtaining an accurate estimate using PCE necessitates a large number of inner loop samples and is computationally expensive. We thus utilize the variational techniques in Chapter 3 to accelerate these computations.

While it might be tempting to directly adopt the *one-point-terminal-information-gain* (one-point-TIG) estimator from Sec. 3.1.4, this is unsuitable because the variance of the one-point-TIG is not

equal to the variance of the TIG (the former involves both the variance induced by the randomness in $y$ and $\theta$, while the latter only incorporates the randomness in $y$ as $\theta$ is marginalized out). Instead, our idea is to use the variational techniques in the way presented in Sec. 3.2.5.1 to learn a posterior approximation (i.e., train $q(\theta|I_N;\phi)$ that approximates $p(\theta|I_N)$), with the understanding that we are only considering PoI inference for a single forward model (i.e., $\mathcal{M}=1$, $\alpha_{\mathcal{M}}=\alpha_Z=0$ and $\alpha_\Theta=1$), and use the variational posterior $q(\theta|I_N;\phi)$ to learn $V_{\nu'}^{\pi_w}(N,x_N)$, which is equivalently the terminal reward (and terminal KL divergence if terminal reward only involves information gain). $V_{\nu'}^{\pi_w}(N,x_N)$ is parameterized by $\nu'$, and trained by minimizing the following loss function:

$$\mathcal{L}(\nu')=\frac{1}{M}\sum_{i=1}^{M}\left[V_{\nu'}^{\pi_w}(N,x_N^{(i)})-\ln\frac{q(\theta^{(i)}|I_N;\phi)}{p(\theta^{(i)}|I_0)}\right]^2. \tag{5.19}$$

In practice, we use a single DNN ($\tilde{Q}$-network) to learn the action-value functions $Q_\nu^{\pi_w}(k,x_k,d_k)$ for $k=0,\dots,N-1$, as well as the state-value function at the terminal stage (i.e., $V_{\nu'}^{\pi_w}(x_N)$). Here, $\nu$ and $\nu'$ refer to the parameters of the same DNN. The input layer for the policy network $\mu_w(k,x_k)$ is

$$I_k^{actor}=[\underbrace{e_k}_{N+1},\overbrace{d_0}^{N_d},\dots,d_{k-1},\underbrace{0,\dots,0}_{N_d(N-k)},\overbrace{y_0}^{N_y},\dots,y_{k-1},\underbrace{0,\dots,0}_{N_y(N-k)}]^T, \tag{5.20}$$

where $e_k$ is a zero-indexed one-hot encoding unit vector of size $N+1$. The $k$th element of $e_k$ is 1, while all other elements are 0. The input for the Q-network and $\tilde{Q}$-network is

$$I_k^{critic}=[I_k^{actor},d_k] \tag{5.21}$$

for $k=0,\dots,N-1$ and

$$I_N^{critic}=[I_N^{actor},\mathbf{0}], \tag{5.22}$$

where $\mathbf{0}$ is a zero vector of size $N_d$.

### 5.2.4 Algorithms of rsOED

Advanced RL techniques, such as replay buffer and target network are also utilized in rsOED. The overall algorithm is provided in Algorithm 3.

**Algorithm 3:** The rsOED algorithm.

Initialize variational parameters $\phi$, actor (policy) parameters $w$, critic parameters $\nu$;
**for** $l = 1, \ldots, n_{\text{update}}$ **do**
  Simulate $n_{\text{episode}}$ episodes: sample $\theta$ from the prior, and then for $k = 0, \ldots, N-1$ sample $d_k = \mu_{k,w}(I_k) + \epsilon_{\text{explore}}$ and $y_k \sim p(y_k|\theta, d_k, I_k)$;
  Update newly generated information sequences $\left\{I_N^{(i)}\right\}_{i=1}^{n_{\text{episode}}}$ into replay buffer;
  Sample $n_{\text{batch}}$ episodes from the replay buffer, update $\phi$ using sampled batch;
  Estimate gradients and update $\nu$ (Eqn. (5.16)) via gradient descent and $w$ (Eqn. (5.15)) via gradient ascent using sampled batch;
**end for**
Return optimized policy network $\pi_w$;

## 5.3 Numerical results

### 5.3.1 Source location finding with stochastic rewards

The source location finding case in Sec. 3.3.2 with 2 random sources is adopted to demonstrate the effectiveness of rsOED. We design $N = 10$ experiments and use the same setting as that described in Sec. 3.3.2, with the exception that now the rewards are stochastic:

$$g_k(x_k, d_k, y_k) \sim \mathcal{N}(0, 25e^{-2\| d_k \|}). \tag{5.23}$$

In this form, we can see that the randomness in the rewards (i.e., the standard deviation of its distribution) is greatest when the design $d_k$ is located at the origin, and lowest when it farthest from the origin at the corner of the squared domain $\mathcal{D}_k = [-4, 4]^2$.

We conduct the rsOED with a number of different variance penalty coefficient settings $\{-0.3, 0, 0.1, 0.3, 1\}$ to reflect different degrees of risk preference. After training, we evaluate the performance of rsOED by randomly generating 2000 episodes. The final KL divergence is estimated using the PCE for these evaluation episodes. Figure 5.1 illustrates the histograms of the total rewards corresponding to each $\lambda$. The title of each subfigure displays the mean and variance of the total rewards. We observe that as $\lambda$ increases, the histogram of total rewards becomes narrower, reflecting a more risk-averse policy. The highest mean reward is observed when $\lambda = 0$, and decreases as $\lambda$ increases.

Figure 5.2 further draws example policies (i.e., sensor measurement locations). When $\lambda = -0.3$, the designs tend to be concentrated around the origin to maximize the randomness in the immediate rewards (risk-seeking). When $\lambda = 0$, the policy takes measurements near the true source location. Many measurements are still taken around the origin because the true source location follows a Gaussian prior centered at the origin. When $\lambda = 0.1$, the design locations tend to be positioned

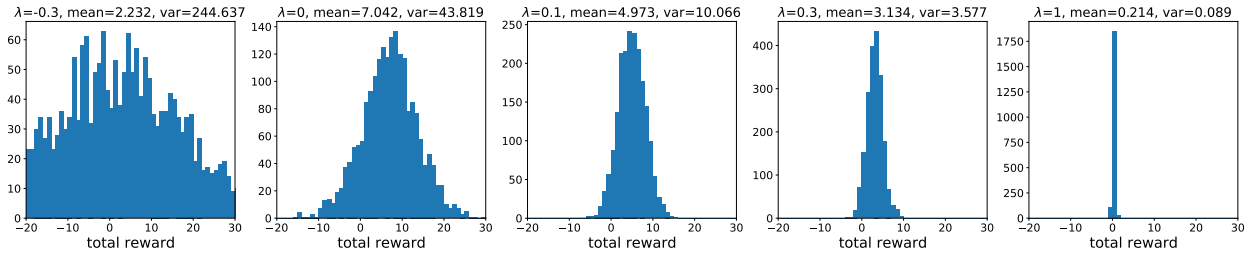



Figure 5.1: Histograms of the total reward of 2000 sampled episodes under various variance penalty coefficient $\lambda$s.

away from the origin to reduce the randomness in the immediate rewards, while still leveraging the knowledge of the true source location. When $\lambda$ increases to 0.3, the policy becomes more spread out in its design locations, and when $\lambda = 1$, the design locations are concentrated at corners to minimize the randomness in the immediate rewards. These policy behaviors are all consistent with our intuitive understanding of the problem mechanics.

Table 5.1 presents the mean and variance of the total rewards estimated with PCE and variational approximation. The results demonstrate that the estimates obtained through PCE consistently align closely with those obtained using the variational method for all $\lambda$. This suggests that the rOED algorithm using variational approximation is effective.

Table 5.1: Mean and variance of the total rewards estimated with PCE and variational approximation under different variance penalty coefficients.

| | $\lambda = -0.3$ | $\lambda = 0$ | $\lambda = 0.1$ | $\lambda = 0.3$ | $\lambda = 1$ |
|---|---|---|---|---|---|
| Mean of total rewards (PCE) | 2.232 | 7.042 | 4.973 | 3.134 | 0.214 |
| Mean of total rewards (variational) | 1.916 | 5.934 | 4.292 | 2.860 | 0.221 |
| Var of total rewards (PCE) | 244.637 | 43.819 | 10.066 | 3.577 | 0.089 |
| Var of total rewards (variational) | 245.495 | 43.416 | 9.271 | 3.327 | 0.091 |

## 5.4 Summary

In this chapter, we integrate the principles of sOED and rOED to reach the robust sequential optimal experimental design (**rsOED**). rsOED shares the same framework as sOED, but with a distinct objective: instead of maximizing the expected utility, it focuses on the mean-minus-variance of the utility. We then provide the numerical techniques for solving rsOED problems, specifically the policy gradient for the variance of utilities and its Monte Carlo estimator, and demonstrate rsOED using a numerical example.

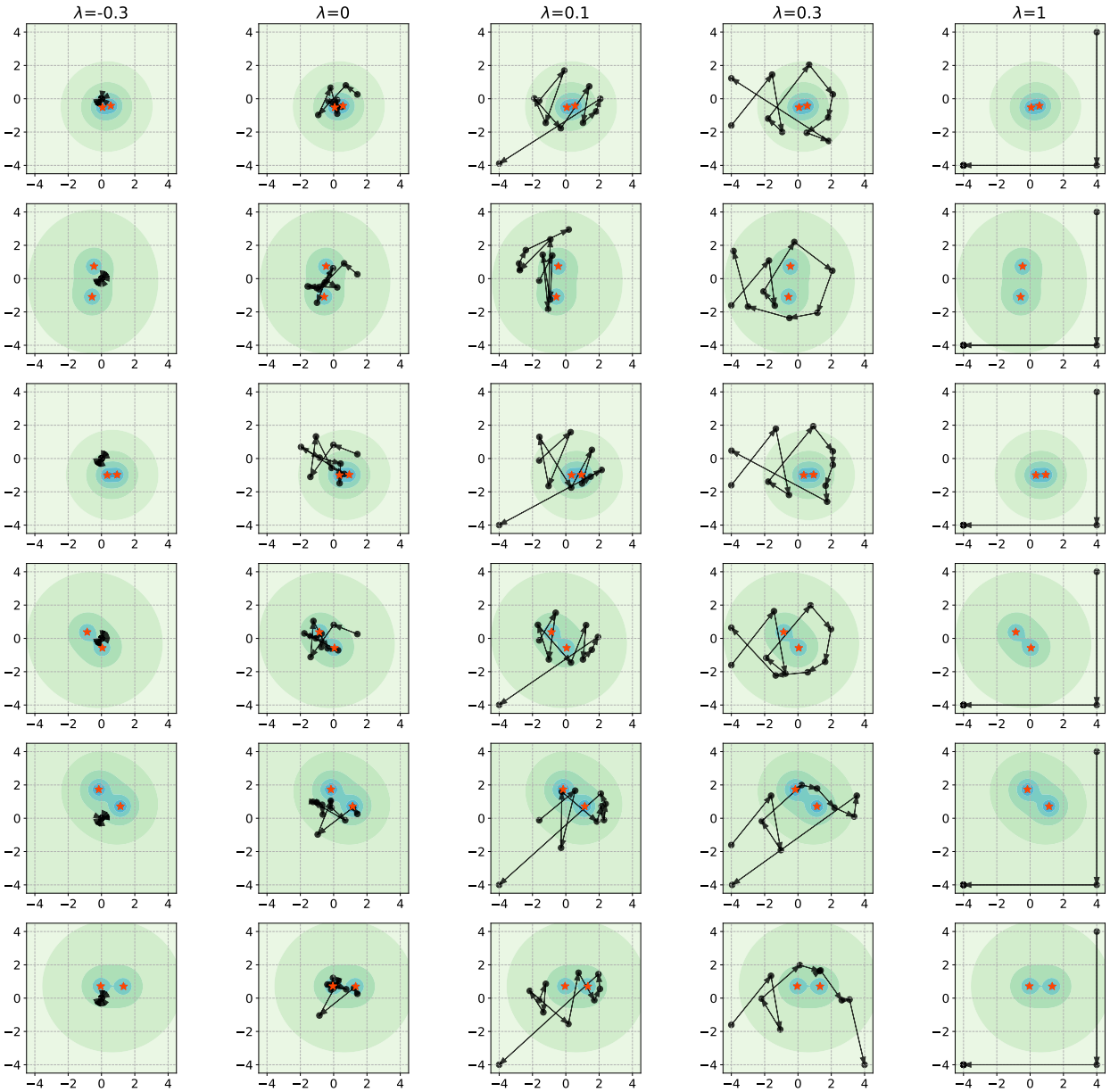



Figure 5.2: Example policies under various variance penalty coefficient $\lambda$s. Each column corresponds to a specific $\lambda$ value, while each row corresponds to a true source location.

The key contributions and novelty of our rsOED method are summarized as follows.

- We formulate the variance-penalized rsOED framework.
- We present the algorithm of rsOED by providing the policy gradient expressions and its Monte Carlo estimator of the variance of utilities.
- We utilize variational approximation to accelerate the computation of information gain.
- We validate rsOED on a benchmark example.

# CHAPTER 6

# Conclusions and future work

## 6.1 Conclusions

This thesis first presents a mathematical framework and computational methods to optimally design a finite number of sequential experiments (sOED) . We formulate sOED as a finite-horizon POMDP. This sOED form is provably optimal, incorporates both elements of feedback and lookahead, and generalizes the batch (static) and greedy (myopic) design strategies. We structure the sOED problem in a fully Bayesian manner and with information-theoretic rewards (utilities), and prove the equivalence of incremental and terminal information gain setups. In particular, sOED can accommodate expensive nonlinear forward models with general non-Gaussian posteriors of continuous random variables. We then introduce numerical methods for solving the sOED problem, which entails finding the optimal policy that maximizes the expected total reward. At the core of our approach is PG, an actor-critic RL technique that parameterizes and learns both the policy and value functions in order to extract the gradient with respect to the policy parameters. We derive and prove the PG expression for finite-horizon sOED, and propose an MC estimator for it. Accessing derivative information enables the use of gradient-based optimization algorithms to achieve efficient policy search. Specifically, we parameterize the policy and value functions as DNNs, and detail architecture design that accommodates a nonparametric representation of the Bayesian posterior belief states. Combining this representation technique with the terminal information gain formulation, PG-sOED sidesteps the need for computing intermediate Bayesian posteriors and incremental KL divergence terms, making it much more computationally efficient than greedy design. We demonstrate PG-sOED to two groups of examples. The first is a linear-Gaussian benchmark to validate PG-sOED against the analytical solution and to illustrate its orders-of-magnitude speedups over an existing ADP-sOED baseline. The second entails sensor movement for contaminant source inversion in a convection-diffusion field. Through multiple cases, we show the advantages of PG-sOED over greedy and batch designs, and provide explanation of the results leveraging the physical knowledge of convection-diffusion process. This demonstration also illustrates the ability

of PG-sOED to accommodate expensive forward models with nonlinear physics and dynamics.

To enhance the computational efficiency of sOED, we further introduce vsOED, a sample-efficient method for Bayesian sequential OED that can handle implicit models and multi-model scenarios, and accommodate diverse OED objectives (parameter inference, model discrimination, goal-oriented prediction). We provide a rigorous proof demonstrating the equivalence, in expectation, between using terminal information gain and incremental information gain, as well as between using the full integral of information gain and its one-point estimate. Therefore, these approaches lead to the same sOED problem formulation. We then present the numerical algorithms for solving vsOED problems, particularly the variational graident and policy gradient, as well as their MC estimator. We validate vsOED on a number of examples, including a source location case under both uni-model and multi-model scenarios, targeting parameter inference, model discrimination and goal-oriented QoI predictions, a CES problem with a highly non-Gaussian posterior, a SIR problem with implicit likelihood and expensive forward model, and a convection-diffusion source inversion problem with a real physics based PDE model. By Leveraging variational approximation and policy gradient, vsOED demonstrates superior performance in the explicit likelihood setting under a fixed computational budget, while achieving similar performance as iDAD in the implicit likelihood setting without needing forward model derivatives.

We then present a mathematical framework and computational methods to solve the rOED problems. To enhance the stability of the utility, we choose to regularize the expected utility function with a penalty on the variance of the utility, and propose a variance-penalized objective formulation. By adjusting the penalty coefficient, this formulation yields a design that may have a slightly lower expected utility compared to the design obtained solely by maximizing the expected utility, and consequently a higher worst-case utility. In order to estimate the variance-penalized objective in an efficient manner, we propose a double-nested Monte Carlo (MC) estimator, where outer MC samples will be reused as inner samples to reduce the forward model evaluations from $\mathcal{O}(n^2)$ to a $\mathcal{O}(n)$ and avoid arithmetic underflow. We also analyze the bias and variance of the proposed estimator. Moreover, Bayesian optimization (BO) is utilized to efficiently find the global optimal design, and common random samples are also employed to introduce artificial correlation among different designs and smoothen the objective function. We then apply robust OED to three examples. The first example is a linear-Gaussian problem with a closed-form solution, which is used to validate the convergence speed of the proposed estimator as the sample number increases. The second example has a nonlinear forward model and is used to illustrate the value of robust OED. We provide some insights to explain why different designs will have significantly different utility variances. Besides, the performance of BO has also been validated in this example. The third example is a contaminant source inversion problem in a diffusion domain, with and without building obstacles. This example further illustrates the usage of robust OED and Bayesian optimization for

more complicated physical problems.

Lastly, we combine the concepts of rOED and sOED, and introduce the rsOED framework that enables the design of a sequence of experiments in a robust manner. We provide the definition of the value functions for the variance of the total rewards, and the policy gradient expressions with the corresponding MC estimator. Variational approximation is utilized to expedite the calculation of the KL divergence information gain. We demonstrate rsOED on a source location problem to showcase how the variance penalty coefficient influences the policy and the accuracy of the variational approximation.

