Entropy Estimation and Optimization and their Applications in Bayesian Experimental Design

By

Ziqiao Ao

A thesis submitted to
the University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Mathematics
College of Engineering and Physical Sciences
University of Birmingham
Nov 2023

ABSTRACT

This thesis delves into the realm of information theory, focusing on the crucial concept of entropy - a measure of uncertainty in datasets or signals. Initially conceptualized by Claude E. Shannon, entropy's applications have significantly expanded, influencing disciplines ranging from physics to biology to machine learning. Central to this thesis is the challenge of entropy estimation and optimization, particularly in high-dimensional spaces, and their applications in Bayesian Experimental Design (BED).

The thesis presents and addresses three primary research objectives. The first is a comprehensive survey of existing and emerging entropy estimation and optimization methodologies, with a special emphasis on their applications in BED. This exploration covers traditional methods like plug-in and k-NN estimators, as well as advanced techniques such as variational estimators and entropy gradient estimation methods. Moreover, the connection between entropy optimization and BED is thoroughly examined, revealing the potential of entropy optimization techniques to inspire innovative BED methodologies.

The second objective of the thesis is centered on reducing the bias inherent in entropy estimators, particularly in the context of high-dimensional entropy estimation challenges. To meet this goal, the thesis introduces an innovative transform-based method for high-dimensional entropy estimation. This method integrates a novel k-NN based estimator with a normalizing flow-based mapping technique, effectively achieving a significant reduction in estimation bias when compared to traditional methods. Additionally, the thesis provides

comprehensive theoretical analyses to validate the consistency and efficiency of this proposed method, demonstrating its effectiveness in high-dimensional settings.

The third objective focuses on advancing BED through entropy gradient estimation. This objective is achieved by directly estimating the gradient of the design criterion, which includes an entropy term, with respect to design variables. Subsequently, stochastic gradient descent is applied to find the optimal design. Within this framework, the thesis introduces two novel methods for estimating the expected information gain (EIG) gradient: UEEG-MCMC and BEEG-AP. Each of these methods has its distinct advantages and limitations, which are meticulously examined and validated through a combination of theoretical analyses and empirical experiments. This comprehensive evaluation underscores the practicality and applicability of these methods in advancing the field of BED.

Finally, the thesis concludes with a discussion on its contributions and future research directions. It proposes two trajectories for further research: developing a global optimization method for BED that synergies local search capabilities of gradient-based methods with the global search efficiency of Bayesian Optimization, and extending entropy gradient estimation techniques to BED for implicit models.

DEDICATION

Dedicated to my dogs

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor, Professor Jinglai Li, whose expertise, understanding, and patience, added considerably to my doctoral experience.

I am also deeply thankful to the examiners of my thesis, Professor Ata Kaban and Dr. Yunpeng Li, for their meticulous reading, insightful analyses, and valuable suggestions, which greatly improved the quality of this thesis.

Additionally, I extend my thanks to my colleagues at the School of Mathematics at the University of Birmingham and the School of Mathematics at Shanghai Jiao Tong University. Their friendship and collaboration have been invaluable and a constant source of support throughout my research.

I gratefully acknowledge the financial support provided by the Graduate School in Mathematics and China Scholarship Council (CSC), without which this research would have been considerably more challenging.

Last but not least, I cannot forget to thank my family and friends for all the personal support, understanding, and love, which gave me the strength I needed to complete this journey.

This thesis would not have been possible without the support of all these wonderful people.

God, grant me the serenity to accept the things I cannot change,

Courage to change the things I can,

And wisdom to know the difference.