## 6.2 Limitations and future work

The main limitation of the PG-sOED approach is its inability to scale to high-dimensional settings, hindered by the need to perform high-dimensional Bayesian inference and KL divergence estimate. The limitations of PG-sOED are alleviated by vsOED via variational approximation, however, vsOED is sensitive to inaccurate posterior representations, which can lead to suboptimal policies when the posteriors are challenging to approximate. Future work for developing accurate and adaptive posterior representations especially in high dimensions, as well as utilizing other variational bounds [76, 117], will be important. The current algorithms for both PG-sOED and vsOED also do not consider discrete designs or stochastic policies, which when enabled, can reach a wider class of design problems. vsOED performance can also be further enhanced through advanced RL techniques [126, 125, 63, 59]. It would also be interesting to explore if we could automatically determine the number of episodes needed to train sOED and vsOED. We anticipate that as the dimension of the parameter space and the horizon of experiments increase, the required sample size may also increase. However, it might reach a plateau due to the diminishing return on information gain. Some work has been done to determine the sample size for modeling human behavior via inverse reinforcement learning [69], and similar ideas can be adapted and utilized for sample size determination in sOED and vsOED. Infinite-horizon sOED (e.g., when we don't know the horizon a priori) is also of great interest, one approach to handle such scenarios is by computing the incremental information gain after each stage of the experiment. The experiment can then be halted once the information gain surpasses a predefined threshold. Additionally, we can consider whether to stop the experiment as a design variable.

For rOED, we have several directions to explore. The first is how to choose the penalty coefficient $\lambda$ in a reasonable way. The second direction would be how to estimate the variance-penalized objective more accurately. Posterior samples could be used in the inner loop of double-nested MC estimator instead of prior samples, however, it will increase the forward model evaluations. Potential strategies to address it might be using Laplace approximated based importance sampling

[11] and multilevel Monte Carlo (MLMC) [62]. The third direction is to consider different risk criteria. A simple alternative for the utility variance could be the standard deviation of the utility. Additionally, we can explore other approaches such as minimizing the probability of undesirable utilities or maximizing the expected utility of worst-case scenarios, also known as the Conditional Value at Risk (CVaR) criterion. Moreover, the robustness against the prior misspecification, model misspecification and design noise is also gaining more and more attentionn.

For rsOED, we plan to apply it to more complex models to demonstrate its practical application in real-world problems.

# APPENDIX A

# Appendix of sequential optimal experimental design (sOED)

## A.1 Equivalence of incremental and terminal formulations in sOED

*Proof of Theorem* 1. Upon substituting Eqn. (2.7) and (2.8) into Eqn. (2.4), the expected utility for a given deterministic policy $\pi$ using the TIG formulation is

$$\begin{aligned} U_T(\pi) &= \mathbb{E}_{y_0,\dots,y_{N-1}|\pi,x_0}\left[\int_\Theta p(\theta|I_N)\ln\frac{p(\theta|I_N)}{p(\theta|I_0)}\,d\theta\right] \\ &= \mathbb{E}_{I_1,\dots,I_N|\pi,x_0}\left[\int_\Theta p(\theta|I_N)\ln\frac{p(\theta|I_N)}{p(\theta|I_0)}\,d\theta\right] \end{aligned} \tag{A.1}$$

where recall $I_k = \{d_0, y_0, \dots, d_{k-1}, y_{k-1}\}$ (and $I_0 = \emptyset$). Similarly, substituting Eqn. (2.9) and (2.10), the expected utility for the same policy $\pi$ using the IIG formulation is

$$\begin{aligned} U_I(\pi) &= \mathbb{E}_{y_0,\dots,y_{N-1}|\pi,x_0}\left[\sum_{k=1}^{N}\int_\Theta p(\theta|I_k)\ln\frac{p(\theta|I_k)}{p(\theta|I_{k-1})}\,d\theta\right] \\ &= \mathbb{E}_{I_1,\dots,I_N|\pi,x_0}\left[\sum_{k=1}^{N}\int_\Theta p(\theta|I_k)\ln\frac{p(\theta|I_k)}{p(\theta|I_{k-1})}\,d\theta\right]. \end{aligned} \tag{A.2}$$

In both cases, $\mathbb{E}_{y_0,\ldots,y_{N-1}|\pi,x_0}$ can be equivalently replaced by $\mathbb{E}_{I_1,\ldots,I_N|\pi,x_0}$ since

$$
\begin{aligned}
\mathbb{E}_{I_1,\ldots,I_N|\pi,x_0}\left[\cdots\right] &= \mathbb{E}_{d_0,y_0,d_1,y_1,\ldots,d_{N-1},y_{N-1}|\pi,x_0}\left[\cdots\right] \\
&= \mathbb{E}_{d_0|\pi}\mathbb{E}_{y_0,d_1,y_1,\ldots,d_{N-1},y_{N-1}|\pi,x_0,d_0}\left[\cdots\right] \\
&= \mathbb{E}_{y_0,d_1,y_1,\ldots,d_{N-1},y_{N-1}|\pi,x_0,\mu_0(x_0)}\left[\cdots\right] \\
&= \mathbb{E}_{y_0,d_1,y_1,\ldots,d_{N-1},y_{N-1}|\pi,x_0}\left[\cdots\right] \\
&= \mathbb{E}_{y_0|\pi,x_0}\mathbb{E}_{d_1|\pi,x_0,y_0}\mathbb{E}_{y_1,\ldots,d_{N-1},y_{N-1}|\pi,x_0,y_0,d_1}\left[\cdots\right] \\
&= \mathbb{E}_{y_0|\pi,x_0}\mathbb{E}_{y_1,\ldots,d_{N-1},y_{N-1}|\pi,x_0,y_0,\mu_1(x_1)}\left[\cdots\right] \\
&= \mathbb{E}_{y_0|\pi,x_0}\mathbb{E}_{y_1,\ldots,d_{N-1},y_{N-1}|\pi,x_0,y_0}\left[\cdots\right] \\
&= \mathbb{E}_{y_0|\pi,x_0}\mathbb{E}_{y_1|\pi,x_0,y_0}\mathbb{E}_{d_2,\ldots,d_{N-1},y_{N-1}|\pi,x_0,y_0,y_1}\left[\cdots\right] \\
&\quad\vdots \\
&= \mathbb{E}_{y_0|\pi,x_0}\mathbb{E}_{y_1|\pi,x_0,y_0}\cdots\mathbb{E}_{y_{N-1}|\pi,x_0,y_0,y_1,\ldots,y_{N-2},\mu_{N-1}(x_{N-1})}\left[\cdots\right] \\
&= \mathbb{E}_{y_0|\pi,x_0}\mathbb{E}_{y_1|\pi,x_0,y_0}\cdots\mathbb{E}_{y_{N-1}|\pi,x_0,y_0,y_1,\ldots,y_{N-2}}\left[\cdots\right] \\
&= \mathbb{E}_{y_0,\ldots,y_{N-1}|\pi,x_0}\left[\cdots\right],
\end{aligned}
$$

where the third equality is due to the deterministic policy (Dirac delta function) $d_0 = \mu_0(x_0)$, the fourth equality is due to $\mu_0(x_0)$ being known if $\pi$ and $x_0$ are given. The seventh equality is due to $\mu_1(x_1)$ being known if $\pi$ and $x_1$ are given, and $x_1$ is known if $x_0$, $d_0 = \mu_0(x_0)$ and $y_0$ are given, and $\mu_0(x_0)$ is known if $\pi$ and $x_0$ are given, so overall $\mu_1(x_1)$ is known if $\pi$, $x_0$ and $y_0$ are given. The eighth to second-to-last equalities all apply the same reasoning recursively. The last equality brings the expression back to a conditional joint expectation.

Taking the difference between Eqn. (A.1) and Eqn. (A.2), we obtain

$$
\begin{aligned}
&U_I(\pi) - U_T(\pi) \\
&= \mathbb{E}_{I_1,\ldots,I_N|\pi,x_0}\left[\sum_{k=1}^{N}\int_\Theta p(\theta|I_k)\ln\frac{p(\theta|I_k)}{p(\theta|I_{k-1})}\,d\theta - \int_\Theta p(\theta|I_N)\ln\frac{p(\theta|I_N)}{p(\theta|I_0)}\,d\theta\right] \\
&= \int_\Theta \mathbb{E}_{I_1,\ldots,I_N|\pi,x_0}\left[\sum_{k=1}^{N} p(\theta|I_k)\ln\frac{p(\theta|I_k)}{p(\theta|I_{k-1})} - p(\theta|I_N)\ln\frac{p(\theta|I_N)}{p(\theta|I_0)}\right]d\theta \\
&= \int_\Theta \mathbb{E}_{I_1,\ldots,I_N|\pi,x_0}\left[\sum_{k=1}^{N-1} p(\theta|I_k)\ln\frac{p(\theta|I_k)}{p(\theta|I_{k-1})} + p(\theta|I_N)\ln\frac{p(\theta|I_0)}{p(\theta|I_{N-1})}\right]d\theta \\
&= \int_\Theta \mathbb{E}_{I_1,\ldots,I_{N-1}|\pi,x_0}\int_{I_N} p(I_N|I_{N-1},\pi)\left[\sum_{k=1}^{N-1} p(\theta|I_k)\ln\frac{p(\theta|I_k)}{p(\theta|I_{k-1})} + p(\theta|I_N)\ln\frac{p(\theta|I_0)}{p(\theta|I_{N-1})}\right]dI_N\,d\theta \\
&= \int_\Theta \mathbb{E}_{I_1,\ldots,I_{N-1}|\pi,x_0}\left[\sum_{k=1}^{N-1} p(\theta|I_k)\ln\frac{p(\theta|I_k)}{p(\theta|I_{k-1})} + \int_{I_N} p(\theta,I_N|I_{N-1},\pi)\ln\frac{p(\theta|I_0)}{p(\theta|I_{N-1})}\,dI_N\right]d\theta \\
&= \int_\Theta \mathbb{E}_{I_1,\ldots,I_{N-1}|\pi,x_0}\left[\sum_{k=1}^{N-1} p(\theta|I_k)\ln\frac{p(\theta|I_k)}{p(\theta|I_{k-1})} + p(\theta|I_{N-1})\ln\frac{p(\theta|I_0)}{p(\theta|I_{N-1})}\right]d\theta \\
&= \int_\Theta \mathbb{E}_{I_1,\ldots,I_{N-1}|\pi,x_0}\left[\sum_{k=1}^{N-2} p(\theta|I_k)\ln\frac{p(\theta|I_k)}{p(\theta|I_{k-1})} + p(\theta|I_{N-1})\ln\frac{p(\theta|I_0)}{p(\theta|I_{N-2})}\right]d\theta \\
&= \int_\Theta \mathbb{E}_{I_1,\ldots,I_{N-2}|\pi,x_0}\left[\sum_{k=1}^{N-3} p(\theta|I_k)\ln\frac{p(\theta|I_k)}{p(\theta|I_{k-1})} + p(\theta|I_{N-2})\ln\frac{p(\theta|I_0)}{p(\theta|I_{N-3})}\right]d\theta \\
&\quad\vdots \\
&= \int_\Theta \mathbb{E}_{I_1|\pi,x_0}\left[p(\theta|I_1)\ln\frac{p(\theta|I_0)}{p(\theta|I_0)}\right]d\theta \\
&= 0,
\end{aligned}
$$

where the third equality takes the last term from the sigma-summation and combines it with the last term, the fourth equality expands the expectation and uses $p(I_N|I_1,\ldots,I_{N-1},\pi) = p(I_N|I_{N-1},\pi)$, the fifth equality makes use of $p(\theta|I_N) = p(\theta|I_N,\pi)$, and the seventh to second-to-last equalities repeat the same procedures recursively. Hence, $U_T(\pi) = U_I(\pi)$. □

## A.2 Policy gradient expression

Before presenting the proof of gradient expression, we first introduce the *state-value function* (or *V-function*). The V-function following policy $\pi_w$ and at the $k$th experiment is

$$V_k^{\pi_w}(x_k) = \mathbb{E}_{y_k,\ldots,y_{N-1}|\pi_w,x_k}\left[\sum_{t=k}^{N-1} g_t(x_t, \mu_{t,w_t}(x_t), y_t) + g_N(x_N)\right] \tag{A.3}$$

$$= \mathbb{E}_{y_k|\pi_w,x_k}\left[g_k(x_k, \mu_{k,w_k}(x_k), y_k) + V_{k+1}^{\pi_w}(x_{k+1})\right] \tag{A.4}$$

$$V_N^{\pi_w}(x_N) = g_N(x_N) \tag{A.5}$$

for $k = 0, \ldots, N-1$, where $x_{k+1} = \mathcal{F}_k(x_k, \mu_{k,w_k}(x_k), y_k)$. The V-function is the expected cumulative remaining reward starting from a given state $x_k$ and following policy $\pi_w$ for all remaining experiments. The V-function and Q-function are related via

$$V_k^{\pi_w}(x_k) = Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k)). \tag{A.6}$$

Our proof for Theorem 2 follows the proof strategy for a general infinite-horizon MDP given by [132]. A shorthand notation for writing the state transition probability is utilized for better understanding:

$$p(x_k \to x_{k+1}|\pi_w) = p(x_{k+1}|x_k, \mu_{k,w}(x_k)). \tag{A.7}$$

When taking an expectation over consecutive state transitions, we further use the simplifying notation

$$\begin{aligned}
&\int_{x_{k+1}} p(x_k \to x_{k+1}|\pi_w) \int_{x_{k+2}} p(x_{k+1} \to x_{k+2}|\pi_w) \\
&\qquad \cdots \int_{x_{k+m}} p(x_{k+(m-1)} \to x_{k+m}|\pi_w)\,[\cdots]\; dx_{k+1}\, dx_{k+2} \cdots dx_{k+m} \\
&= \int_{x_{k+m}} p(x_k \to x_{k+m}|\pi_w)\,[\cdots]\; dx_{k+m} \qquad \text{(A.8)} \\
&= \mathbb{E}_{x_{k+m}|\pi_w,x_k}\,[\cdots]\,. \qquad \text{(A.9)}
\end{aligned}$$

To avoid notation congestion, below we will omit the subscript on $w$ and shorten $\mu_{k,w_k}(x_k)$ to $\mu_{k,w}(x_k)$, with the understanding that $w$ takes the same subscript as the $\mu$ function.

*Proof of Theorem* 2. We begin by recognizing that the gradient of expected utility in Eqn. (2.12)

can be written using the V-function:

$$\nabla_w U(w) = \nabla_w V_0^{\pi_w}(x_0). \tag{A.10}$$

The goal is then to derive the gradient expression for the V-functions.

We apply the definitions and recursive relations for the V- and Q-functions, and obtain a recursive relationship for the gradient of V-function:

$$\begin{aligned}
\nabla_w V_k^{\pi_w}(x_k) &= \nabla_w Q_k^{\pi_w}(x_k, \mu_{k,w}(x_k)) \\
&= \nabla_w \Bigg[ \int_{y_k} p(y_k|x_k, \mu_{k,w}(x_k)) g_k(x_k, \mu_{k,w}(x_k), y_k)\, dy_k \\
&\qquad + \int_{x_{k+1}} p(x_{k+1}|x_k, \mu_{k,w}(x_k)) V_{k+1}^{\pi_w}(x_{k+1})\, dx_{k+1} \Bigg] \\
&= \nabla_w \int_{y_k} p(y_k|x_k, \mu_{k,w}(x_k)) g_k(x_k, \mu_{k,w}(x_k), y_k)\, dy_k \\
&\qquad + \nabla_w \int_{x_{k+1}} p(x_{k+1}|x_k, \mu_{k,w}(x_k)) V_{k+1}^{\pi_w}(x_{k+1})\, dx_{k+1} \\
&= \int_{y_k} \nabla_w \mu_{k,w}(x_k) \nabla_{d_k} \left[ p(y_k|x_k, d_k) g_k(x_k, d_k, y_k) \right] \Big|_{d_k = \mu_{k,w}(x_k)} dy_k \\
&\qquad + \int_{x_{k+1}} \Big[ p(x_{k+1}|x_k, \mu_{k,w}(x_k)) \nabla_w V_{k+1}^{\pi_w}(x_{k+1}) \\
&\qquad + \nabla_w \mu_{k,w}(x_k) \nabla_{d_k} p(x_{k+1}|x_k, d_k) \Big|_{d_k = \mu_{k,w}(x_k)} V_{k+1}^{\pi_w}(x_{k+1}) \Big]\, dx_{k+1} \\
&= \nabla_w \mu_{k,w}(x_k) \nabla_{d_k} \Bigg[ \int_{y_k} p(y_k|x_k, d_k) g_k(x_k, d_k, y_k)\, dy_k \\
&\qquad + \int_{x_{k+1}} p(x_{k+1}|x_k, d_k) V_{k+1}^{\pi_w}(x_{k+1}) dx_{k+1} \Bigg] \Bigg|_{d_k = \mu_{k,w}(x_k)} \\
&\qquad + \int_{x_{k+1}} p(x_{k+1}|x_k, \mu_{k,w}(x_k)) \nabla_w V_{k+1}^{\pi_w}(x_{k+1})\, dx_{k+1} \\
&= \nabla_w \mu_{k,w}(x_k) \nabla_{d_k} Q_k^{\pi_w}(x_k, d_k) \Big|_{d_k = \mu_{k,w}(x_k)} \\
&\qquad + \int_{x_{k+1}} p(x_k \to x_{k+1}|\pi_w) \nabla_w V_{k+1}^{\pi_w}(x_{k+1})\, dx_{k+1}.
\end{aligned} \tag{A.11}$$

Applying the recursive formula Eqn. (A.11) to itself repeatedly and expanding out the overall

expression, we obtain

$$
\begin{aligned}
&\nabla_w V_k^{\pi_w}(x_k) \\
&= \nabla_w \mu_{k,w}(x_k) \nabla_{d_k} Q_k^{\pi_w}(x_k, d_k)\Big|_{d_k=\mu_{k,w}(x_k)} \\
&\quad + \int_{x_{k+1}} p(x_k \to x_{k+1}|\pi_w) \nabla_w \mu_{k+1,w}(x_{k+1}) \nabla_{d_{k+1}} Q_{k+1}^{\pi_w}(x_{k+1}, d_{k+1})\Big|_{d_{k+1}=\mu_{k+1,w}(x_{k+1})} dx_{k+1} \\
&\quad + \int_{x_{k+1}} p(x_k \to x_{k+1}|\pi_w) \int_{x_{k+2}} p(x_{k+1} \to x_{k+2}|\pi_w) \nabla_w V_{k+2}^{\pi_w}(x_{k+2})\, dx_{k+2}\, dx_{k+1} \\
&= \nabla_w \mu_{k,w}(x_k) \nabla_{d_k} Q_k^{\pi_w}(x_k, d_k)\Big|_{d_k=\mu_{k,w}(x_k)} \\
&\quad + \int_{x_{k+1}} p(x_k \to x_{k+1}|\pi_w) \nabla_w \mu_{k+1,w}(x_{k+1}) \nabla_{d_{k+1}} Q_{k+1}^{\pi_w}(x_{k+1}, d_{k+1})\Big|_{d_{k+1}=\mu_{k+1,w}(x_{k+1})} dx_{k+1} \\
&\quad + \int_{x_{k+2}} p(x_k \to x_{k+2}|\pi_w) \nabla_w V_{k+2}^{\pi_w}(x_{k+2})\, dx_{k+2} \\
&= \nabla_w \mu_{k,w}(x_k) \nabla_{d_k} Q_k^{\pi_w}(x_k, d_k)\Big|_{d_k=\mu_{k,w}(x_k)} \\
&\quad + \int_{x_{k+1}} p(x_k \to x_{k+1}|\pi_w) \nabla_w \mu_{k+1,w}(x_{k+1}) \nabla_{d_{k+1}} Q_{k+1}^{\pi_w}(x_{k+1}, d_{k+1})\Big|_{d_{k+1}=\mu_{k+1,w}(x_{k+1})} dx_{k+1} \\
&\quad + \int_{x_{k+2}} p(x_k \to x_{k+2}|\pi_w) \nabla_w \mu_{k+2,w}(x_{k+2}) \nabla_{d_{k+2}} Q_{k+2}^{\pi_w}(x_{k+2}, d_{k+2})\Big|_{d_{k+2}=\mu_{k+2,w}(x_{k+2})} dx_{k+2} \\
&\quad \vdots \\
&\quad + \int_{x_N} p(x_k \to x_N|\pi_w) \nabla_w V_N^{\pi_w}(x_N)\, dx_N \\
&= \sum_{l=k}^{N-1} \int_{x_l} p(x_k \to x_l|\pi_w) \nabla_w \mu_{l,w}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)\Big|_{d_l=\mu_{l,w}(x_l)} dx_l \\
&= \sum_{l=k}^{N-1} \mathbb{E}_{x_l|\pi_w,x_k}\left[ \nabla_w \mu_{l,w}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)\Big|_{d_l=\mu_{l,w}(x_l)} \right] dx_l,
\end{aligned}
\tag{A.12}
$$

where for the second-to-last equality, we absorb the first term into the sigma-notation by using

$$
\begin{aligned}
&\nabla_w \mu_{k,w}(x_k) \nabla_{d_k} Q_k^{\pi_w}(x_k, d_k)\Big|_{d_k=\mu_{k,w}(x_k)} \\
&\quad = \int_{x_k} p(x_k|x_k, \mu_{k,w}(x_k)) \nabla_w \mu_{k,w}(x_k) \nabla_{d_k} Q_k^{\pi_w}(x_k, d_k)\Big|_{d_k=\mu_{k,w}(x_k)} dx_k \\
&\quad = \int_{x_k} p(x_k \to x_k|\pi_w) \nabla_w \mu_{k,w}(x_k) \nabla_{d_k} Q_k^{\pi_w}(x_k, d_k)\Big|_{d_k=\mu_{k,w}(x_k)} dx_k,
\end{aligned}
$$

and we eliminate the last term in the summation since $\nabla_w V_N^{\pi_w}(x_N) = \nabla_w g_N(x_N) = 0$.

At last, substituting Eqn. (A.12) into Eqn. (A.10), we obtain the policy gradient expression:

$$\begin{aligned}\nabla_w U(w) &= \nabla_w V_0^{\pi_w}(x_0) \\ &= \sum_{l=0}^{N-1} \mathbb{E}_{x_l|\pi_w,x_0}\left[\nabla_w \mu_{l,w}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)\Big|_{d_l=\mu_{l,w}(x_l)}\right].\end{aligned}$$

Renaming the iterator from $l$ to $k$ arrives at Eqn. (2.16) in Theorem 2, completing the proof. □

## A.3 Equivalence of fixing and resampling model parameters in an sOED episode

When generating the $i$th episode as described in Sec. 2.2.2, employing a fixed model parameter $\theta^{(i)}$ throughout the entire $i$th episode or resampling $\theta_k^{(i)}$ at each stage $k$ from its posterior belief state $x_{k,b}^{(i)}$ both produce mathematically equivalent results. This can be seen from factoring out the expectations:

$$\begin{aligned}U(w) &= \mathbb{E}_{y_0,\ldots,y_{N-1}|\pi_w,x_0}\left[\sum_{k=0}^{N-1} g_k(x_k, d_k, y_k) + g_N(x_N)\right] \\ &= \mathbb{E}_{\theta|x_{0,b}}\mathbb{E}_{y_0|\pi_w,\theta,x_0}\mathbb{E}_{y_1|\pi_w,\theta,x_0,y_0}\cdots \\ &\qquad \cdots \mathbb{E}_{y_{N-1}|\pi_w,\theta,x_0,y_0,\ldots,y_{N-2}}\left[\sum_{k=0}^{N-1} g_k(x_k, d_k, y_k) + g_N(x_N)\right] \qquad \text{(A.13)} \\ &= \mathbb{E}_{\theta_0|x_{0,b}}\mathbb{E}_{y_0|\pi_w,\theta_0,x_0}\mathbb{E}_{\theta_1|x_{1,b}}\mathbb{E}_{y_1|\pi_w,\theta_1,x_1}\cdots \\ &\qquad \cdots \mathbb{E}_{\theta_{N-1}|x_{N-1,b}}\mathbb{E}_{y_{N-1}|\pi_w,\theta_{N-1},x_{N-1}}\left[\sum_{k=0}^{N-1} g_k(x_k, d_k, y_k) + g_N(x_N)\right], \qquad \text{(A.14)}\end{aligned}$$

where the second equality corresponds to the case of episode-fixed $\theta^{(i)}$, and the last equality corresponds to the case of resampling of $\theta_k^{(i)}$.

## A.4 Equivalence of using $x_k$ and $I_k$ as the state representation

Different sequences $I_k$'s can correspond to the same state $x_k$. However, if two sequences $I_k^{(1)}$ and $I_k^{(2)}$ share the identical state $x_k$, and $x_k$ carries sufficient information for subsequent experiments along with $\theta$ and $d_k$ (i.e., the forward model is always $G_k(\theta, d_k; x_{k,p})$), then the optimal designs based on using $x_k$ and $I_k$ as the state representation would also be identical if it is unique, and they should yield the same maximum expected tail utility. Consequently, employing either $x_k$ or $I_k$ for

$k = 0, \ldots, N-1$ would result in identical policies and yield the same maximal expected utility.

This can be proven using backward induction. We start from the $(N-1)$th stage, and assume that we have a sequence $I_{N-1}$ which corresponds to state $x_{N-1}$, then the expected tail reward of taking design $d_{N-1}$ is

$$\begin{aligned}
Q_{N-1}(I_{N-1}, d_{N-1}) &= \mathbb{E}_{y_{N-1}|I_{N-1},d_{N-1}}\left[g_{N-1}(x_{N-1}, d_{N-1}, y_{N-1}) + D_{\mathrm{KL}}(x_N||x_{N-1})\right] \\
&= \mathbb{E}_{\theta|x_{N-1,b}}\mathbb{E}_{y_{N-1}|\theta,d_{N-1},x_{N-1,p}}\left[g_{N-1}(x_{N-1}, d_{N-1}, y_{N-1}) + D_{\mathrm{KL}}(x_N||x_{N-1})\right] \\
&= \mathbb{E}_{y_{N-1}|x_{N-1},d_{N-1}}\left[g_{N-1}(x_{N-1}, d_{N-1}, y_{N-1}) + D_{\mathrm{KL}}(x_N||x_{N-1})\right] \\
&= Q_{N-1}(x_{N-1}, d_{N-1}),
\end{aligned}$$

where the second equality is due to $(y_{N-1}|\theta, d_{N-1}, x_{N-1,p}) = (y_{N-1}|\theta, d_{N-1}, I_{N-1})$ and $(\theta|x_{N-1,b}) = (\theta|I_{N-1})$. Therefore, two sequences $I^{(1)}_{N-1}$ and $I^{(2)}_{N-1}$ with the same state $x_{N-1}$ share the same maximal expected tail reward, and the same optimal design $d^*_{N-1}$ if it is unique. In other words, $V^*(I_{N-1}) = V^*(x_{N-1})$. Then we go back to stage $N-2$, and assume that we have a sequence $I_{N-2}$ which corresponds to state $x_{N-2}$, the expected tail reward of taking design $d_{N-2}$ and then follow optimal policy $\pi_I$ (note that $\pi_I$ is the optimal policy w.r.t. $I$) is

$$\begin{aligned}
Q^{\pi_I}_{N-2}(I_{N-2}, d_{N-2}) &= \mathbb{E}_{y_{N-2}|I_{N-2},d_{N-2}}\left[g_{N-2}(x_{N-2}, d_{N-2}, y_{N-2}) + Q^{\pi_I}_{N-1}(I_{N-1}, \mu_I(I_{N-1}))\right] \\
&= \mathbb{E}_{y_{N-2}|x_{N-2},d_{N-2}}\left[g_{N-2}(x_{N-2}, d_{N-2}, y_{N-2}) + V^{\pi_I}_{N-1}(I_{N-1})\right] \\
&= \mathbb{E}_{y_{N-2}|x_{N-2},d_{N-2}}\left[g_{N-2}(x_{N-2}, d_{N-2}, y_{N-2}) + V^{\pi_x}_{N-1}(x_{N-1})\right] \\
&= Q^{\pi_x}_{N-2}(x_{N-2}, d_{N-2}),
\end{aligned}$$

where the third equality is because $V^*(I_{N-1}) = V^*(x_{N-1})$. Therefore, using $I_{N-2}$ and $x_{N-2}$ share the same optimal design $d^*_{N-2}$ if it is unique, and the maximal expected tail reward that can be reached by following $\pi_I$ after $I_{N-1}$ (i.e., $V^{\pi_I}_{N-1}(I_{N-1})$) is the same as the maximal expected tail reward that can be reached by following $\pi_x$ after the corresponding $x_{N-1}$ (i.e., $V^{\pi_x}_{N-1}(x_{N-1})$), where $\pi_x$ is the optimal policy w.r.t. the state $x$.

Using the backward induction method, we can further show that if two sequences share the same state, the optimal policy should also be the same if it is unique. Moreover, both sequences should yield the same maximum expected tail reward. This implies that whether the policy function and the value function are based on $I_k$ or $x_k$ does not affect the results.

## A.5 Convergence of Q-network

Eqn. (2.21) is formed where each episode is generated by fixing the model parameter throughout, as described in Sec. 2.2.2. Among all possible episodes, the probability that $(x_k, d_k)$ takes place is $\mathbb{E}_{\theta|x_{0,b}}[p(x_k, d_k|\pi_w, \theta, x_0)]$, and the probability that $(x_k, d_k, y_k)$ takes place is $\mathbb{E}_{\theta|x_{0,b}}[p(x_k, d_k|\pi_w, \theta, x_0)p(y_k|\theta, x_k, d_k)]$. Thus, the probability that $(x_k, d_k, y_k)$ takes place conditioned on $(x_k, d_k)$ is $\frac{\mathbb{E}_{\theta|x_{0,b}}[p(x_k,d_k|\pi_w,\theta,x_0)p(y_k|\theta,x_k,d_k)]}{\mathbb{E}_{\theta|x_{0,b}}[p(x_k,d_k|\pi_w,\theta,x_0)]}$, and each $(x_k, d_k, y_k)$ contributes to $g_k(x_k, d_k, y_k) + Q^{\pi_w}_{k+1}(x_{k+1}, d_{k+1})$ as the target for the Q-network training at $(x_k, d_k)$. As Eqn. (2.21) employs the Mean Squared Error, in the limit of an infinite number of training episodes $M$ along with the unlimited representation capabilities of the DNN architecture, the value of Q-network at $(x_k, d_k)$ converges to the expectation of the target over $(x_k, d_k, y_k)$ conditioned on $(x_k, d_k)$, which is:

$$Q^{\pi_w}_v(k, x_k, d_k) = \frac{\mathbb{E}_{\theta|x_{0,b}}\left[p(x_k, d_k|\pi_w, \theta, x_0)\,\mathbb{E}_{y_k|\theta,x_k,d_k}\left[g_k(x_k, d_k, y_k) + Q^{\pi_w}_{k+1}(x_{k+1}, d_{k+1})\right]\right]}{\mathbb{E}_{\theta|x_{0,b}}[p(x_k, d_k|\pi_w, \theta, x_0)]} \tag{A.15}$$

$$\begin{aligned}
&= \frac{\mathbb{E}_{\theta|x_{0,b}}\left[p(x_k, d_k|\pi_w, \theta, x_0)\,\mathbb{E}_{y_k|\theta,x_k,d_k}\left[g_k(x_k, d_k, y_k) + Q^{\pi_w}_{k+1}(x_{k+1}, d_{k+1})\right]\right]}{p(x_k, d_k|\pi_w, x_0)} \\
&= \mathbb{E}_{\theta|x_k,d_k,\pi_w,x_0}\mathbb{E}_{y_k|\theta,x_k,d_k}\left[g_k(x_k, d_k, y_k) + Q^{\pi_w}_{k+1}(x_{k+1}, d_{k+1})\right] \\
&= \mathbb{E}_{\theta|x_k}\mathbb{E}_{y_k|\theta,x_k,d_k}\left[g_k(x_k, d_k, y_k) + Q^{\pi_w}_{k+1}(x_{k+1}, d_{k+1})\right] \\
&= \mathbb{E}_{y_k|x_k,d_k}\left[g_k(x_k, d_k, y_k) + Q^{\pi_w}_{k+1}(x_{k+1}, d_{k+1})\right]
\end{aligned} \tag{A.16}$$

where $d_{k+1} = \mu_{k+1,w_{k+1}}(x_{k+1})$. The third equality applies Bayes' rule, and the fourth equality follows because $\theta$ depends only on $x_k$ when $x_k$ is given. Eqn. (A.16) is identical to Eqn. (2.14), therefore showing the Q-network converges to the true Q-function.

# APPENDIX B

# Appendix of variational sequential optimal experimental design (vsOED)

## B.1 Information gain jointly with model probability

Akin to the total entropy described in [17], the information gain (IG) jointly on the model probability and model parameters of interest (PoIs) is:

$$
\begin{aligned}
&D_{\mathrm{KL}}\left(p(m,\theta_m|I_{k_2}) \,||\, p(m,\theta_m|I_{k_1})\right) \\
&= \sum_{m=1}^{\mathcal{M}} \int_{\Theta} p(m,\theta_m|I_{k_2}) \ln \frac{p(m,\theta_m|I_{k_2})}{p(m,\theta_m|I_{k_1})}\, d\theta_m \\
&= \sum_{m=1}^{\mathcal{M}} P(m|I_{k_2}) \int_{\Theta} p(\theta_m|I_{k_2}) \ln \frac{P(m|I_{k_2})\, p(\theta_m|I_{k_2})}{P(m|I_{k_1})\, p(\theta_m|I_{k_1})}\, d\theta_m \\
&= \sum_{m=1}^{\mathcal{M}} P(m|I_{k_2}) \ln \frac{P(m|I_{k_2})}{P(m|I_{k_1})} + \sum_{m=1}^{\mathcal{M}} P(m|I_{k_2}) \int_{\Theta} p(\theta_m|I_{k_2}) \ln \frac{p(\theta_m|I_{k_2})}{p(\theta_m|I_{k_1})}\, d\theta_m \\
&= D_{\mathrm{KL}}\left(P(m|I_{k_2}) \,||\, P(m|I_{k_1})\right) + \mathbb{E}_{m|I_{k_2}}\left[D_{\mathrm{KL}}\left(p(\theta_m|I_{k_2}) \,||\, p(\theta_m|I_{k_1})\right)\right],
\end{aligned}
$$

where $0 \le k_1 \le k_2 \le N$. Note that we use the convention where when $m$ is not explicitly mentioned, conditioning on $m$ is implied through other variables' subscripts, e.g., $p(\theta_m|I_k) = p(\theta_m|m, I_k)$. When setting $k_1 = 0$ and $k_2 = N$, we recover the terminal reward in Eqn. (3.7) under the special case of $\alpha_{\mathcal{M}} = \alpha_{\Theta} = 1$ and $\alpha_Z = 0$.

Similarly, the IG jointly on the model probability and predictive quantities of interest (QoIs) is:

$$
\begin{aligned}
&D_{\mathrm{KL}}\left(p(m, z_m|I_{k_2}) \,||\, p(m, z_m|I_{k_1})\right) \\
&= \sum_{m=1}^{\mathcal{M}} \int_Z p(m, z_m|I_{k_2}) \ln \frac{p(m, z_m|I_{k_2})}{p(m, z_m|I_{k_1})}\, dz_m \\
&= \sum_{m=1}^{\mathcal{M}} P(m|I_{k_2}) \int_Z p(z_m|I_{k_2}) \ln \frac{P(m|I_{k_2})\, p(z_m|I_{k_2})}{P(m|I_{k_1})\, p(z_m|I_{k_1})}\, dz_m \\
&= \sum_{m=1}^{\mathcal{M}} P(m|I_{k_2}) \frac{P(m|I_{k_2})}{P(m|I_{k_1})} + \sum_{m=1}^{\mathcal{M}} P(m|I_{k_2}) \int_Z p(z_m|I_{k_2}) \ln \frac{p(z_m|I_{k_2})}{p(z_m|I_{k_1})}\, dz_m \\
&= D_{\mathrm{KL}}\left(P(m|I_{k_2}) \,||\, P(m|I_{k_1})\right) + \mathbb{E}_{m|I_{k_2}}\left[D_{\mathrm{KL}}\left(p(z_m|I_{k_2}) \,||\, p(z_m|I_{k_1})\right)\right],
\end{aligned}
$$

where $0 \le k_1 \le k_2 \le N$. When setting $k_1 = 0$ and $k_2 = N$, we recover the terminal reward in Eqn. (3.7) under the special case of $\alpha_{\mathcal{M}} = \alpha_Z = 1$ and $\alpha_\Theta = 0$.