Serenity Prayer

Contents

List of Figures

List of Tables

Chapter One

Introduction

At the heart of information theory lies the concept of entropy, a measure of uncertainty or unpredictability within a dataset or signal. Initially introduced by Claude E. Shannon in his seminal 1948 paper [149], entropy has since transcended its telecommunications origins to become a cornerstone concept across a myriad of disciplines, from physics to finance, and from computer science to biology. It quantifies the amount of information contained in a dataset or signal and serves as a foundational pillar for data compression, channel coding, and the broader analysis of information systems. Within this vast landscape of application, the precise estimation of entropy has emerged as a critical challenge. Accurate entropy estimation enables us to assess the amount of information in a dataset or signal and is crucial for tasks such as image processing [29], complex network characterization [51], and cryptographic system design [172]. This estimation, however, is often beset by the intricacies of the data's underlying distribution, particularly in high-dimensional spaces where theoretical models struggle to keep pace with empirical data. Advancing a step further, the optimization of entropy has profound implications for improving signal processing and decision-making processes. By actively manipulating the parameters of a system to maximize or minimize entropy, we can direct the system toward states of either maximal randomness and unpredictability or minimal uncertainty and greater predictability. This concept is particularly pertinent in

the field of machine learning, where entropy optimization can be used to fine-tune models and algorithms for better performance. Within the expansive domain of information theory applications, Bayesian Experimental Design (BED) stands out as a prime exemplar of the application of entropy estimation and optimization. Within the BED framework, the goal is achieved by designing experiments or data collection processes that are the most informative, typically under constraints of limited resources. When utilizing Bayesian principles to update our state of knowledge about a system, entropy serves as the pivotal metric for assessing the potential value of experiments. By optimizing expected information gain of the system of interest — essentially minimizing the entropy of the posterior parameter distribution or maximizing the entropy of the observed data — we are able to determine the most efficient experimental setups.

This thesis studies the intersection of information theory and experimental design, with a specific focus on entropy estimation and optimization in the Bayesian framework. It delves deep into the challenges and practicalities of applying these theoretical concepts to real-world problems, particularly examining how entropy can be effectively estimated and optimized to maximize the amount of information extracted from experimental data. Importantly, the work uncovers the link between entropy estimation and optimization and BED. This discovery not only aligns some existing entropy estimation and optimization methods with BED approaches but also indicates that other techniques could lead to innovative BED methodologies. Building on this insight, the thesis addresses the challenges of high-dimensional entropy estimation and introduces groundbreaking approaches inspired by entropy gradient estimation in BED, emphasizing their potential to improve the efficiency of the utilization of experimental resources. By exploring the complexities of entropy estimation and optimization and their applications in experimental design, this thesis provides a comprehensive case study of how information theory plays a crucial role in the realm of data-driven research and decision-making, paving the way for future advancements in the field.

1.1 Motivations, primary objectives and contributions

1.1.1 Surveying approaches in entropy estimation and optimization, and connecting them to BED

Entropy is perhaps the most basic concept in information theory and its estimation and optimization find a large range of applications in fields such as physics [16], stochastic processes [110], graphs [37], biological signal analysis [52], survival analysis [38] and modern machine learning [178, 90, 39, 185, 168, 137, 64]. Defined mathematically as the negative expected value of the logarithm of probability, entropy, despite its seemingly straightforward definition, often requires numerical evaluation, particularly in cases where it lacks an analytical expression. This is especially true in real-world scenarios, where the relevant distribution is not analytically known, making the direct computation of entropy through numerical integration challenging, if not impossible. This leads to our first objective of this research:

Research Objective 1: *conducting a comprehensive survey of both existing and emerging methodologies for entropy estimation and optimization, with a particular emphasis on their applications in BED.*

Addressed in Chapter 2, this exploration spans from established methods like plug-in and k-NN estimators to advanced techniques such as variational estimators and entropy gradient estimation methods. Moreover, a significant part of this exploration involves linking entropy estimation and optimization principles to BED. We discover that BED objectives often align with either minimizing the entropy of posterior parameter distributions or maximizing the entropy of observed data. This insight allows for a categorization of traditional BED approaches into two distinct groups and suggests that methods developed for entropy optimization can be naturally adapted for BED, providing a foundation for

developing innovative BED methodologies.