## B.2 Information gain jointly on model parameters and predictive quantities

When the nuisance parameters $\eta_m$ are absent, the IG jointly on the PoIs and QoIs given model $m$ is:

$$
\begin{aligned}
&D_{\mathrm{KL}}\left(p(\theta_m, z_m|I_{k_2}) \,||\, p(\theta_m, z_m|I_{k_1})\right) \\
&= \int_{\Theta,Z} p(\theta_m, z_m|I_{k_2}) \ln \frac{p(\theta_m, z_m|I_{k_2})}{p(\theta_m, z_m|I_{k_1})}\, dz_m\, d\theta_m \\
&= \int_{\Theta,Z} p(\theta_m, z_m|I_{k_2}) \ln \frac{p(\theta_m|I_{k_2})\, p(z_m|\theta_m, I_{k_2})}{p(\theta_m|I_{k_1})\, p(z_m|\theta_m, I_{k_1})}\, dz_m\, d\theta_m \\
&= \int_{\Theta,Z} p(\theta_m, z_m|I_{k_2}) \ln \frac{p(\theta_m|I_{k_2})\, p(z_m|\theta_m)}{p(\theta_m|I_{k_1})\, p(z_m|\theta_m)}\, dz_m\, d\theta_m \\
&= \int_{\Theta,Z} p(\theta_m, z_m|I_{k_2}) \ln \frac{p(\theta_m|I_{k_2})}{p(\theta_m|I_{k_1})}\, dz_m\, d\theta_m \\
&= \int_{\Theta} p(\theta_m|I_{k_2}) \ln \frac{p(\theta_m|I_{k_2})}{p(\theta_m|I_{k_1})}\, d\theta_m \\
&= D_{\mathrm{KL}}\left(p(\theta_m|I_{k_2}) \,||\, p(\theta_m|I_{k_1})\right),
\end{aligned}
$$

where the third equality is due to $z_m$ only dependent on $\theta_m$ when $\eta_m$ is absent (see Eqn. (3.4)). Hence, the IG on the QoIs is fully absorbed into the IG on the PoIs when nuisance parameters are absent.

## B.3 Proof of Theorem 3 (terminal-incremental equivalence)

*Proof.* We first decompose $U_T(\pi)$ into four additive parts:

$$U_T(\pi) = U_T(\pi; \text{non-IG}) + U_T(\pi; \alpha_{\mathcal{M}}) + U_T(\pi; \alpha_{\Theta}) + U_T(\pi; \alpha_Z),$$

where $U_T(\pi; \text{non-IG})$ captures any non-IG reward contributions, and the other three parts are (while explicitly writing out $I_0$)

$$\begin{aligned}
U_T(\pi; \alpha_{\mathcal{M}}) &= \alpha_{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0} \left[ D_{\mathrm{KL}} \left( P(m|I_N) \,||\, P(m|I_0) \right) \right] \\
U_T(\pi; \alpha_{\Theta}) &= \alpha_{\Theta} \mathbb{E}_{I_N|\pi,I_0} \mathbb{E}_{m|I_N} \left[ D_{\mathrm{KL}} \left( p(\theta_m|I_N) \,||\, p(\theta_m|I_0) \right) \right] \\
U_T(\pi; \alpha_Z) &= \alpha_Z \mathbb{E}_{I_N|\pi,I_0} \mathbb{E}_{m|I_N} \left[ D_{\mathrm{KL}} \left( p(z_m|I_N) \,||\, p(z_m|I_0) \right) \right].
\end{aligned}$$

Similarly, $U_I(\pi)$ can also be decomposed into four additive parts:

$$U_I(\pi) = U_I(\pi; \text{non-IG}) + U_I(\pi; \alpha_{\mathcal{M}}) + U_I(\pi; \alpha_{\Theta}) + U_I(\pi; \alpha_Z),$$

where $U_I(\pi; \text{non-IG})$ captures any non-IG reward contributions, and the other three parts are (while explicitly writing out $I_0$)

$$\begin{aligned}
U_I(\pi; \alpha_{\mathcal{M}}) &= \alpha_{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0} \left[ \sum_{k=0}^{N-1} D_{\mathrm{KL}} \left( P(m|I_{k+1}) \,||\, P(m|I_k) \right) \right] \\
U_I(\pi; \alpha_{\Theta}) &= \alpha_{\Theta} \mathbb{E}_{I_N|\pi,I_0} \sum_{k=0}^{N-1} \mathbb{E}_{m|I_{k+1}} \left[ D_{\mathrm{KL}} \left( p(\theta_m|I_{k+1}) \,||\, p(\theta_m|I_k) \right) \right] \\
U_I(\pi; \alpha_Z) &= \alpha_Z \mathbb{E}_{I_N|\pi,I_0} \sum_{k=0}^{N-1} \mathbb{E}_{m|I_{k+1}} \left[ D_{\mathrm{KL}} \left( p(z_m|I_{k+1}) \,||\, p(z_m|I_k) \right) \right].
\end{aligned}$$

Since TIG and IIG formulations only entail the IG contributions, the non-IG reward contributions are therefore not affected by this choice and hence

$$U_T(\pi; \text{non-IG}) = U_I(\pi; \text{non-IG}).$$

For the part corresponding to IG on model probability:

$$
\begin{aligned}
&U_I(\pi;\alpha_{\mathcal{M}}) - U_T(\pi;\alpha_{\mathcal{M}}) \\
&= \alpha_{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0}\left[\sum_{k=0}^{N-1} D_{\mathrm{KL}}\left(P(m|I_{k+1}) \,||\, P(m|I_k)\right) - D_{\mathrm{KL}}\left(P(m|I_N) \,||\, P(m|I_0)\right)\right] \\
&= \alpha_{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0}\left[\sum_{k=0}^{N-1}\sum_{m=1}^{\mathcal{M}} P(m|I_{k+1}) \ln \frac{P(m|I_{k+1})}{P(m|I_k)} - \sum_{m=1}^{\mathcal{M}} P(m|I_N) \ln \frac{P(m|I_N)}{P(m|I_0)}\right] \\
&= \alpha_{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0}\left[\sum_{k=0}^{N-1} P(m|I_{k+1}) \ln \frac{P(m|I_{k+1})}{P(m|I_k)} - P(m|I_N) \ln \frac{P(m|I_N)}{P(m|I_0)}\right] \\
&= \alpha_{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0}\left[\sum_{k=0}^{N-2} P(m|I_{k+1}) \ln \frac{P(m|I_{k+1})}{P(m|I_k)} + P(m|I_N) \ln \frac{P(m|I_N)}{P(m|I_{N-1})}\right. \\
&\qquad\qquad \left. - P(m|I_N) \ln \frac{P(m|I_N)}{P(m|I_0)}\right] \\
&= \alpha_{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0}\left[\sum_{k=0}^{N-2} P(m|I_{k+1}) \ln \frac{P(m|I_{k+1})}{P(m|I_k)} - P(m|I_N) \ln \frac{P(m|I_{N-1})}{P(m|I_0)}\right] \\
&= \alpha_{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_{N-1}|\pi,I_0}\left[\sum_{k=0}^{N-2} P(m|I_{k+1}) \ln \frac{P(m|I_{k+1})}{P(m|I_k)} - \mathbb{E}_{I_N|\pi,I_{N-1}} P(m|I_N) \ln \frac{P(m|I_{N-1})}{P(m|I_0)}\right] \\
&= \alpha_{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_{N-1}|\pi,I_0}\left[\sum_{k=0}^{N-2} P(m|I_{k+1}) \ln \frac{P(m|I_{k+1})}{P(m|I_k)} - P(m|I_{N-1}) \ln \frac{P(m|I_{N-1})}{P(m|I_0)}\right] \\
&\quad \vdots \\
&= \alpha_{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_1|\pi,I_0}\left[\sum_{k=0}^{0} P(m|I_{k+1}) \ln \frac{P(m|I_{k+1})}{P(m|I_k)} - P(m|I_1) \ln \frac{P(m|I_1)}{P(m|I_0)}\right] \\
&= 0,
\end{aligned}
$$

where the seventh equality is due to

$$
\begin{aligned}
\mathbb{E}_{I_N|\pi,I_{N-1}} P(m|I_N) \ln \frac{P(m|I_{N-1})}{P(m|I_0)} &= \int_{\mathcal{Y}} p(y_{N-1}|\pi, I_{N-1}) P(m|I_N) \ln \frac{P(m|I_{N-1})}{P(m|I_0)}\, dy_{N-1} \\
&= \int_{\mathcal{Y}} p(y_{N-1}, m|\pi, I_{N-1}) \ln \frac{P(m|I_{N-1})}{P(m|I_0)}\, dy_{N-1} \\
&= P(m|I_{N-1}) \ln \frac{P(m|I_{N-1})}{P(m|I_0)}
\end{aligned}
$$

with $P(m|I_N) = P(m|I_{N-1}, \pi, y_{N-1})$ since the policy is deterministic, and the eighth equality results from repeatedly applying the steps between the third and seventh equalities until $N = 1$.

For the part corresponding to IG on the PoIs:

$$
\begin{aligned}
&U_I(\pi;\alpha_\Theta) - U_T(\pi;\alpha_\Theta) \\
&= \alpha_\Theta \mathbb{E}_{I_N|\pi,I_0}\left[\sum_{k=0}^{N-1} \mathbb{E}_{m|I_{k+1}} D_{\mathrm{KL}}\left(\, p(\theta_m|I_{k+1}) \,||\, p(\theta_m|I_k)\,\right) - \mathbb{E}_{m|I_N} D_{\mathrm{KL}}\left(\, p(\theta_m|I_N) \,||\, p(\theta_m|I_0)\,\right)\right] \\
&= \alpha_\Theta \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0}\left[\sum_{k=0}^{N-1} P(m|I_{k+1}) D_{\mathrm{KL}}\left(\, p(\theta_m|I_{k+1}) \,||\, p(\theta_m|I_k)\,\right)\right. \\
&\qquad\qquad \left. - P(m|I_N) D_{\mathrm{KL}}\left(\, p(\theta_m|I_N) \,||\, p(\theta_m|I_0)\,\right)\right] \\
&= \alpha_\Theta \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0}\left[\sum_{k=0}^{N-2} P(m|I_{k+1}) D_{\mathrm{KL}}\left(\, p(\theta_m|I_{k+1}) \,||\, p(\theta_m|I_k)\,\right)\right. \\
&\qquad\qquad + P(m|I_N) D_{\mathrm{KL}}\left(\, p(\theta_m|I_N) \,||\, p(\theta_m|I_{N-1})\,\right) \\
&\qquad\qquad \left. - P(m|I_N) D_{\mathrm{KL}}\left(\, p(\theta_m|I_N) \,||\, p(\theta_m|I_0)\,\right)\right] \\
&= \alpha_\Theta \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0}\left[\sum_{k=0}^{N-2} P(m|I_{k+1}) D_{\mathrm{KL}}\left(\, p(\theta_m|I_{k+1}) \,||\, p(\theta_m|I_k)\,\right)\right. \\
&\qquad\qquad \left. + P(m|I_N) \int_\Theta p(\theta_m|I_N) \ln \frac{p(\theta_m|I_0)}{p(\theta_m|I_{N-1})}\, d\theta_m\right] \\
&= \alpha_\Theta \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_{N-1}|\pi,I_0}\left[\sum_{k=0}^{N-2} P(m|I_{k+1}) D_{\mathrm{KL}}\left(\, p(\theta_m|I_{k+1}) \,||\, p(\theta_m|I_k)\,\right)\right. \\
&\qquad\qquad \left. + \mathbb{E}_{I_N|\pi,I_{N-1}} P(m|I_N) \int_\Theta p(\theta_m|I_N) \ln \frac{p(\theta_m|I_0)}{p(\theta_m|I_{N-1})}\, d\theta_m\right] \\
&= \alpha_\Theta \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_{N-1}|\pi,I_0}\left[\sum_{k=0}^{N-2} P(m|I_{k+1}) D_{\mathrm{KL}}\left(\, p(\theta_m|I_{k+1}) \,||\, p(\theta_m|I_k)\,\right)\right. \\
&\qquad\qquad \left. - P(m|I_{N-1}) D_{\mathrm{KL}}\left(\, p(\theta_m|I_{N-1}) \,||\, p(\theta_m|I_0)\,\right)\right] \\
&\quad \vdots \\
&= \alpha_\Theta \sum_{m=1}^{\mathcal{M}} \mathbb{E}_{I_1|\pi,I_0}\left[\sum_{k=0}^{0} P(m|I_{k+1}) D_{\mathrm{KL}}\left(\, p(\theta_m|I_{k+1}) \,||\, p(\theta_m|I_k)\,\right)\right. \\
&\qquad\qquad \left. - P(m|I_1) D_{\mathrm{KL}}\left(\, p(\theta_m|I_1) \,||\, p(\theta_m|I_0)\,\right)\right] \\
&= 0,
\end{aligned}
$$

where the fourth equality is due to

$$
\begin{aligned}
&P(m|I_N)D_{\mathrm{KL}}\left(\, p(\theta_m|I_N) \,||\, p(\theta_m|I_{N-1}) \,\right) - P(m|I_N)D_{\mathrm{KL}}\left(\, p(\theta_m|I_N) \,||\, p(\theta_m|I_0) \,\right) \\
&= P(m|I_N) \int_\Theta p(\theta_m|I_N) \left[ \ln \frac{p(\theta_m|I_N)}{p(\theta_m|I_{N-1})} - \ln \frac{p(\theta_m|I_N)}{p(\theta_m|I_0)} \right] d\theta_m \\
&= P(m|I_N) \int_\Theta p(\theta_m|I_N) \ln \frac{p(\theta_m|I_0)}{p(\theta_m|I_{N-1})} \, d\theta_m,
\end{aligned}
$$

and the sixth equality is due to

$$
\begin{aligned}
&\mathbb{E}_{I_N|\pi,I_{N-1}} P(m|I_N) \int_\Theta p(\theta_m|I_N) \ln \frac{p(\theta_m|I_0)}{p(\theta_m|I_{N-1})} \, d\theta_m \\
&= \int_{\mathcal{Y}} P(y_{N-1}|\pi, I_{N-1}) P(m|I_N) \int_\Theta p(\theta_m|I_N) \ln \frac{p(\theta_m|I_0)}{p(\theta_m|I_{N-1})} \, d\theta_m \, dy_{N-1} \\
&= \int_{\mathcal{Y}} \int_\Theta P(y_{N-1}, m, \theta_m|\pi, I_{N-1}) \ln \frac{p(\theta_m|I_0)}{p(\theta_m|I_{N-1})} \, d\theta_m \, dy_{N-1} \\
&= P(m|I_{N-1}) \int_\Theta P(\theta_m|I_{N-1}) \ln \frac{p(\theta_m|I_0)}{p(\theta_m|I_{N-1})} \, d\theta_m \, dy_{N-1} \\
&= -P(m|I_{N-1})D_{\mathrm{KL}}\left(\, p(\theta_m|I_{N-1}) \,||\, p(\theta_m|I_0) \,\right),
\end{aligned}
$$

and the seventh equality results from repeatedly applying the steps between the second and sixth equalities until $N = 1$.

For the part corresponding to IG on the QoIs, the derivation is identical as above for the PoIs except replacing $\theta_m$ with $z_m$, to arrive at

$$U_I(\pi; \alpha_Z) - U_T(\pi; \alpha_Z) = 0.$$

Combining the equivalence results from the four parts, we obtain

$$U_I(\pi) = U_T(\pi)$$

for any policy $\pi$. □

## B.4 Proof of Theorem 4 (one-point-estimate equivalence)

*Proof.* We begin by proving the equivalence of expected utility under the TIG and one-point-TIG:

$$
\begin{aligned}
U_T(\pi) &= \mathbb{E}_{I_N|\pi,I_0}\left[\sum_{k=0}^{N-1} g_k(I_k, d_k, y_k) + g_N(I_N)\right] \\
&= \mathbb{E}_{I_N|\pi,I_0}\Bigg[\alpha_{\mathcal{M}} D_{\mathrm{KL}}\left(\, P(m|I_N) \,||\, P(m) \,\right) \\
&\qquad + \mathbb{E}_{m|I_N}\left[\alpha_\Theta D_{\mathrm{KL}}\left(\, p(\theta_m|I_N) \,||\, p(\theta_m) \,\right) + \alpha_Z D_{\mathrm{KL}}\left(\, p(z_m|I_N) \,||\, p(z_m) \,\right)\right]\Bigg] \\
&= \mathbb{E}_{I_N|\pi,I_0}\Bigg[\alpha_{\mathcal{M}} \mathbb{E}_{m|I_N} \ln \frac{P(m|I_N)}{P(m)} \\
&\qquad + \mathbb{E}_{m|I_N}\left[\alpha_\Theta \mathbb{E}_{\theta_m|I_N} \ln \frac{p(\theta_m|I_N)}{p(\theta_m)} + \alpha_Z \mathbb{E}_{z_m|I_N} \ln \frac{p(z_m|I_N)}{p(z_m)}\right]\Bigg] \\
&= \mathbb{E}_{m,I_N|\pi,I_0}\Bigg[\alpha_{\mathcal{M}} \ln \frac{P(m|I_N)}{P(m)} \\
&\qquad + \alpha_\Theta \mathbb{E}_{\theta_m|I_N} \ln \frac{p(\theta_m|I_N)}{p(\theta_m)} + \alpha_Z \mathbb{E}_{z_m|I_N} \ln \frac{p(z_m|I_N)}{p(z_m)}\Bigg] \\
&= \mathbb{E}_{m,I_N|\pi,I_0}\Bigg[\alpha_{\mathcal{M}} \mathbb{E}_{\theta_m,z_m|I_N} \ln \frac{P(m|I_N)}{P(m)} \\
&\qquad + \alpha_\Theta \mathbb{E}_{\theta_m,z_m|I_N} \ln \frac{p(\theta_m|I_N)}{p(\theta_m)} + \alpha_Z \mathbb{E}_{\theta_m,z_m|I_N} \ln \frac{p(z_m|I_N)}{p(z_m)}\Bigg] \\
&= \mathbb{E}_{m,\theta_m,z_m,I_N|\pi,I_0}\left[\alpha_{\mathcal{M}} \ln \frac{P(m|I_N)}{P(m)} + \alpha_\Theta \ln \frac{p(\theta_m|I_N)}{p(\theta_m)} + \alpha_Z \ln \frac{p(z_m|I_N)}{p(z_m)}\right] \\
&= \mathbb{E}_{m,\theta_m,\eta_m,z_m,I_N|\pi,I_0}\left[\alpha_{\mathcal{M}} \ln \frac{P(m|I_N)}{P(m)} + \alpha_\Theta \ln \frac{p(\theta_m|I_N)}{p(\theta_m)} + \alpha_Z \ln \frac{p(z_m|I_N)}{p(z_m)}\right] \\
&= \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \mathbb{E}_{I_N|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m}\left[\alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_N)}{P(\dot{m})} + \alpha_\Theta \ln \frac{p(\dot{\theta}_m|I_N)}{p(\dot{\theta}_m)} + \alpha_Z \ln \frac{p(\dot{z}_m|I_N)}{p(\dot{z}_m)}\right] \\
&= \dot{U}_T(\pi).
\end{aligned}
$$

Next, we have already established the equivalence $U_T(\pi) = U_I(\pi)$ in Appendix B.3. Finally, we

show the equivalence between $\dot{U}_I(\pi)$ and $\dot{U}_T(\pi)$ by cancelling out all intermediate posteriors:

$$\begin{aligned}
&\dot{U}_I(\pi) \\
&= \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \mathbb{E}_{I_N|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \sum_{k=0}^{N-1} \left[ \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_{k+1})}{P(\dot{m}|I_k)} + \alpha_{\Theta} \ln \frac{p(\dot{\theta}_m|I_{k+1})}{p(\dot{\theta}_m|I_k)} + \alpha_Z \ln \frac{p(\dot{z}_m|I_{k+1})}{p(\dot{z}_m|I_k)} \right] && \text{(B.1)} \\
&= \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \mathbb{E}_{I_N|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \left[ \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_N)}{P(\dot{m})} + \alpha_{\Theta} \ln \frac{p(\dot{\theta}_m|I_N)}{p(\dot{\theta}_m)} + \alpha_Z \ln \frac{p(\dot{z}_m|I_N)}{p(\dot{z}_m)} \right] && \text{(B.2)} \\
&= \dot{U}_T(\pi).
\end{aligned}$$

Combining the above equivalence results together, we have

$$U_T(\pi) = \dot{U}_T(\pi) = \dot{U}_I(\pi) = U_I(\pi)$$

for any policy $\pi$. □

## B.5 Omitting prior terms

The difference between the expected utilities under the one-point IG estimate formulations that omit and include the prior term is:

$$\begin{aligned}
&\delta\dot{U}(\pi) \\
&= \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \mathbb{E}_{I_N|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \left[ \alpha_{\mathcal{M}} \ln P(\dot{m}) + \alpha_{\Theta} \ln p(\dot{\theta}_m) + \alpha_Z \ln p(\dot{z}_m) \right] \\
&= \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \left[ \alpha_{\mathcal{M}} \ln P(\dot{m}) + \alpha_{\Theta} \ln p(\dot{\theta}_m) + \alpha_Z \ln p(\dot{z}_m) \right],
\end{aligned}$$

which is constant with respect to the policy $\pi$. Therefore, whether including or omitting the prior terms will not affect the optimal policy (i.e. the arg-max to the expected utilities). The same conclusion can be drawn for the expected utilities under the full IG formulations.