1.1.2 Tackling high-dimensional entropy estimation

The initial part of this thesis includes a comprehensive summary of general entropy estimation methods. However, it is widely acknowledged that entropy estimation becomes increasingly challenging as the dimensionality of data increases. This challenge primarily arises from the estimation bias, which decays at a notably slow rate with respect to the sample size in high-dimensional contexts. For example, in commonly used methods such as the k-NN estimator [84], the bias decay rate is described by $O(N^{-\gamma/d})$, where N represents the sample size, d the data's dimensionality, and γ a positive constant [87, 80, 54, 160]. Consequently, most existing entropy estimation techniques struggle to effectively manage high-dimensional problems without imposing substantial assumptions about the distribution's smoothness [80]. Acknowledging these challenges, our second objective focuses on:

Research Objective 2: *developing effective entropy estimation methods capable of a faster bias decay rate under mild smoothness assumptions.*

Such approaches aim to more efficiently handle high-dimensional challenges, narrowing the gap between the actual estimation bias and its theoretical limit. We address this objective in Chapter 3 and 4. In Chapter 3, we introduce a transform-based method for high-dimensional entropy estimation. This method comprises two primary components. The first is a novel estimator based on a modified k-NN approach. This estimator is especially effective for samples that approximate a uniform distribution, showcasing a reduced estimation bias in such instances. The second component is a normalizing flow-based mapping technique designed to transform samples towards a uniform distribution. With this mapping in place, we can analytically establish the relationship between the entropies of the original and the

transformed samples. The process for achieving a more accurate estimation of entropy involves several steps. Initially, the original samples are transformed to approximate uniformity using the normalizing flow-based mapping. Following this transformation, the newly formulated estimator is applied to these transformed samples to estimate their entropy. Finally, leveraging the established analytical relationship, we backtrack to accurately estimate the entropy of the original samples. In Chapter 4, we delve into the theoretical foundations of the proposed estimators. This chapter includes a comprehensive analysis of the bias and variance bounds of the methods, indicating their reliability and the potential for quicker convergence in high-dimensional entropy estimation scenarios.

1.1.3 Advancing BED with entropy gradient estimation

After exploring methods for high-dimensional entropy estimation, we turn our attention to the utilization of entropy optimization techniques in BED. As previously discussed, BED can be conceptualized as a problem of entropy optimization. While accurately estimating entropy poses a challenge, especially in high-dimensional spaces, our primary interest lies in identifying design variables that optimize entropy values. This shift in focus leads us to a novel strategy: directly estimating the gradient of design criterion, which includes an entropy term, with respect to these design variables, and then use stochastic gradient descent to determine the most informative design. This strategy forms the cornerstone of our third and final objective:

Research Objective 3: *developing novel and efficient BED schemes leveraging entropy gradient estimation techniques.*

Our endeavor to devise innovative and efficient BED strategies leveraging entropy gradient estimation techniques is elaborated upon in Chapter 5, where we introduce two

distinct methods for estimating the Expected Information Gain (EIG) gradient. The first method, named UEEG-MCMC, utilizes posterior samples derived from Markov Chain Monte Carlo (MCMC) simulations for the EIG gradient estimation. This method has been validated as effective in various scenarios, independent of the actual EIG values. The second method, referred to as BEEG-AP, stands out for its simulation efficiency. Nevertheless, its efficacy diminishes in cases involving substantial ground-truth EIG values. Additionally, we establish a linkage with nested Monte Carlo techniques to further analyze and understand the constraints and limitations of the BEEG-AP method in such situations.

1.1.4 Discussing the contributions and future directions

The thesis concludes with a discussion chapter that serves a dual purpose. Firstly, it encapsulates the key contributions of the research, providing a comprehensive summary of the advancements made in the field of BED, entropy estimation and optimization. Secondly, the chapter sets the stage for future exploration by proposing two promising research trajectories.

The first research trajectory focuses on the development of a global optimization method for BED. This method aims to harmoniously integrate the robust local search capabilities exhibited by our newly proposed gradient-based methods with the expansive global search efficiency inherent in Bayesian Optimization (BO). Such an integration promises to enhance the precision and effectiveness of BED strategies, particularly in complex, high-dimensional scenarios.

The second research trajectory further explores the use of of entropy gradient estimation techniques in BED, with a special focus on their application to BED for implicit models. These models, known for their inherent complexity and lack of explicitly defined likelihood, introduce unique challenges in experimental design. The proposed trajectory seeks to tackle

these challenges by employing entropy gradient estimation methods that are specially tailored for implicit models, potentially opening new avenues for optimizing experimental designs for a broader spectrum of models.

1.2 Structure of the thesis

This chapter serves as the introduction. The subsequent organization of the thesis is delineated below:

Chapter 2: This chapter furnishes readers with foundational knowledge encompassing entropy, existing techniques for entropy estimation and optimization, an introduction to BED, and the intricate interconnections between BED and entropy optimization. Some of the material in this chapter has appeared in [6].

Chapter 3: This chapter introduces a transform-based approach for high-dimensional entropy estimation. The material in this chapter has appeared in [7] and [8].

Chapter 4: Delving deeper into the entropy estimators proposed in Chapter 3, this chapter elucidates rigorous theoretical underpinnings. The material in this chapter has appeared in [8].

Chapter 5: This chapter unfurls BED methodologies grounded in entropy gradient estimation. It meticulously weighs their advantages and limitations, both from theoretical and empirical perspectives. The material in this chapter has appeared in [9].

Chapter 6: Culminating the discourse, this chapter provides a recapitulation of the thesis's core contributions and sheds light on potential avenues for future research endeavors.

Chapter Two

Preliminaries

2.1 Background on Entropy

Entropy is a fundamental concept that permeates various fields of science, from physics and chemistry to economics, sociology. It is a measure of disorder, randomness, or uncertainty within a system, and its implications extend far beyond its origins in thermodynamics. The concept of entropy was first introduced in the 19th century by Rudolf Clausius, who sought to understand the behavior of heat in physical systems. Since then, entropy has evolved into a powerful and versatile concept, offering insights into the behavior of everything from molecules and particles to complex social systems and information networks. In this exploration of entropy, we will delve into its origins, its estimation and optimization methods, and its role in different disciplines, especially in Bayesian experimental design.

2.1.1 Basic Notations and Definitions

In this thesis, we exclusively focus on differential entropy (also referred to as continuous entropy). We start by introducing essential notations and definitions pertaining to differential

entropy. We represent two continuous-valued random variables as X and Y, each characterized by their respective probability density functions (PDFs) denoted as $p(\mathrm{x})$ and $q(\mathrm{y})$. We also denote their joint PDF by $p(\mathrm{x}, \mathrm{y})$, and the conditional PDF of Y given X by $p(\mathrm{y}|\mathrm{x})$.

Definition 2.1 (differential entropy)**.** *The differential entropy of X is defined as*

$$H(X) = -\int \log[p(\mathrm{x})]p(\mathrm{x})d\mathrm{x}. \tag{2.1}$$

The differential entropy provides a measure of uncertainty or information content associated with a continuous-valued random variable. Now, we extend the definition of differential entropy to the two-variable case.

Definition 2.2 (joint entropy)**.** *The joint entropy between X and Y is defined as*

$$H(X, Y) = -\int \log[p(\mathrm{x}, \mathrm{y})]p(\mathrm{x}, \mathrm{y})d\mathrm{x}d\mathrm{y}. \tag{2.2}$$

Another important concept related to entropy is the conditional entropy, which quantifies the uncertainty or information content of one random variable given the knowledge or observation of another random variable.

Definition 2.3 (conditional entropy)**.** *The conditional entropy of Y given X is defined as*

$$H(Y|X) = -\int \log[p(\mathrm{y}|\mathrm{x})]p(\mathrm{x}, \mathrm{y})d\mathrm{x}d\mathrm{y}. \tag{2.3}$$

Finally in this subsection, we introduce two relative measures in information theory: mutual information that measures the reduction in uncertainty about one variable due to the knowledge of another, and Kullback–Leibler divergence that measures how one probability distribution differs from another.