## B.6 Proof of Theorem 5 (variational lower bound)

*Proof.* Appendix B.7 shows that the expected utilities using the variational-one-point-TIG and variational-one-point-IIG are equivalent. Thus, below we prove the lower bound under the variational-one-point-TIG, and the same result carries over to the variational-one-point-IIG due to their equivalence.

The difference between the expected utility and the variational expected utility is

$$
\begin{aligned}
&\dot{U}(\pi) - \dot{U}(\pi; \phi) \\
&= \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \mathbb{E}_{I_N|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \left[ \alpha_{\mathcal{M}} \ln \frac{P(\dot{m}|I_N)}{q(\dot{m}|I_N;\phi_{\mathcal{M}})} + \alpha_{\Theta} \ln \frac{p(\dot{\theta}_m|I_N)}{q(\dot{\theta}_m|I_N;\phi_{\Theta_m})} \right. \\
&\qquad\qquad \left. + \alpha_Z \ln \frac{p(\dot{z}_m|I_N)}{q(\dot{z}_m|I_N;\phi_{Z_m})} \right] \\
&= \alpha_{\mathcal{M}} \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m,I_N|\pi,I_0} \left[ \ln \frac{P(\dot{m}|I_N)}{q(\dot{m}|I_N;\phi_{\mathcal{M}})} \right] \\
&\quad + \alpha_{\Theta} \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m,I_N|\pi,I_0} \left[ \ln \frac{p(\dot{\theta}_m|I_N)}{q(\dot{\theta}_m|I_N;\phi_{\Theta_m})} \right] \\
&\quad + \alpha_Z \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m,I_N|\pi,I_0} \left[ \ln \frac{p(\dot{z}_m|I_N)}{q(\dot{z}_m|I_N;\phi_{Z_m})} \right] \\
&= \alpha_{\mathcal{M}} \mathbb{E}_{\dot{m},I_N|\pi,I_0} \left[ \ln \frac{P(\dot{m}|I_N)}{q(\dot{m}|I_N;\phi_{\mathcal{M}})} \right] \\
&\quad + \alpha_{\Theta} \mathbb{E}_{\dot{m},\dot{\theta}_m,I_N|\pi,I_0} \left[ \ln \frac{p(\dot{\theta}_m|I_N)}{q(\dot{\theta}_m|I_N;\phi_{\Theta_m})} \right] \\
&\quad + \alpha_Z \mathbb{E}_{\dot{m},\dot{z}_m,I_N|\pi,I_0} \left[ \ln \frac{p(\dot{z}_m|I_N)}{q(\dot{z}_m|I_N;\phi_{Z_m})} \right] \\
&= \alpha_{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0} \mathbb{E}_{\dot{m}|I_N} \left[ \ln \frac{P(\dot{m}|I_N)}{q(\dot{m}|I_N;\phi_{\mathcal{M}})} \right] \\
&\quad + \alpha_{\Theta} \mathbb{E}_{\dot{m},I_N|\pi,I_0} \mathbb{E}_{\dot{\theta}_m|\dot{m},I_N} \left[ \ln \frac{p(\dot{\theta}_m|I_N)}{q(\dot{\theta}_m|I_N;\phi_{\Theta_m})} \right] \\
&\quad + \alpha_Z \mathbb{E}_{\dot{m},I_N|\pi,I_0} \mathbb{E}_{\dot{z}_m|\dot{m},I_N} \left[ \ln \frac{p(\dot{z}_m|I_N)}{q(\dot{z}_m|I_N;\phi_{Z_m})} \right] \\
&= \alpha_{\mathcal{M}} \mathbb{E}_{I_N|\pi,I_0} \left[ D_{\mathrm{KL}} \left( P(\dot{m}|I_N) \,||\, q(\dot{m}|I_N;\phi_{\mathcal{M}}) \right) \right] \\
&\quad + \alpha_{\Theta} \mathbb{E}_{\dot{m},I_N|\pi,I_0} \left[ D_{\mathrm{KL}} \left( p(\dot{\theta}_m|I_N) \,||\, q(\dot{\theta}_m|I_N;\phi_{\Theta_m}) \right) \right] \\
&\quad + \alpha_Z \mathbb{E}_{\dot{m},I_N|\pi,I_0} \left[ D_{\mathrm{KL}} \left( p(\dot{z}_m|I_N) \,||\, q(\dot{z}_m|I_N;\phi_{Z_m}) \right) \right] \\
&\geq 0
\end{aligned}
$$

where $\alpha_{\mathcal{M}} \geq 0$, $\alpha_{\Theta} \geq 0$, $\alpha_Z \geq 0$. The sixth equality is due to $p(\dot{\theta}_m|I_N)$ being equivalent to $p(\dot{\theta}_m|\dot{m}, I_N)$ and $p(\dot{z}_m|I_N)$ being equivalent to $p(\dot{z}_m|\dot{m}, I_N)$, due to the notation convention adopted in this paper. The bound is tight if and only if $q(\cdot|I_N; \phi_{(\cdot)}) = p(\cdot|I_N)$ (except the trivial case when $\alpha_{\mathcal{M}} = \alpha_{\Theta} = \alpha_Z = 0$).

□

## B.7 Cancellation of intermediate posteriors

Similar to Eqn. (B.1) and Eqn. (B.2), all intermediate variational posteriors $q(\cdot|I_k; \phi_{(\cdot)})$ for $k = 1, \ldots, N-1$ cancel out:

$$
\begin{aligned}
&\dot{U}_I(\pi; \phi) \\
&= \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \mathbb{E}_{I_N|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \sum_{k=0}^{N-1} \left[ \alpha_{\mathcal{M}} \ln \frac{q(\dot{m}|I_{k+1}; \phi_{\mathcal{M}})}{q(\dot{m}|I_k; \phi_{\mathcal{M}})} \right. \\
&\qquad \left. + \alpha_{\Theta} \ln \frac{q(\dot{\theta}_m|I_{k+1}; \phi_{\Theta_m})}{q(\dot{\theta}_m|I_k; \phi_{\Theta_m})} + \alpha_Z \ln \frac{q(\dot{z}_m|I_{k+1}; \phi_{Z_m})}{q(\dot{z}_m|I_k; \phi_{Z_m})} \right] \\
&= \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \mathbb{E}_{I_N|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \Bigg[ \\
&\qquad \alpha_{\mathcal{M}} \left( \ln \frac{q(\dot{m}|I_1; \phi_{\mathcal{M}})}{P(\dot{m})} + \ln \frac{q(\dot{m}|I_2; \phi_{\mathcal{M}})}{q(\dot{m}|I_1; \phi_{\mathcal{M}})} + \cdots + \ln \frac{q(\dot{m}|I_N; \phi_{\mathcal{M}})}{q(\dot{m}|I_{N-1}; \phi_{\mathcal{M}})} \right) \\
&\qquad + \alpha_{\Theta} \left( \ln \frac{q(\dot{\theta}_m|I_1; \phi_{\Theta_m})}{p(\dot{\theta}_m)} + \ln \frac{q(\dot{\theta}_m|I_2; \phi_{\Theta_m})}{q(\dot{\theta}_m|I_1; \phi_{\Theta_m})} + \cdots + \ln \frac{q(\dot{\theta}_m|I_N; \phi_{\Theta_m})}{q(\dot{\theta}_m|I_{N-1}; \phi_{\Theta_m})} \right) \\
&\qquad + \alpha_Z \left( \ln \frac{q(\dot{z}_m|I_1; \phi_{Z_m})}{p(\dot{z}_m)} + \ln \frac{q(\dot{z}_m|I_2; \phi_{Z_m})}{q(\dot{z}_m|I_1; \phi_{Z_m})} + \cdots + \ln \frac{q(\dot{z}_m|I_N; \phi_{Z_m})}{q(\dot{z}_m|I_{N-1}; \phi_{Z_m})} \right) \Bigg] \\
&= \mathbb{E}_{\dot{m},\dot{\theta}_m,\dot{\eta}_m,\dot{z}_m} \mathbb{E}_{I_N|\pi,I_0,\dot{m},\dot{\theta}_m,\dot{\eta}_m} \left[ \alpha_{\mathcal{M}} \ln \frac{q(\dot{m}|I_N; \phi_{\mathcal{M}})}{P(\dot{m})} \right. \\
&\qquad \left. + \alpha_{\Theta} \ln \frac{q(\dot{\theta}_m|I_N; \phi_{\Theta_m})}{p(\dot{\theta}_m)} + \alpha_Z \ln \frac{q(\dot{z}_m|I_N; \phi_{Z_m})}{p(\dot{z}_m)} \right] \\
&= \dot{U}_T(\pi; \phi).
\end{aligned}
$$

Therefore, only the prior and the final variational posterior terms contribute to the variational expected utility. Since the prior PDF is either known or omitted, the accuracy of the variational expected utility only depends on the quality of the final variational posterior approximation.

# APPENDIX C

# Appendix of robust optimal experimental design (rOED)

In this appendix, all the expectations and variances are conditioned on the design $d$. However, we will omit this conditioning for simplicity with the understanding that it is always implied.

## C.1 Variance and bias of $\hat{U}^{N,M_1}(d)^2$

The variance of $\hat{U}^{N,M_1}(d)$ can be estimated using Taylor expansions for the moments of functions of random variables:

$$\begin{aligned}
\mathbb{V}\left[\hat{U}^{N,M_1}(d)^2\right] &\approx \left\{2\mathbb{E}\left[\hat{U}^{N,M_1}(d)\right]\right\}^2 \mathbb{V}\left[\hat{U}^{N,M_1}(d)\right] \\
&\approx 4\left[U(d) + \frac{E_1(d)}{M_1}\right]^2 \left[\frac{A_1(d)}{N} + \frac{B_1(d)}{NM_1}\right] \\
&\approx \frac{A_2(d)}{N} + \frac{B_2(d)}{NM_1}.
\end{aligned}$$

The bias of $\hat{U}^{N,M_1}(d)$ is

$$\begin{aligned}
\mathbb{E}\left[\hat{U}^{N,M_1}(d)^2 - U(d)^2\right] &= \mathbb{E}\left[(\hat{U}^{N,M_1}(d) - U(d))^2 - 2U(d)^2 + 2\hat{U}^{N,M_1}(d)U(d)\right] \\
&= \mathbb{V}\left[\hat{U}^{N,M_1}(d)\right] - 2U(d)^2 + 2U(d)\mathbb{E}\left[\hat{U}^{N,M_1}(d)\right] \\
&\approx \frac{A_1(d)}{N} + \frac{B_1(d)}{NM_1} + 2U(d)\mathbb{E}\left[\hat{U}^{N,M_1}(d) - U(d)\right] \\
&\approx \frac{A_1(d)}{N} + \frac{B_1(d)}{NM_1} + 2U(d)\frac{E_1(d)}{M_1} \\
&\approx \frac{D_2(d)}{N} + \frac{E_2(d)}{M_1},
\end{aligned}$$

note that the $\frac{1}{NM_1}$ term has been discarded in the last equality.

## C.2 Variance and bias of $\hat{\tilde{U}}^{N,M_1}_{\mu_2,1}(d)$

The variance of $\hat{\tilde{U}}^{N,M_1}_{\mu_2,1}(d)$ can be decomposed as

$$\begin{aligned}
\mathbb{V}\left[\hat{\tilde{U}}^{N,M_1}_{\mu_2,1}(d)\right] &= \frac{1}{N}\mathbb{V}\left\{\left[\ln\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]^2\right\} \\
&= \frac{1}{N}\mathbb{V}\mathbb{E}\left\{\left[\ln\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]^2\middle|y\right\} \\
&\quad + \frac{1}{N}\mathbb{E}\mathbb{V}\left\{\left[\ln\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]^2\middle|y\right\}
\end{aligned} \tag{C.1}$$

where the first equality is due to the independence between $\left[\ln\frac{1}{M_1}\sum_{j=1}^{M_1} p(y^{(i_1)}|\theta^{(i_1,j)},d)\right]^2$ and $\left[\ln\frac{1}{M_1}\sum_{j=1}^{M_1} p(y^{(i_2)}|\theta^{(i_2,j)},d)\right]^2$ when $i_1 \neq i_2$, and the second equality is due to the law of total variance. It is obvious that

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\middle|y\right] &= p(y|d) \\
\mathbb{V}\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\middle|y\right] &= \frac{1}{M_1}\mathbb{V}\left[p(y|\theta^*,d)|y\right]
\end{aligned}$$

where $\theta^*$ represents the random variable $\theta^{(\cdot,j)}$, and the superscript is used to distinguish the inner $\theta^*$ from the outer random variable $\theta$ for $\theta^{(\cdot)}$. By applying Taylor expansions for the moments of function of random variables, we can get

$$\begin{aligned}
\mathbb{E}\left[\ln\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\middle|y\right] &\approx \ln p(y|d) - \frac{\mathbb{V}\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\middle|y\right]}{2p(y|d)^2} \\
&= \ln p(y|d) - \frac{\mathbb{V}\left[p(y|\theta^*,d)|y\right]}{2p(y|d)^2}\frac{1}{M_1},
\end{aligned} \tag{C.2}$$

$$\mathbb{V}\left[\ln \frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\middle| y\right] \approx \frac{1}{p(y|d)^2}\frac{1}{M_1}\mathbb{V}\left[p(y|\theta^*,d)|y\right]$$
$$= \frac{\mathbb{V}\left[p(y|\theta^*,d)|y\right]}{p(y|d)^2}\frac{1}{M_1}, \tag{C.3}$$

and

$$\mathbb{E}\left\{\left[\ln \frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]^2\middle| y\right\}$$
$$\approx \left\{\ln \mathbb{E}\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\middle| y\right]\right\}^2 + \frac{1-\ln p(y|d)}{p(y|d)^2}\mathbb{V}\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\middle| y\right]$$
$$= \left[\ln p(y|d)\right]^2 + \frac{1-\ln p(y|d)}{p(y|d)^2}\frac{1}{M_1}\mathbb{V}\left[p(y|\theta^*,d)|y\right], \tag{C.4}$$

as well as

$$\mathbb{V}\left\{\left[\ln \frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]^2\middle| y\right\} \approx \left[\frac{2\ln p(y|d)}{p(y|d)}\right]^2\frac{1}{M_1}\mathbb{V}\left[p(y|\theta^*,d)|y\right]. \tag{C.5}$$

By plugging Eqn. (C.4) and Eqn. (C.5) into Eqn. (C.1), we can get the variance of $\hat{\tilde{U}}_{\mu_2,1}^{N,M_1}(d)$ as

$$\mathbb{V}\left[\hat{\tilde{U}}_{\mu_2,1}^{N,M_1}(d)\right] \approx \frac{1}{N}\mathbb{V}\left\{\left[\ln p(y|d)\right]^2 + \frac{1-\ln p(y|d)}{p(y|d)^2}\frac{1}{M_1}\mathbb{V}\left[p(y|\theta^*,d)|y\right]\right\}$$
$$+ \frac{1}{NM_1}\mathbb{E}\left\{\left[\frac{2\ln p(y|d)}{p(y|d)}\right]^2\mathbb{V}\left[p(y|\theta^*,d)|y\right]\right\}$$
$$= \frac{A_3(d)}{N} + \frac{B_3(d)}{NM_1}$$

The bias of $\hat{\tilde{U}}^{N,M_1}_{\mu_2,1}(d)$ is

$$
\begin{aligned}
&\mathbb{E}\left[\hat{\tilde{U}}^{N,M_1}_{\mu_2,1}(d) - \tilde{U}_{\mu_2,1}(d)\right] \\
=&\mathbb{E}\left\{\left[\ln\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]^2\right\} - \mathbb{E}\left\{[\ln p(y|d)]^2\right\} \\
=&\mathbb{E}\mathbb{E}\left\{\left[\ln\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]^2\middle|y\right\} - \mathbb{E}\left\{[\ln p(y|d)]^2\right\} \\
\approx&\mathbb{E}\left\{[\ln p(y|d)]^2 + \frac{1-\ln p(y|d)}{p(y|d)^2}\frac{1}{M_1}\mathbb{V}\left[p(y|\theta^*,d)|y\right]\right\} - \mathbb{E}\left\{[\ln p(y|d)]^2\right\} \\
=&\frac{E_3(d)}{M_1}
\end{aligned}
$$