Definition 2.4 (mutual information)**.** *The mutual information between Y and X is defined as*

$$I(X; Y) = \int \log\Big[\frac{p(\mathrm{x}, \mathrm{y})}{p(\mathrm{x})q(\mathrm{y})}\Big]p(\mathrm{x}, \mathrm{y})d\mathrm{x}d\mathrm{y}. \tag{2.4}$$

Definition 2.5 (Kullback–Leibler divergence)**.** *The Kullback–Leibler divergence between X and Y is defined as*

$$D_{\mathrm{KL}}(X\|Y) = \int \log\left[\frac{p(\mathrm{x})}{q(\mathrm{x})}\right] p(\mathrm{x}) d\mathrm{x}. \tag{2.5}$$

2.1.2 The relationships between entropy and other information-theoretic quantities

In this section, we explore the relationships between entropy, mutual information and Kullback-Leibler divergence. These relationships are critical in fields such as data compression, cryptography, and communication theory, where understanding and optimizing information transmission is essential. Beyond their foundational role, these relationships also form the bedrock of our exploration into Bayesian experimental design in the following sections. Three propositions are presented to elucidate these relationships:

Proposition 2.1. *The relationships between mutual information and entropy in a two-variable system can be described by the following equations:*

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{2.6}$$

$$I(X;Y) = H(Y) - H(Y|X) \tag{2.7}$$

$$I(X;Y) = I(Y;X) \tag{2.8}$$

Remark 2.1. *The first proposition establishes basic equations that relate mutual information and entropy in a two-variable system, highlighting how mutual information can be viewed as a function of the individual, joint and conditional entropies of the variables. These relationships are further explored by applying Eq.* (2.7) *and Eq.* (2.8) *in Section 2.4.3, which are pivotal in leading to Proposition 2.4 and 2.5.*

Proposition 2.2. *The mutual information between two random variables X and Y can be expressed as the Kullback-Leibler divergence between the joint probability distribution $p(x, y)$*

and the product of the marginal probability distributions $p(\mathrm{x})q(\mathrm{y})$*. This is formally represented by:*

$$I(X;Y) = D_{\mathrm{KL}}(p(\mathrm{x},\mathrm{y})\|p(\mathrm{x})q(\mathrm{y})). \tag{2.9}$$

Remark 2.2. *The second proposition introduces the concept of Kullback-Leibler divergence as a means to express mutual information. This perspective offers a probabilistic interpretation of mutual information as a measure of divergence or 'distance' between probability distributions. This interpretation has inspired the development of advanced mutual information estimators that are based on dual representations of the KL-divergence [19], which are then tailored to address challenges in Bayesian experimental design [83].*

Proposition 2.3 (concavity of the entropy and entropy power inequality [34])**.**

1. *The entropy is concave in the PDF, i.e.,*

$$H(\lambda p(\mathrm{x}) + (1-\lambda)q(\mathrm{x})) \geq \lambda H(X) + (1-\lambda)H(Y), \tag{2.10}$$

   where $0 \leq \lambda \leq 1$*.*

2. *If* X *and* Y *are independent, then we have*

$$e^{\frac{2}{n}H(X+Y)} \geq e^{\frac{2}{n}H(X)} + e^{\frac{2}{n}H(Y)}, \tag{2.11}$$

   where n *is the dimensionality of the random variables.*

Remark 2.3. *The third proposition provides two inequalities highlighting a distinct aspect of entropy. The first inequality demonstrates the concavity of the entropy function with respect to PDFs, implying that mixing distributions tends to increase uncertainty or randomness, as measured by entropy. The second inequality, known as the Entropy Power Inequality, reveals that the entropy power of random variables, defined as* $N(\cdot) = \frac{1}{2\pi e}e^{\frac{2}{n}H(\cdot)}$*, exhibits superadditivity. This proposition forms the foundation in the proof of Proposition 2.6 discussed in Section 2.4.3.*

Together, these propositions offer a comprehensive view of how entropy interacts with and informs other key quantities in information theory, providing a foundation for more advanced studies and applications in this thesis.