## C.3 Variance and bias of $\hat{\tilde{U}}^{N,M_1}_{\mu_2,2}(d)$

The variance of $\hat{\tilde{U}}^{N,M_1}_{\mu_2,2}(d)$ can be decomposed as

$$
\begin{aligned}
\mathbb{V}\left[\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d)\right] &= \frac{4}{N}\mathbb{V}\left\{\ln p(y|\theta,d)\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]\right\} \\
&= \frac{4}{N}\mathbb{V}\mathbb{E}\left\{\ln p(y|\theta,d)\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]\middle|\theta,y\right\} \\
&\quad + \frac{4}{N}\mathbb{E}\mathbb{V}\left\{\ln p(y|\theta,d)\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]\middle|\theta,y\right\}
\end{aligned} \tag{C.6}
$$

where $\theta$ stands for the random variable in the outer integral, and $\theta^{(\cdot,j)}$ are the samples in the inner loop. We can easily get that

$$
\begin{aligned}
&\mathbb{E}\left\{\ln p(y|\theta,d)\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]\middle|\theta,y\right\} \\
=&\ln p(y|\theta,d)\mathbb{E}\left\{\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\right]\middle|\theta,y\right\} \\
\approx&\ln p(y|\theta,d)\left[\ln p(y|d) - \frac{\mathbb{V}\left[p(y|\theta^*,d)|y\right]}{2p(y|d)^2}\frac{1}{M_1}\right]
\end{aligned}
$$

where the first equality is because $\theta$ and $y$ are given in this conditional expectation, thus $\ln p(y|\theta,d)$ can be pulled out of the expectation, and the second equality is simply using Eqn. (C.2). We can also get

$$\begin{aligned}
&\mathbb{V}\left\{\ln p(y|\theta,d)\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1}p(y|\theta^{(\cdot,j)},d)\right]\middle|\theta,y\right\}\\
=&\left[\ln p(y|\theta,d)\right]^2\mathbb{V}\left\{\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1}p(y|\theta^{(\cdot,j)},d)\right]\middle|\theta,y\right\}\\
\approx&\left[\ln p(y|\theta,d)\right]^2\frac{\mathbb{V}\left[p(y|\theta^*,d)|y\right]}{p(y|d)^2}\frac{1}{M_1}\\
=&\frac{\left[\ln p(y|\theta,d)\right]^2\mathbb{V}\left[p(y|\theta^*,d)\right]}{p(y|d)^2}\frac{1}{M_1}
\end{aligned}$$

where the second equality is using Eqn. (C.3). By plugging the above two equations into Eqn. (C.6), we can obtain the variance of $\hat{\tilde{U}}^{N,M_1}_{\mu_2,2}(d)$ as

$$\begin{aligned}
\mathbb{V}\left[\hat{\tilde{U}}^{N,M_1}_{\mu_2,2}(d)\right]\approx&\frac{4}{N}\mathbb{V}\left\{\ln p(y|\theta,d)\left[\ln p(y|d)-\frac{\mathbb{V}\left[p(y|\theta^*,d)|y\right]}{2p(y|d)^2}\frac{1}{M_1}\right]\right\}\\
&+\frac{4}{NM_1}\mathbb{E}\left[\frac{\left[\ln p(y|\theta,d)\right]^2\mathbb{V}\left[p(y|\theta^*,d)|y\right]}{p(y|d)^2}\right]\\
=&\frac{A_4(d)}{N}+\frac{B_4(d)}{NM_1}
\end{aligned}$$

The bias of $\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d)$ is

$$\begin{aligned}
&\mathbb{E}\left[\hat{\tilde{U}}^{N,M_1}_{\mu_2,2}(d)-\tilde{U}_{\mu_2,2}(d)\right]\\
=&-2\mathbb{E}\left\{\ln p(y|\theta,d)\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1}p(y|\theta^{(\cdot,j)},d)\right]\right\}+2\mathbb{E}\left[\ln p(y|\theta,d)\ln p(y|d)\right]\\
=&-2\mathbb{E}\mathbb{E}\left\{\ln p(y|\theta,d)\ln\left[\frac{1}{M_1}\sum_{j=1}^{M_1}p(y|\theta^{(\cdot,j)},d)\right]\middle|\theta,y\right\}+2\mathbb{E}\left[\ln p(y|\theta,d)\ln p(y|d)\right]\\
=&-2\mathbb{E}\left\{\ln p(y|\theta,d)\left[\ln p(y|d)-\frac{\mathbb{V}\left[p(y|\theta^*,d)|y\right]}{2p(y|d)^2}\frac{1}{M_1}\right]\right\}+2\mathbb{E}\left[\ln p(y|\theta,d)\ln p(y|d)\right]\\
=&\frac{1}{M_1}\mathbb{E}\left[\frac{\ln p(y|\theta,d)\mathbb{V}\left[p(y|\theta^*,d)|y\right]}{p(y|d)^2}\right]\\
=&\frac{E_4(d)}{M_1}
\end{aligned}$$

## C.4 Variance and bias of $\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d)$

The variance of $\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d)$ can be decomposed as

$$\mathbb{V}\left[\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d)\right] = \frac{1}{N}\mathbb{V}\left\{\left[\frac{M_1}{\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)}\frac{1}{M_2}\sum_{k=1}^{M_2} p(y|\theta^{(\cdot,k)},d)\ln p(y|\theta^{(\cdot,k)},d)\right]^2\right\} \tag{C.7}$$

$$= \frac{1}{N}\mathbb{V}\mathbb{E}\left\{\left[\frac{M_1}{\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)}\frac{1}{M_2}\sum_{k=1}^{M_2} p(y|\theta^{(\cdot,k)},d)\ln p(y|\theta^{(\cdot,k)},d)\right]^2\middle| y\right\} \tag{C.8}$$

$$+ \frac{1}{N}\mathbb{E}\mathbb{V}\left\{\left[\frac{M_1}{\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)}\frac{1}{M_2}\sum_{k=1}^{M_2} p(y|\theta^{(\cdot,k)},d)\ln p(y|\theta^{(\cdot,k)},d)\right]^2\middle| y\right\}$$

It is easy to get

$$\begin{aligned}
&\mathbb{E}\left[\frac{M_1}{\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)}\frac{1}{M_2}\sum_{k=1}^{M_2} p(y|\theta^{(\cdot,k)},d)\ln p(y|\theta^{(\cdot,k)},d)\middle| y\right]\\
=&\mathbb{E}\left[\frac{\frac{1}{M_2}\sum_{k=1}^{M_2} p(y|\theta^{(\cdot,k)},d)\ln p(y|\theta^{(\cdot,k)},d)}{\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)}\middle| y\right]\\
\approx&\frac{\mathbb{E}\left[p(y|\theta',d)\ln p(y|\theta',d)|y\right]}{p(y|d)} + \frac{\mathbb{E}\left[p(y|\theta',d)\ln p(y|\theta',d)|y\right]}{p(y|d)^3}\mathbb{V}\left[\frac{1}{M_1}\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)\middle| y\right]\\
=&\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right] + \frac{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]}{p(y|d)^2}\mathbb{V}\left[p(y|\theta^*,d)|y\right]\frac{1}{M_1}
\end{aligned}$$

where $\theta'$ represents the random variable $\theta^{(\cdot,k)}$, and the superscript is used to distinguish the inner $\theta'$ from the outer random variable $\theta$ for $\theta^{(\cdot)}$ and the other inner random variable $\theta^*$ for $\theta^{(\cdot,j)}$. In the second equality for expanding the condition expectation, we discard the covariance term because the numerator and denominator are independent. Notice that $\mathbb{E}\left[f(\theta)|y\right]$ and $\mathbb{E}_{\theta|y}\left[f(\theta)\right]$ are different. $\theta$ follows the prior distribution in the former expectation with a fixed $y$, while in the latter one, $\theta$

follows the posterior distribution conditioned on $y$. The last equality is due to

$$\begin{aligned}\frac{\mathbb{E}\left[p(y|\theta',d)\ln p(y|\theta',d)|y\right]}{p(y|d)} &= \frac{\int_\Theta p(\theta')p(y|\theta',d)\ln p(y|\theta',d)\,d\theta'}{p(y|d)}\\ &= \int_\Theta \frac{p(\theta')p(y|\theta',d)}{p(y|d)}\ln p(y|\theta',d)\,d\theta'\\ &= \int_\Theta p(\theta'|y,d)\ln p(y|\theta',d)\,d\theta'\\ &= \mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]\end{aligned}$$

with the understanding that the conditioning on $d$ is always implied. We can also get

$$\begin{aligned}&\mathbb{V}\left[\frac{M_1}{\sum_{j=1}^{M_1}p(y|\theta^{(\cdot,j)},d)}\frac{1}{M_2}\sum_{k=1}^{M_2}p(y|\theta^{(\cdot,k)},d)\ln p(y|\theta^{(\cdot,k)},d)\middle|y\right]\\ =&\mathbb{V}\left[\frac{\frac{1}{M_2}\sum_{k=1}^{M_2}p(y|\theta^{(\cdot,k)},d)\ln p(y|\theta^{(\cdot,k)},d)}{\frac{1}{M_1}\sum_{j=1}^{M_1}p(y|\theta^{(\cdot,j)},d)}\middle|y\right]\\ \approx&\frac{\frac{1}{M_2}\mathbb{V}\left[p(y|\theta',d)\ln p(y|\theta',d)|y\right]}{p(y|d)^2}+\frac{\left\{\mathbb{E}\left[p(y|\theta',d)\ln p(y|\theta',d)|y\right]\right\}^2}{p(y|d)^4}\mathbb{V}\left[\frac{1}{M_1}\sum_{j=1}^{M_1}p(y|\theta^{(\cdot,j)},d)\middle|y\right]\\ =&\frac{\mathbb{V}\left[p(y|\theta',d)\ln p(y|\theta',d)|y\right]}{p(y|d)^2}\frac{1}{M_2}+\frac{\left\{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]\right\}^2}{p(y|d)^2}\mathbb{V}\left[p(y|\theta^*,d)|y\right]\frac{1}{M_1}\end{aligned}$$

Hence,

$$\begin{aligned}&\mathbb{E}\left\{\left[\frac{M_1}{\sum_{j=1}^{M_1}p(y|\theta^{(\cdot,j)},d)}\frac{1}{M_2}\sum_{k=1}^{M_2}p(y|\theta^{(\cdot,k)},d)\ln p(y|\theta^{(\cdot,k)},d)\right]^2\middle|y\right\}\\ \approx&\left\{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]+\frac{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]}{p(y|d)^2}\mathbb{V}\left[p(y|\theta^*,d)|y\right]\frac{1}{M_1}\right\}^2\\ &+\frac{\mathbb{V}\left[p(y|\theta',d)\ln p(y|\theta',d)|y\right]}{p(y|d)^2}\frac{1}{M_2}\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{V}\left\{\left[\frac{M_1}{\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)}\frac{1}{M_2}\sum_{k=1}^{M_2} p(y|\theta^{(\cdot,k)},d)\ln p(y|\theta^{(\cdot,k)},d)\right]^2\middle| y\right\}\\
\approx& 4\left\{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]+\frac{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]}{p(y|d)^2}\mathbb{V}\left[p(y|\theta^*,d)|y\right]\frac{1}{M_1}\right\}^2\\
&\times\frac{\mathbb{V}\left[p(y|\theta',d)\ln p(y|\theta',d)|y\right]}{p(y|d)^2}\frac{1}{M_2}
\end{aligned}$$

By plugging the above two equations into Eqn. (C.8), we can then get the variance of $\tilde{U}_{\mu_2,3}(d)$ as

$$\begin{aligned}
&\mathbb{V}\left[\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d)\right]\\
\approx&\frac{1}{N}\mathbb{V}\left\{\left\{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]+\frac{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]}{p(y|d)^2}\mathbb{V}\left[p(y|\theta^*,d)|y\right]\frac{1}{M_1}\right\}^2\right.\\
&\qquad\left.+\frac{\mathbb{V}\left[p(y|\theta',d)\ln p(y|\theta',d)|y\right]}{p(y|d)^2}\frac{1}{M_2}\right\}\\
&+\frac{1}{NM_2}\mathbb{E}\left\{4\left\{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]+\frac{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]}{p(y|d)^2}\mathbb{V}\left[p(y|\theta^*,d)|y\right]\frac{1}{M_1}\right\}^2\right.\\
&\qquad\left.\times\frac{\mathbb{V}\left[p(y|\theta',d)\ln p(y|\theta',d)|y\right]}{p(y|d)^2}\frac{1}{M_2}\right\}\\
=&\frac{A_5(d)}{N}+\frac{B_5(d)}{NM_1}+\frac{C_5(d)}{NM_2}
\end{aligned}$$

and the bias is

$$
\begin{aligned}
&\mathbb{E}\left[\hat{\tilde{U}}^{N,M_1,M_2}_{\mu_2,3}(d) - \tilde{U}_{\mu_2,3}(d)\right] \\
=&\mathbb{E}\left\{\left[\frac{M_1}{\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)}\frac{1}{M_2}\sum_{k=1}^{M_2} p(y|\theta^{(\cdot,k)},d)\ln p(y|\theta^{(\cdot,k)},d)\right]^2\right\} \\
&-\mathbb{E}\left\{\left[\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]\right]^2\right\} \\
=&\mathbb{E}\mathbb{E}\left\{\left[\frac{M_1}{\sum_{j=1}^{M_1} p(y|\theta^{(\cdot,j)},d)}\frac{1}{M_2}\sum_{k=1}^{M_2} p(y|\theta^{(\cdot,k)},d)\ln p(y|\theta^{(\cdot,k)},d)\right]^2\middle| y\right\} \\
&-\mathbb{E}\left\{\left[\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]\right]^2\right\} \\
\approx&\mathbb{E}\left\{\left\{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right] + \frac{\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]}{p(y|d)^2}\mathbb{V}\left[p(y|\theta^*,d)|y\right]\frac{1}{M_1}\right\}^2\right. \\
&\qquad\left.+\frac{\mathbb{V}\left[p(y|\theta',d)\ln p(y|\theta',d)|y\right]}{p(y|d)^2}\frac{1}{M_2}\right\} \\
&-\mathbb{E}\left\{\left[\mathbb{E}_{\theta'|y}\left[\ln p(y|\theta',d)\right]\right]^2\right\} \\
\approx&\frac{E_5(d)}{M_1} + \frac{F_5(d)}{M_2}
\end{aligned}
$$

# APPENDIX D

# Appendix of robust sequential optimal experimental design (rsOED)

## D.1 The recursive relationship of the variance action-value function

We denote $G_{k+1} = \sum_{t=k+1}^{N} g_t(x_t, d_t, y_t)$ and note that $\mathbb{E}_{\dots|\pi,x_k,d_k,y_k} = \mathbb{E}_{\dots|\pi,x_{k+1}}$. The recursive relationship of the variance action-value function can be obtained by the following steps.

$$
\begin{aligned}
&\tilde{Q}_k^{\pi_w}(x_k, d_k) \\
&= \mathbb{E}_{y_k,\dots,y_{N-1}|\pi_w,x_k,d_k} \left\{ \left[ g_k(x_k, d_k, y_k) + G_{k+1} - Q_k^{\pi_w}(x_k, d_k) \right]^2 \right\} \\
&= \mathbb{E}_{y_k,\dots,y_{N-1}|\pi_w,x_k,d_k} \left\{ \left[ g_k + G_{k+1} - Q_k^{\pi_w}(x_k, d_k) + V_{k+1}^{\pi_w}(x_{k+1}) - V_{k+1}^{\pi_w}(x_{k+1}) \right]^2 \right\} \\
&= \mathbb{E}_{y_k,\dots,y_{N-1}|\pi_w,x_k,d_k} \Big\{ \left[ g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, d_k) \right]^2 + \left[ G_{k+1} - V_{k+1}^{\pi_w}(x_{k+1}) \right]^2 \\
&\qquad\qquad + 2 \left[ g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, d_k) \right] \left[ G_{k+1} - V_{k+1}^{\pi_w}(x_{k+1}) \right] \Big\} \\
&= \mathbb{E}_{y_k|x_k,d_k} \left\{ \left[ g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, d_k) \right]^2 \right\} \\
&\quad + \mathbb{E}_{y_k|x_k,d_k} \left\{ \mathbb{E}_{y_{k+1},\dots,y_{N-1}|\pi_w,x_k,d_k,y_k} \left\{ \left[ G_{k+1} - V_{k+1}^{\pi_w}(x_{k+1}) \right]^2 \right\} \right\} \\
&\quad + 2\mathbb{E}_{y_k|x_k,d_k} \left\{ \left[ g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, d_k) \right] \underbrace{\mathbb{E}_{y_{k+1},\dots,y_{N-1}|\pi_w,x_k,d_k,y_k} \left[ G_{k+1} - V_{k+1}^{\pi_w}(x_{k+1}) \right]}_{=0} \right\} \\
&= \mathbb{E}_{y_k|x_k,d_k} \left\{ \left[ g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, d_k) \right]^2 \right\} \\
&\quad + \mathbb{E}_{y_k|x_k,d_k} \left\{ \tilde{V}_{k+1}^{\pi_w}(x_{k+1}) \right\} \\
&\quad + 0 \\
&= \mathbb{E}_{y_k|x_k,d_k} \left\{ \underbrace{\left[ g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, d_k) \right]^2}_{:=\hat{V}_{k+1}^{\pi_w}(x_{k+1})} + \tilde{V}_{k+1}^{\pi_w}(x_{k+1}) \right\} \\
&= \mathbb{E}_{y_k|x_k,d_k} \left\{ \hat{V}_{k+1}^{\pi_w}(x_{k+1}) + \tilde{V}_{k+1}^{\pi_w}(x_{k+1}) \right\}.
\end{aligned}
$$

Note that in the fourth equality, $g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, d_k)$ is not depending on $y_{k+1}$ up to $y_{N-1}$, so it can be factored out of the expectation over $y_{k+1}$ up to $y_{N-1}$. The expectation $\mathbb{E}_{y_{k+1},\dots,y_{N-1}|\pi_w,x_k,d_k,y_k} \left[ G_{k+1} - V_{k+1}^{\pi_w}(x_{k+1}) \right]$ is equal to 0 as $\mathbb{E}_{y_{k+1},\dots,y_{N-1}|\pi_w,x_k,d_k,y_k} \left[ G_{k+1} \right]$ is exactly the definition of $V_{k+1}^{\pi_w}(x_{k+1})$.

## D.2 Policy Gradient Expression of Total Utility Variance

For the proof of the policy gradient expression of total utility variance, we use the same shorthand notations as Appendix A.2.

*Proof of Theorem* 6. We begin by recognizing that the gradient of the variance of the total utility is equivalent to the gradient of the variance state-value function at the initial stage:

$$\nabla_w \tilde{U}(w) = \nabla_w \tilde{V}_0^{\pi_w}(x_0). \tag{D.1}$$

The goal is then to derive the gradient expression for the variance state-value functions.

The recursive relationship for the gradient of the variance state-value function is

$$
\begin{aligned}
&\nabla_w \tilde{V}_k^{\pi_w}(x_k) \\
&= \nabla_w \tilde{Q}_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k)) \\
&= \nabla_w \int_{y_k} P(y_k|x_k, \mu_{k,w_k}(x_k)) \left[\hat{V}_{k+1}^{\pi_w}(x_{k+1}) + \tilde{V}_{k+1}^{\pi_w}(x_{k+1})\right] dy_k \\
&= \nabla_w \int_{x_{k+1}} P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \left[\hat{V}_{k+1}^{\pi_w}(x_{k+1}) + \tilde{V}_{k+1}^{\pi_w}(x_{k+1})\right] dx_{k+1} \\
&= \int_{x_{k+1}} \nabla_w \mu_{k,w_k}(x_k) \nabla_{d_k} P(x_{k+1}|x_k, d_k)|_{d_k=\mu_{k,w_k}(x_k)} \left[\hat{V}_{k+1}^{\pi_w}(x_{k+1}) + \tilde{V}_{k+1}^{\pi_w}(x_{k+1})\right] dx_{k+1} \\
&\quad + \int_{x_{k+1}} P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \nabla_w \left[\hat{V}_{k+1}^{\pi_w}(x_{k+1}) + \tilde{V}_{k+1}^{\pi_w}(x_{k+1})\right] dx_{k+1} \\
&= \nabla_w \mu_{k,w_k}(x_k) \nabla_{d_k} \int_{x_{k+1}} P(x_{k+1}|x_k, d_k)|_{d_k=\mu_{k,w_k}(x_k)} \left[\hat{V}_{k+1}^{\pi_w}(x_{k+1}) + \tilde{V}_{k+1}^{\pi_w}(x_{k+1})\right] dx_{k+1} \\
&\quad + \int_{x_{k+1}} P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \nabla_w \left\{\left[g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k))\right]^2\right\} dx_{k+1} \\
&\quad + \int_{x_{k+1}} P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \nabla_w \tilde{V}_{k+1}^{\pi_w}(x_{k+1}) dx_{k+1} \\
&= \nabla_w \mu_{k,w_k}(x_k) \nabla_{d_k} \tilde{Q}_k^{\pi_w}(x_k, d_k)|_{d_k=\mu_{k,w_k}(x_k)} \\
&\quad + \int_{x_{k+1}} 2P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \left[g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k))\right] \nabla_w V_{k+1}^{\pi_w}(x_{k+1}) dx_{k+1} \\
&\quad - \int_{x_{k+1}} 2P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \left[g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k))\right] \nabla_w Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k)) dx_{k+1} \\
&\quad + \int_{x_{k+1}} P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \nabla_w \tilde{V}_{k+1}^{\pi_w}(x_{k+1}) dx_{k+1} \\
&= \nabla_w \mu_{k,w_k}(x_k) \nabla_{d_k} \tilde{Q}_k^{\pi_w}(x_k, d_k)|_{d_k=\mu_{k,w_k}(x_k)} \\
&\quad + \int_{x_{k+1}} 2P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \left[g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k))\right] \nabla_w V_{k+1}^{\pi_w}(x_{k+1}) dx_{k+1} \\
&\quad - \nabla_w Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k)) \underbrace{\int_{x_{k+1}} 2P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \left[g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k))\right] dx_{k+1}}_{=0} \\
&\quad + \int_{x_{k+1}} P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \nabla_w \tilde{V}_{k+1}^{\pi_w}(x_{k+1}) dx_{k+1} \\
&= \cdots \text{to be continued in the next page}
\end{aligned}
$$

$$
\begin{aligned}
&\nabla_w \tilde{V}_k^{\pi_w}(x_k) \\
&= \nabla_w \mu_{k,w_k}(x_k) \nabla_{d_k} \tilde{Q}_k^{\pi_w}(x_k, d_k)|_{d_k=\mu_{k,w_k}(x_k)} \\
&\quad + \int_{x_{k+1}} 2P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \left[ g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k)) \right] \times \\
&\qquad \left[ \sum_{l=k+1}^{N-1} \int_{x_l} P(x_{k+1} \to x_l|\pi_w) \nabla_w \mu_{l,w_l}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)|_{d_l=\mu_{l,w_l}(x_l)} dx_l \right] dx_{k+1} \\
&\quad + \int_{x_{k+1}} P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \nabla_w \tilde{V}_{k+1}^{\pi_w}(x_{k+1}) dx_{k+1}
\end{aligned} \tag{D.2}
$$

Applying the recursive formula in Eqn. (D.2) to itself repeatedly and expanding out the overall

expression, we obtain

$$
\begin{aligned}
&\nabla_w \tilde{V}_k^{\pi_w}(x_k) \\
&= \nabla_w \mu_{k,w_k}(x_k) \nabla_{d_k} \tilde{Q}_k^{\pi_w}(x_k, d_k)|_{d_k=\mu_{k,w_k}(x_k)} \\
&\quad + \int_{x_{k+1}} 2P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \left[ g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k)) \right] \times \\
&\qquad \left[ \sum_{l=k+1}^{N-1} \int_{x_l} P(x_{k+1} \to x_l|\pi_w) \nabla_w \mu_{l,w_l}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)|_{d_l=\mu_{l,w_l}(x_l)} dx_l \right] dx_{k+1} \\
&\quad + \int_{x_{k+1}} P(x_{k+1}|x_k, \mu_{k,w_k}(x_k)) \Bigg\{ \\
&\qquad \nabla_w \mu_{k+1,w_{k+1}}(x_{k+1}) \nabla_{d_{k+1}} \tilde{Q}_{k+1}^{\pi_w}(x_{k+1}, d_{k+1})|_{d_{k+1}=\mu_{k+1,w_{k+1}}(x_{k+1})} \\
&\qquad + \int_{x_{k+2}} 2P(x_{k+2}|x_{k+1}, \mu_{k+1,w_{k+1}}(x_{k+1})) \left[ g_{k+1} + V_{k+2}^{\pi_w}(x_{k+2}) - Q_{k+1}^{\pi_w}(x_{k+1}, \mu_{k+1,w_{k+1}}(x_{k+1})) \right] \times \\
&\qquad\quad \left[ \sum_{l=k+2}^{N-1} \int_{x_l} P(x_{k+2} \to x_l|\pi_w) \nabla_w \mu_{l,w_l}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)|_{d_l=\mu_{l,w_l}(x_l)} dx_l \right] dx_{k+2} \\
&\qquad + \int_{x_{k+2}} P(x_{k+2}|x_{k+1}, \mu_{k+1,w_{k+1}}(x_{k+1})) \nabla_w \tilde{V}_{k+2}^{\pi_w}(x_{k+2}) dx_{k+2} \Bigg\} dx_{k+1} \\
&= \sum_{t=k}^{k+1} \int_{x_t} P(x_k \to x_t|\pi_w) \nabla_w \mu_{t,w}(x_t) \nabla_{d_t} \tilde{Q}_t^{\pi_w}(x_t, d_t)|_{d_t=\mu_{t,w}(x_t)} dx_t \\
&\quad + \sum_{t=k}^{k+1} \int_{x_{t+1}} 2P(x_k \to x_{t+1}|\pi_w) \left[ g_t + V_{t+1}^{\pi_w}(x_{t+1}) - Q_t^{\pi_w}(x_t, \mu_{t,w}(x_t)) \right] \times \\
&\qquad \left[ \sum_{l=t+1}^{N-1} \int_{x_l} P(x_{t+1} \to x_l|\pi_w) \nabla_w \mu_{l,w_l}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)|_{d_l=\mu_{l,w_l}(x_l)} dx_l \right] dx_{t+1} \\
&\quad + \int_{x_{k+2}} P(x_k \to x_{k+2}|\pi_w) \nabla_w \tilde{V}_{k+2}^{\pi_w}(x_{k+2}) dx_{k+2} \\
&\vdots \\
&= \sum_{t=k}^{N-2} \int_{x_t} P(x_k \to x_t|\pi_w) \nabla_w \mu_{t,w}(x_t) \nabla_{d_t} \tilde{Q}_t^{\pi_w}(x_t, d_t)|_{d_t=\mu_{t,w}(x_t)} dx_t \\
&\quad + \sum_{t=k}^{N-2} \int_{x_{t+1}} 2P(x_k \to x_{t+1}|\pi_w) \left[ g_t + V_{t+1}^{\pi_w}(x_{t+1}) - Q_t^{\pi_w}(x_t, \mu_{t,w}(x_t)) \right] \times \\
&\qquad \left[ \sum_{l=t+1}^{N-1} \int_{x_l} P(x_{t+1} \to x_l|\pi_w) \nabla_w \mu_{l,w_l}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)|_{d_l=\mu_{l,w_l}(x_l)} dx_l \right] dx_{t+1} \\
&\quad + \int_{x_{N-1}} P(x_k \to x_{N-1}|\pi_w) \nabla_w \tilde{V}_{N-1}^{\pi_w}(x_{N-1}) dx_{N-1}
\end{aligned}
$$

For the last term, we have that

$$
\begin{aligned}
&\nabla_w \tilde{V}_{N-1}^{\pi_w}(x_{N-1}) \\
&= \nabla_w \tilde{Q}_{N-1}^{\pi_w}(x_{N-1}, \mu_{N-1,w}(x_{N-1})) \\
&= \nabla_w \int_{x_N} P(x_N|x_{N-1}, \mu_{N-1,w}(x_{N-1})) \left[\hat{V}_N^{\pi_w}(x_N) + \tilde{V}_N^{\pi_w}(x_N)\right] dx_N \\
&= \int_{x_N} \nabla_w \mu_{N-1,w}(x_{N-1}) \nabla_{d_{N-1}} P(x_N|x_{N-1}, d_{N-1})|_{d_{N-1}=\mu_{N-1,w}(x_{N-1})} \left[\hat{V}_N^{\pi_w}(x_N) + \tilde{V}_N^{\pi_w}(x_N)\right] dx_N \\
&\quad + \int_{x_N} P(x_N|x_{N-1}, \mu_{N-1,w}(x_{N-1})) \nabla_w \left[\hat{V}_N^{\pi_w}(x_N) + \tilde{V}_N^{\pi_w}(x_N)\right] dx_N \\
&= \nabla_w \mu_{N-1,w}(x_{N-1}) \nabla_{d_{N-1}} \int_{x_N} P(x_N|x_{N-1}, d_{N-1})|_{d_{N-1}=\mu_{N-1,w}(x_{N-1})} \left[\hat{V}_N^{\pi_w}(x_N) + \tilde{V}_N^{\pi_w}(x_N)\right] dx_N \\
&\quad + 0 \\
&= \nabla_w \mu_{N-1,w}(x_{N-1}) \nabla_{d_{N-1}} \tilde{Q}_{N-1}^{\pi_w}(x_{N-1}, d_{N-1})|_{d_{N-1}=\mu_{N-1,w}(x_{N-1})}
\end{aligned}
$$

The fourth equality is because both $\hat{V}_N^{\pi_w}(x_N)$ and $\tilde{V}_N^{\pi_w}(x_N)$ are constants with respect to $w$. Therefore, we have

$$
\begin{aligned}
&\nabla_w \tilde{V}_k^{\pi_w}(x_k) \\
&= \sum_{t=k}^{N-1} \int_{x_t} P(x_k \to x_t|\pi_w) \nabla_w \mu_{t,w}(x_t) \nabla_{d_t} \tilde{Q}_t^{\pi_w}(x_t, d_t)|_{d_t=\mu_{t,w}(x_t)} dx_t \\
&\quad + \sum_{t=k}^{N-2} \int_{x_{t+1}} 2P(x_k \to x_{t+1}|\pi_w) \left[g_t + V_{t+1}^{\pi_w}(x_{t+1}) - Q_t^{\pi_w}(x_t, \mu_{t,w}(x_t))\right] \times \\
&\qquad\qquad \left[\sum_{l=t+1}^{N-1} \int_{x_l} P(x_{t+1} \to x_l|\pi_w) \nabla_w \mu_{l,w_l}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)|_{d_l=\mu_{l,w_l}(x_l)} dx_l\right] dx_{t+1} \quad \text{(D.3)}
\end{aligned}
$$

Finally, by substituting Eqn. (D.3) into Eqn. (D.1), we obtain the policy gradient expression:

$$
\begin{aligned}
\nabla_w \tilde{U}(w) &= \nabla_w \tilde{V}_0^{\pi_w}(x_0) \\
&= \sum_{k=0}^{N-1} \int_{x_k} P(x_0 \to x_k | \pi_w) \nabla_w \mu_{k,w_k}(x_k) \nabla_{d_k} \tilde{Q}_k^{\pi_w}(x_k, d_k)|_{d_k=\mu_{k,w_k}(x_k)} dx_k \\
&\quad + \sum_{k=0}^{N-2} \int_{x_{k+1}} 2P(x_0 \to x_{k+1} | \pi_w) \left[ g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k)) \right] \times \\
&\qquad\qquad \left[ \sum_{l=k+1}^{N-1} \int_{x_l} P(x_{k+1} \to x_l | \pi_w) \nabla_w \mu_{l,w_l}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)|_{d_l=\mu_{l,w_l}(x_l)} dx_l \right] dx_{k+1} \\
&= \sum_{k=0}^{N-1} \mathbb{E}_{x_k|\pi_w,x_0} \left[ \nabla_w \mu_{k,w_k}(x_k) \nabla_{d_k} \tilde{Q}_k^{\pi_w}(x_k, d_k) \Big|_{d_k=\mu_{k,w_k}(x_k)} \right] \\
&\quad + \sum_{k=0}^{N-2} \mathbb{E}_{x_{k+1}|\pi_w,x_0} \Bigg\{ 2 \left[ g_k + V_{k+1}^{\pi_w}(x_{k+1}) - Q_k^{\pi_w}(x_k, \mu_{k,w_k}(x_k)) \right] \times \\
&\qquad\qquad \sum_{l=k+1}^{N-1} \mathbb{E}_{x_l|\pi_w,x_{k+1}} \left[ \nabla_w \mu_{l,w_l}(x_l) \nabla_{d_l} Q_l^{\pi_w}(x_l, d_l)|_{d_l=\mu_{l,w_l}(x_l)} \right] \Bigg\}.
\end{aligned}
$$

□

# BIBLIOGRAPHY


[1] A. Alexanderian, N. Petra, G. Stadler, and O. Ghattas. A fast and scalable method for A-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems. *SIAM Journal on Scientific Computing*, 38(1):A243–A272, 2016.

[2] L. J. Allen. A primer on stochastic epidemic models: Formulation, numerical simulation, and analysis. *Infectious Disease Modelling*, 2(2):128–142, 2017.

[3] K. J. Arrow, H. B. Chenery, B. S. Minhas, and R. M. Solow. Capital-labor substitution and economic efficiency. *The review of Economics and Statistics*, pages 225–250, 1961.

[4] A. Atkinson, A. Donev, and R. Tobias. *Optimum experimental designs, with SAS*, volume 34. Oxford University Press, 2007.

[5] A. C. Atkinson and V. Fedorov. The design of experiments for discriminating between two rival models. *Biometrika*, 62(1):57–70, 1975.

[6] A. Attia, A. Alexanderian, and A. K. Saibaba. Goal-oriented optimal design of experiments for large-scale Bayesian linear inverse problems. *Inverse Problems*, 34(9):095009, 2018.

[7] D. Barber and F. Agakov. The IM algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16(320):201, 2004.

[8] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. Tb, A. Muldal, N. Heess, and T. Lillicrap. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

[9] A. Basu, T. Bhattacharyya, and V. S. Borkar. A learning algorithm for risk-sensitive cost. *Mathematics of operations research*, 33(4):880–898, 2008.

[10] R. A. Bates, R. S. Kenett, D. M. Steinberg, and H. P. Wynn. Achieving robust design from computer simulations. *Quality Technology & Quantitative Management*, 3(2):161–177, 2006.

[11] J. Beck, B. M. Dia, L. F. Espath, Q. Long, and R. Tempone. Fast Bayesian experimental design: Laplace-based importance sampling for the expected information gain. *Computer Methods in Applied Mechanics and Engineering*, 334:523–553, 2018.

[12] J. Berger and L. M. Berliner. Robust Bayes and empirical Bayes analysis with $\varepsilon$-contaminated priors. *The Annals of Statistics*, pages 461–486, 1986.

[13] J. O. Berger. *Statistical decision theory and Bayesian analysis*. Springer New York, New York, NY, 1985.

[14] T. Blau, E. V. Bonilla, I. Chades, and A. Dezfouli. Optimizing sequential experimental design with deep reinforcement learning. In *International Conference on Machine Learning*, pages 2107–2128. PMLR, 2022.

[15] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[16] V. S. Borkar. A sensitivity formula for risk-sensitive cost and the actor–critic algorithm. *Systems & Control Letters*, 44(5):339–346, 2001.

[17] D. M. Borth. A total entropy criterion for the dual problem of model discrimination and parameter estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, 37(1):77–87, 1975.

[18] G. E. Box and W. U.-M. M. R. CENTER. Choice of response surface design and alphabetic optimality. In *Proceedings of the... Conference on the Design of Experiments in Army Research, Development and Testing*, volume 28, page 237, 1982.

[19] G. E. Box and W. J. Hill. Discrimination among mechanistic models. *Technometrics*, 9(1):57–71, 1967.

[20] G. E. P. Box. Sequential experimentation and sequential assembly of designs. *Quality Engineering*, 5(2):321–330, 1992.

[21] G. E. P. Box and H. L. Lucas. Design of experiments in non-linear situations. *Biometrika*, 46(1-2):77–90, 1959.

[22] G. L. Boylan, P. L. Goethals, and B. R. Cho. Robust parameter design in resource-constrained environments: An investigation of trade-offs between costs and precision within variable processes. *Applied Mathematical Modelling*, 37(4):2394–2416, 2013.

[23] E. Brochu, V. Cora, and N. D. Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[24] A. E. Brockwell and J. B. Kadane. A gridding method for Bayesian sequential decision problems. *Journal of Computational and Graphical Statistics*, 12(3):566–584, 2003.

[25] M. Brown, F. He, and L. F. Yeung. Robust measurement selection for biochemical pathway experimental design. *International journal of bioinformatics research and applications*, 4(4):400–416, 2008.