2.2 Entropy Estimation

Differential entropy can be estimated via Monte Carlo (MC):

$$H(X) \approx -\frac{1}{M}\sum_{i=1}^{M} \log p(\mathrm{x}^{(i)}), \tag{2.12}$$

where $\mathrm{x}^{(i)}$ are drawn from $p(\mathrm{x})$. However, in many practices the PDF $p(\mathrm{x})$ is not known in closed-form. Therefore, alternative methods are required for this estimation task only using from independent samples of distribution. In this section, we give an overview of several approaches to addressing this challenge.

2.2.1 Nested Monte Carlo Estimators

The Nested Monte Carlo (NMC) approach can be used to estimate differential entropy for semi-implicit distributions. The semi-implicit distribution for a random variable X is defined in a hierarchical manner as

$$\mathrm{x} \sim p(\mathrm{x}|\mathrm{z}), \quad \mathrm{z} \sim p_{\mathrm{z}}(\mathrm{z}), \tag{2.13}$$

where x is a realization of X and z is a realization of the latent variable Z. For the semi-implicit distribution, the conditional PDF $p(\mathrm{x}|\mathrm{z})$ is analytically known and the PDF for the latent variable $p_{\mathrm{z}}(\mathrm{z})$ can be sampled from and allowed to be implicit. In this case, unless $p(\mathrm{x}|\mathrm{z})$ and $p_{\mathrm{z}}(\mathrm{z})$ are conjugate, the PDF for X, which is given by

$$p(\mathrm{x}) = \int p(\mathrm{x}|\mathrm{z}) p_{\mathrm{z}}(\mathrm{z}) d\mathrm{z}, \tag{2.14}$$

is often analytically intractable. Using this semi-implicit structure, each value of PDF $p(\mathrm{x}^{(i)})$ in Eq. (2.12) can be estimated via inner MC as

$$p(\mathrm{x}^{(i)}) \approx \frac{1}{N}\sum_{j=1}^{N} p(\mathrm{x}^{(i)}|\mathrm{z}^{(j)}), \tag{2.15}$$

where $\mathrm{z}^{(j)}$ are drawn from $p_{\mathrm{z}}(\mathrm{z})$. Combining Eq. (2.12) and Eq. (2.15), we obtain an estimator of $H(X)$. The theoretical results in [144, 135] show that the mean squared error of the NMC estimator decays at a rate of $O(\frac{1}{M} + \frac{1}{N})$.

2.2.2 Plug-in Estimators

A "plug-in estimator" refers to an approach where one use empirical data to estimate the PDF and then use this estimated PDF to compute the entropy. Methods such as the Parzen-Rosenblatt estimator [124, 140] and histogram estimator [63] are popular choices for the PDF estimation. After the estimated PDF $\hat{p}$ which is obtained from a set of empirical data, the differential entropy of X is then approximated by

$$H(X) \approx -\frac{1}{M}\sum_{i=1}^{M} \log \hat{p}(\mathrm{x}^{(i)}), \tag{2.16}$$

where $\{\mathrm{x}^{(i)}\}_{i=1}^{M}$ is the same set of samples used for PDF estimation or additional set of samples. Approaches to allocating samples for PDF estimation and the subsequent use of these estimated PDFs can be categorized into three groups (see [18] for a detailed review), as summarized in Table 2.1.

2.2.3 K-nearest neighbors (k-NN) entropy estimators

K-NN estimators have garnered increasing attention when compared to plug-in methods due to their advantages in three key aspects:

Method	Description	Ref.
Resubstitution estimate	Using the same set of samples for both PDF estimation and "plug-in"	[2] [79] [66]
Splitting data estimate	Using distinct sets of samples for PDF estimation and "plug-in"	[145] [61, 63, 62], [128]
Cross-validation estimate	Using leave-one-out density estimate (i.e. estimating PDF using all samples except the one employed for "plug-in")	[77] [66] [80]

Table 2.1: The summary of different types of plug-in estimators.

1. **Direct Differential Entropy Estimation**: Unlike plug-in methods, k-NN estimators directly approximate differential entropy from empirical data without the need for estimating the probability density function (PDF) as an intermediate step.

2. **Non-Specific Probability Distribution**: K-NN estimators do not assume a specific probability distribution for the data. They are non-parametric and make no prior distributional assumptions.

3. **Adaptability to Data Complexity**: K-NN estimators can adapt to the complexity of the data, offering the flexibility to capture intricate patterns and structures within the dataset.

The exemplar fixed k-NN estimator is that proposed by Kozachenko and Leonenko [84] which we refer to as the KL estimator. Let ϵ_i be twice the distance between $\mathrm{x}^{(i)}$ and its k-th nearest neighbor among the set of samples. KL estimator is given by:

$$\widehat{H}_{\mathrm{KL}}(X) = -\psi(k) + \psi(N) + \log c_d + \frac{d}{N}\sum_{i=1}^{N} \log \epsilon_i, \tag{2.17}$$

where d is the dimension of X and $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ with $\Gamma(x)$ being the Gamma function [85], and

$$c_d = \Gamma(1 + \frac{1}{p})^d / \Gamma(1 + \frac{d}{p})$$

is the volume of the d-dimensional unit ball with respect to p-norm.

The original [84] paper established the asymptotic unbiasedness for $k = 1$ while [152] obtained the same result for general k. For distributions with unbounded support, [170] proved that the bias bound decays at a rate of $O(\frac{1}{\sqrt{N}})$ for $d = 1$. [54] generalized it to higher dimensions, obtaining a bias bound of $O(N^{-\frac{1}{d}})$ up to polylogarithmic factors. For distributions compactly supported, usually densities satisfying the β-Hölder condition are considered. [21] gave a quick-and-dirty upper bound of bias, $O(N^{-\beta})$, for a simple class of univariate densities supported on $[0, 1]$ and bounded away from zero. [153] proved the bias is around $O(N^{-\frac{\beta}{d}})$ ($\beta \in (0, 2]$) for general d with some additional conditions on the boundary of support. All these works obtained a variance bound of $O(N^{-1})$.

2.2.4 Variational Estimators

Parametric entropy estimators are a class of methods used to estimate entropy by assuming a specific parametric probability distribution for the data. These estimators rely on the assumption that the data follows a particular family of probability distributions with parameters to be learned, and they use this assumed distribution to estimate the entropy. Given a family of probability distributions parameterized by ϕ, a variational approximation to the differential entropy for X can be formulated as the following optimization problem.

Theorem 2.1. *For any family of probability distributions $q_\phi(\mathrm{x})$, the differential entropy for X can be upper bounded by*

$$H(X) \leq \inf_{\phi \in \Phi} \mathbb{E}_{p(\mathrm{x})}[-\log q_\phi(\mathrm{x})], \tag{2.18}$$

where Φ is the space of possible parameter values. Moreover, the infimum bound is achieved when p belongs to the family of $q_\phi(\mathrm{x})$ for $\phi \in \Phi$.

Proof. We note that for any for $\phi \in \Phi$,

$$\mathbb{E}_{p(\mathrm{x})}[-\log q_\phi(\mathrm{x})] - H(X) = D_{\mathrm{KL}}(p \| q_\phi(\mathrm{x})) \geq 0. \tag{2.19}$$

Therefore we have

$$\inf_{\phi \in \Phi} \mathbb{E}_{p(\mathrm{x})}[-\log q_\phi(\mathrm{x})] - H(X) = \inf_{\phi \in \Phi} D_{\mathrm{KL}}(p \| q_\phi(\mathrm{x})) \geq 0, \tag{2.20}$$

which implies

$$H(X) \leq \inf_{\phi \in \Phi} \mathbb{E}_{p(\mathrm{x})}[-\log q_\phi(\mathrm{x})]. \tag{2.21}$$

Moreover, when p belongs to the family of $q_\phi(\mathrm{x})$ for $\phi \in \Phi$, we have

$$\inf_{\phi \in \Phi} D_{\mathrm{KL}}(p \| q_\phi(\mathrm{x})) = 0, \tag{2.22}$$

which implies that in this case we have

$$H(X) = \inf_{\phi \in \Phi} \mathbb{E}_{p(\mathrm{x})}[-\log q_\phi(\mathrm{x})]. \tag{2.23}$$

□

The proof of the above theorem reveals a crucial insight: the gap between the true entropy and the upper bound approximation is exactly the minimum possible KL divergence between the distribution for X and the parametric probability distribution. This underscores the central importance of developing general and expressive parametric PDF models. In this context, Normalizing Flows, as introduced in the work by Papamakarios et al. [122], emerge as a powerful tool for parametric probabilistic modeling. They offer the capability to capture complex data distributions while allowing for tractable calculations of the necessary probability densities. This makes them perfectly suited for the entropy estimation task, where accurate modeling of data distributions is paramount.