[26] J. Burkardt. The truncated normal distribution. *Department of Scientific Computing Website, Florida State University*, 1:35, 2014.

[27] A. N. Burnetas and M. N. Katehakis. Optimal adaptive policies for markov decision processes. *Mathematics of Operations Research*, 22(1):222–255, 1997.

[28] T. Butler, J. D. Jakeman, and T. Wildey. Optimal experimental design for prediction based on push-forward probability measures. *Journal of Computational Physics*, 416:109518, Sept. 2020.

[29] B. P. Carlin, J. B. Kadane, and A. E. Gelfand. Approaches for optimal sequential decision analysis in clinical trials. *Biometrics*, 54(3):964–975, 1998.

[30] A. R. Cassandra. A survey of POMDP applications. In *Working notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes*, volume 1724, 1998.

[31] A. R. Cassandra, L. P. Kaelbling, and M. L. Littman. Acting optimally in partially observable stochastic domains. In *Aaai*, volume 94, pages 1023–1028, 1994.

[32] D. R. Cavagnaro, J. I. Myung, M. A. Pitt, and J. V. Kujala. Adaptive design optimization: A mutual information-based approach to model discrimination in cognitive science. *Neural Computation*, 22(4):887–905, 2010.

[33] K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[34] K. M. Chaloner. *Optimal Bayesian experimental design for linear models*. Carnegie Mellon University, 1982.

[35] Y. Chow and M. Ghavamzadeh. Algorithms for CVaR optimization in MDPs. *Advances in neural information processing systems*, 27, 2014.

[36] J. A. Christen and M. Nakamura. Sequential stopping rules for species accumulation. *Journal of Agricultural, Biological & Environmental Statistics*, 8(2):184–195, 2003.

[37] A. R. Cook, G. J. Gibson, and C. A. Gilligan. Optimal observation times in experimental epidemic processes. *Biometrics*, 64(3):860–868, 2008.

[38] R. T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13, 1946.

[39] K. Csilléry, M. G. Blum, O. E. Gaggiotti, and O. François. Approximate Bayesian computation (ABC) in practice. *Trends in ecology & evolution*, 25(7):410–418, 2010.

[40] A. DasGupta and W. Studden. Robust Bayesian experimental designs in normal linear models. *The Annals of Statistics*, 19(3):1244–1256, 1991.

[41] T. Degris, M. White, and R. S. Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

[42] L. Dinh, J. N. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803*, 2016.

[43] H. A. Dror and D. M. Steinberg. Robust experimental design for multivariate generalized linear models. *Technometrics*, 48(4):520–529, 2006.

[44] H. A. Dror and D. M. Steinberg. Sequential experimental designs for generalized linear models. *Journal of the American Statistical Association*, 103(481):288–298, 2008.

[45] C. C. Drovandi, J. M. McGree, and A. N. Pettitt. Sequential Monte Carlo for Bayesian sequentially designed experiments for discrete data. *Computational Statistics & Data Analysis*, 57(1):320–335, 2013.

[46] C. C. Drovandi, J. M. McGree, and A. N. Pettitt. A sequential Monte Carlo algorithm to incorporate model uncertainty in Bayesian sequential design. *Journal of Computational and Graphical Statistics*, 23(1):3–24, 2014.

[47] J. A. Duersch and T. A. Catanach. Generalizing information to the evolution of rational belief. *Entropy*, 22(1):108, 2020.

[48] D. Duffie and J. Pan. An overview of value at risk. *Journal of derivatives*, 4(3):7–49, 1997.

[49] J. Engel and A. F. Huele. A generalized linear modeling approach to robust design. *Technometrics*, 38(4):365–373, 1996.

[50] V. V. Fedorov. *Theory of optimal experiments*. Academic Press, New York, NY, 1972.

[51] C. Feng and Y. M. Marzouk. A layered multiple importance sampling scheme for focused optimal Bayesian experimental design. *arXiv preprint arXiv:1903.11187*, 2019.

[52] J. A. Filar, L. C. Kallenberg, and H.-M. Lee. Variance-penalized Markov decision processes. *Mathematics of Operations Research*, 14(1):147–161, 1989.

[53] I. Ford, D. M. Titterington, and C. P. Kitsos. Recent advances in nonlinear experimental design. *Technometrics*, 31(1):49–60, 1989.

[54] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, et al. Noisy networks for exploration. *arXiv preprint arXiv:1706.10295*, 2017.

[55] A. Foster, D. R. Ivanova, I. Malik, and T. Rainforth. Deep adaptive design: Amortizing sequential Bayesian experimental design. In *International Conference on Machine Learning*, pages 3384–3395. PMLR, 2021.

[56] A. Foster, M. Jankowiak, E. Bingham, P. Horsfall, Y. W. Teh, T. Rainforth, and N. Goodman. Variational Bayesian optimal experimental design. *Advances in Neural Information Processing Systems*, 32, 2019.

[57] A. Foster, M. Jankowiak, M. O'Meara, Y. W. Teh, and T. Rainforth. A unified stochastic gradient approach to designing Bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*, pages 2959–2969. PMLR, 2020.

[58] P. I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[59] S. Fujimoto, H. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[60] R. Gautier and L. Pronzato. Adaptive control for sequential design. *Discussiones Mathematicae Probability and Statistics*, 20(1):97–113, 2000.

[61] J. Ginebra. On the measure of the information in a statistical experiment. *Bayesian Analysis*, 2(1):167–212, 2007.

[62] T. Goda, T. Hironaka, and T. Iwamoto. Multilevel Monte Carlo estimation of expected information gains. *Stochastic Analysis and Applications*, 38(4):581–600, 2020.

[63] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

[64] M. Hainy, C. C. Drovandi, and J. M. McGree. Likelihood-free extensions for Bayesian sequentially designed experiments. In *mODa 11-Advances in Model-Oriented Design and Analysis: Proceedings of the 11th International Workshop in Model-Oriented Design and Analysis held in Hamminkeln, Germany, June 12-17, 2016*, pages 153–161. Springer, 2016.

[65] M. Hainy, D. J. Price, O. Restif, and C. Drovandi. Optimal Bayesian design for model discrimination via classification. *Statistics and Computing*, 32(2):25, 2022.

[66] H. Hasselt. Double Q-learning. *Advances in neural information processing systems*, 23, 2010.

[67] F. He, M. Brown, and H. Yue. Maximin and Bayesian robust experimental design for measurement set selection in modelling biochemical regulatory systems. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 20(9):1059–1078, 2010.

[68] M. Heger. Consideration of risk in reinforcement learning. In *Machine Learning Proceedings 1994*, pages 105–111. Elsevier, 1994.

[69] T. Hossain, W. Shen, A. Antar, S. Prabhudesai, S. Inoue, X. Huan, and N. Banovic. A Bayesian approach for quantifying data scarcity when modeling human behavior via inverse reinforcement learning. *ACM Transactions on Computer-Human Interaction*, 30(1):1–27, 2023.

[70] R. A. Howard and J. E. Matheson. Risk-sensitive Markov decision processes. *Management science*, 18(7):356–369, 1972.

[71] X. Huan. *Numerical approaches for sequential Bayesian optimal experimental design*. PhD thesis, Massachusetts Institute of Technology, 2015.

[72] X. Huan and Y. M. Marzouk. Simulation-based optimal Bayesian experimental design for nonlinear systems. *Journal of Computational Physics*, 232(1):288–317, 2013.

[73] X. Huan and Y. M. Marzouk. Gradient-based stochastic optimization methods in Bayesian experimental design. *International Journal for Uncertainty Quantification*, 4(6):479–510, 2014.

[74] X. Huan and Y. M. Marzouk. Sequential Bayesian optimal experimental design via approximate dynamic programming. *arXiv preprint arXiv:1604.08320*, 2016.

[75] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson. Deep variational reinforcement learning for POMDPs. In *International Conference on Machine Learning*, pages 2117–2126. PMLR, 2018.

[76] D. R. Ivanova, A. Foster, S. Kleinegesse, M. U. Gutmann, and T. Rainforth. Implicit deep adaptive design: policy-based experimental design without likelihoods. *Advances in Neural Information Processing Systems*, 34:25785–25798, 2021.

[77] A. Jain, G. Patil, A. Jain, K. Khetarpal, and D. Precup. Variance penalized on-policy and off-policy actor-critic. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7899–7907, 2021.

[78] S. M. Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

[79] W. Kim, M. A. Pitt, Z.-L. Lu, M. Steyvers, and J. I. Myung. A hierarchical adaptive approach to optimal experimental design. *Neural Computation*, 26:2565–2492, 2014.

[80] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[81] S. Kleinegesse, C. Drovandi, and M. U. Gutmann. Sequential Bayesian experimental design for implicit models via mutual information. *Bayesian Analysis*, 16(3):773–802, sep 2021.

[82] S. Kleinegesse and M. U. Gutmann. Efficient Bayesian experimental design for implicit models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 476–485. PMLR, 2019.

[83] S. Kleinegesse and M. U. Gutmann. Gradient-based Bayesian experimental design for implicit models using mutual information lower bounds. *arXiv preprint arXiv:2105.04379*, 2021.

[84] S. Körkel*, E. Kostina, H. G. Bock, and J. P. Schlöder. Numerical methods for optimal control problems in design of robust optimal experiments for nonlinear dynamic processes. *Optimization Methods and Software*, 19(3-4):327–338, 2004.

[85] A. Krishna, V. R. Joseph, S. Ba, W. A. Brenneman, and W. R. Myers. Robust experimental designs for model calibration. *Journal of Quality Technology*, pages 1–12, 2021.

[86] A. Kumar, J. Motwani, and L. Otero. An application of Taguchi's robust experimental design technique to improve service performance. *International Journal of Quality & Reliability Management*, 1996.

[87] H. Kurniawati and V. Yadav. An online POMDP solver for uncertainty planning in dynamic environment. In *Robotics Research*, pages 611–629. Springer, 2016.

[88] P. La and M. Ghavamzadeh. Actor-critic algorithms for risk-sensitive MDPs. *Advances in neural information processing systems*, 26, 2013.

[89] X. Li and X. Wang. Variance-penalized response-adaptive randomization with mismeasurement. *Journal of Statistical Planning and Inference*, 142(7):2128–2135, 2012.

[90] Y. Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[91] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[92] D. V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

[93] D. V. Lindley. *Bayesian statistics: A review*. SIAM (Society for Industrial and Applied Mathematics), Philadelphia, PA, 1972.

[94] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling. Efficient dynamic-programming updates in partially observable Markov decision processes, 1995.

[95] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling. Learning policies for partially observable environments: Scaling up. In *Machine Learning Proceedings 1995*, pages 362–370. Elsevier, 1995.

[96] Y. Liu, P. Ramachandran, Q. Liu, and J. Peng. Stein variational policy gradient. *arXiv preprint arXiv:1704.02399*, 2017.

[97] Q. Long, M. Scavino, R. Tempone, and S. Wang. Fast estimation of expected information gains for Bayesian experimental designs based on Laplace approximations. *Computer Methods in Applied Mechanics and Engineering*, 259:24–39, 2013.

[98] Q. Long, M. Scavino, R. Tempone, and S. Wang. A Laplace method for under-determined Bayesian optimal experimental designs. *Computer Methods in Applied Mechanics and Engineering*, 285:849–876, 2015.

[99] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*, 2017.

[100] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[101] H. M. Markowitz and G. P. Todd. *Mean-variance analysis in portfolio choice and capital markets*, volume 66. John Wiley & Sons, 2000.

[102] S. Masoumi, T. A. Duever, and P. M. Reilly. Sequential markov chain monte carlo (mcmc) model discrimination. *The Canadian Journal of Chemical Engineering*, 91(5):862–869, 2013.

[103] J. McGree, C. C. Drovandi, and A. N. Pettitt. A sequential Monte Carlo approach to the sequential design for discriminating between rival continuous data models. 2012.

[104] O. Mihatsch and R. Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49:267–290, 2002.

[105] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[106] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[107] R. Moriconi, M. P. Deisenroth, and K. S. Kumar. High-dimensional Bayesian optimization using low-dimensional feature spaces. *Machine Learning*, 109(9):1925–1943, 2020.

[108] P. Müller. Simulation based optimal design. *Handbook of Statistics*, 25:509–518, 2005.

[109] P. Müller, D. A. Berry, A. P. Grieve, M. Smith, and M. Krams. Simulation-based sequential Bayesian design. *Journal of Statistical Planning and Inference*, 137(10):3140–3150, 2007.

[110] P. Müller and G. Parmigiani. Optimal design via curve fitting of Monte Carlo experiments. *Journal of the American Statistical Association*, 90(432):1322–1330, 1995.

[111] S. A. Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):331–366, 2003.

[112] D. Nass, B. Belousov, and J. Peters. Entropic risk measure in policy search. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1101–1106. IEEE, 2019.

[113] F. Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python, 2014–.

[114] A. M. Overstall and D. C. Woods. Bayesian design of experiments using approximate coordinate exchange. *Technometrics*, 59(4):458–470, 2017.

[115] M. Pelikan, D. E. Goldberg, E. Cantú-Paz, et al. BOA: The Bayesian optimization algorithm. In *Proceedings of the genetic and evolutionary computation conference GECCO-99*, volume 1, pages 525–532. Citeseer, 1999.

[116] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*, 2017.

[117] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

[118] L. Prashanth and M. Ghavamzadeh. Variance-constrained actor-critic algorithms for discounted and average reward MDPs. *Machine Learning*, 105:367–417, 2016.

[119] L. Pronzato and É. Thierry. Sequential experimental design and response optimisation. *Statistical Methods and Applications*, 11(3):277–292, 2002.

[120] T. Rainforth, A. Foster, D. R. Ivanova, and F. B. Smith. Modern Bayesian experimental design. *arXiv preprint arXiv:2302.14545*, 2023.

[121] R. T. Rockafellar, S. Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[122] P. M. Roth. Design of experiments for discrimination among rival models. 1967.

[123] E. G. Ryan, C. C. Drovandi, J. M. McGree, and A. N. Pettitt. A review of modern computational algorithms for Bayesian optimal design. *International Statistical Review*, 84(1):128–154, 2016.

[124] K. J. Ryan. Estimating expected information gains for experimental designs with application to the random fatigue-limit model. *Journal of Computational and Graphical Statistics*, 12(3):585–603, 2003.

[125] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[126] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[127] D. W. Scott. *Multivariate density estimation: Theory, practice, and visualization*. John Wiley & Sons, 2015.

[128] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[129] W. Shen, J. Dong, and X. Huan. Variational sequential optimal experimental design using reinforcement learning. *arXiv preprint arXiv:2306.10430*, 2023.

[130] W. Shen and X. Huan. Bayesian sequential optimal experimental design for nonlinear models using policy gradient reinforcement learning. *Computer Methods in Applied Mechanics and Engineering, In press*, 2023.

[131] C. Sherstan, D. R. Ashley, B. Bennett, K. Young, A. White, M. White, and R. S. Sutton. Comparing direct and indirect temporal-difference methods for estimating the variance of the return. In *UAI*, pages 63–72, 2018.

[132] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *ICML*, 2014.

[133] S. Silvey. *Optimal design: An introduction to the theory for parameter estimation*, volume 1. Springer Science & Business Media, 2013.

[134] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[135] M. J. Sobel. The variance of discounted Markov decision processes. *Journal of Applied Probability*, 19(4):794–802, 1982.

[136] A. Solonen, H. Haario, and M. Laine. Simulation-based optimal design using a response variance criterion. *Journal of Computational and Graphical Statistics*, 21(1):234–252, 2012.

[137] N.-Z. Sun. Structure reduction and robust experimental design for distributed parameter identification. *Inverse Problems*, 21(2):739, 2005.

[138] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[139] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[140] G. Taguchi. System of experimental design; engineering methods to optimize quality and minimize costs. Technical report, 1987.

[141] A. Tamar, Y. Glassner, and S. Mannor. Optimizing the CVaR via sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[142] A. Tamar and S. Mannor. Variance adjusted actor critic algorithms. *arXiv preprint arXiv:1310.3697*, 2013.

[143] G. Terejanu, R. R. Upadhyay, and K. Miki. Bayesian experimental design for the active nitridation of graphite by atomic nitrogen. *Experimental Thermal and Fluid Science*, 36:178–193, 2012.

[144] O. Thomas, R. Dutta, J. Corander, S. Kaski, M. U. Gutmann, et al. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 2021.

[145] B. Toman and J. L. Gastwirth. Robust Bayesian experimental design and estimation for analysis of variance models using a class of normal mixtures. *Journal of statistical planning and inference*, 35(3):383–398, 1993.

[146] P. Tsilifis, R. G. Ghanem, and P. Hajali. Efficient Bayesian experimentation using an expected information gain lower bound. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):30–62, 2017.

[147] U. Von Toussaint. Bayesian inference in physics. *Reviews of Modern Physics*, 83:943–999, 2011.

[148] E. Walter and L. Pronzato. Robust experiment design: between qualitative and quantitative identifiabilities. *Identifiability of parametric models*, pages 104–113, 1987.

[149] J. K. Wathen and J. A. Christen. Implementation of backward induction for sequentially adaptive clinical trials. *Journal of Computational and Graphical Statistics*, 15(2):398–413, 2006.

[150] B. P. Weaver, B. J. Williams, C. M. Anderson-Cook, and D. M. Higdon. Computational enhancements to Bayesian design of experiments using Gaussian processes. *Bayesian Analysis*, 11(1):191–213, 2016.

[151] D. White. A mathematical programming approach to a problem in variance penalised Markov decision processes. *Operations-Research-Spektrum*, 15:225–230, 1994.

[152] M. White and A. White. A greedy approach to adapting the trace parameter for temporal difference learning. *arXiv preprint arXiv:1607.00446*, 2016.

[153] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[154] K. Wu, P. Chen, and O. Ghattas. A fast and scalable computational framework for large-scale high-dimensional Bayesian optimal experimental design. *SIAM/ASA Journal on Uncertainty Quantification*, 11(1):235–261, 2023.

[155] K. Wu, P. Chen, and O. Ghattas. An offline-online decomposition method for efficient linear Bayesian goal-oriented optimal experimental design: Application to optimal sensor placement. *SIAM Journal on Scientific Computing*, 45(1):B57–B77, 2023.

[156] Y. Yi and X. Wang. Response adaptive designs with a variance-penalized criterion. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 51(5):763–773, 2009.

[157] C. Ying, X. Zhou, H. Su, D. Yan, N. Chen, and J. Zhu. Towards safe reinforcement learning via constraining conditional value-at-risk. *arXiv preprint arXiv:2206.04436*, 2022.