2.2.5 Applications

Entropy serves as a natural tool for quantifying the randomness and uncertainty associated with random variables of interest. Entropy-based measures find ubiquitous applications in scientific and engineering analyses, spanning fields such as physics [16], stochastic processes [110], graphs [37], biological signal analysis [52], survival analysis [38]. Consequently, the estimation of entropy holds profound significance in these domains.

2.3 Entropy Optimization

Entropy optimization is a fundamental challenge encountered across various disciplines, where the goal is to identify solutions that either maximize or minimize the entropy of a system or model. Let X be a random variable affected by a control variable $\lambda \in D$, and $p(\mathrm{x}|\lambda)$ represent the conditional probability density of X given λ. Entropy optimization can be formulated as the following maximization problem

$$\lambda^* = \arg\max_{\lambda \in D} H(X|\lambda) \tag{2.24}$$

or minimization problem

$$\lambda^* = \arg\min_{\lambda \in D} H(X|\lambda), \tag{2.25}$$

where $H(X|\lambda) = -\int \log[p(\mathrm{x}|\lambda)]p(\mathrm{x}|\lambda)d\mathrm{x}$.

In many practical applications, the random variable X does not have an explicit analytical form for its conditional probability density $p(\mathrm{x}|\lambda)$ due to the complexity of the underlying generative processes. For instance, X could be the outcome of a sampling path $x = g(\lambda, \epsilon)$, where ϵ represents source noise with a known probability density function $\pi(\epsilon)$. A common scenario where this occurs is when x is the solution of a stochastic differential equation (SDE) where λ parameterizes the drift and diffusion terms, and ϵ embodies the

stochastic driving noise. In such cases, the interaction between λ and ϵ through the nonlinear dynamics of the SDE makes it difficult to derive a closed-form expression for $p(\mathrm{x}|\lambda)$, thereby necessitating numerical methods or simulation-based approaches to understand and optimize the entropy of X.

Basically, entropy optimization methods can be categorized into two main approaches: two-stage optimization and entropy gradient estimation.

2.3.1 Two-stage Optimization

In the two-stage optimization approach, the optimization process involves two main steps:

- Stage 1: **Entropy estimation**. In this stage, the entropy of the system or model is estimated under specific control variable. This stage is often encapsulated as a callable function $\widehat{H}(X|\lambda)$, prepared to be invoked within the optimization stage. Depending on the specific setups of the system or model, we have the flexibility to choose from a range of entropy estimators, as discussed in Section 2.2.

- Stage 2: **Optimization**. Following the acquisition of the callable function from Stage 1, the optimization process proceeds to Stage 2, which is the actual optimization step. In this stage, we can leverage various general-purpose optimization methods to search for the approximate optimal control variable that either maximizes or minimizes the estimated entropy $\widehat{H}(X|\lambda)$. We conclude this subsection by listing several popular choices.

Bayesian optimization

Bayesian Optimization (BO), as introduced by Snoek et al. in their work [155], stands out as a powerful and versatile technique tailored for optimizing complex, computationally expensive functions. Its noteworthy capability to gracefully address noisy, expensive, and derivative-free objective functions renders it an excellent choice for tackling the entropy optimization problem, complementing any of the previously mentioned entropy estimators.

At its core, BO combines elements of surrogate modeling, probabilistic modeling, and decision theory to make informed decisions about where to sample the function next. This is achieved by maintaining a probabilistic surrogate model, typically a Gaussian Process (GP) [147], which provides a statistical representation of the underlying objective function. The GP model captures not only the function's values but also the associated uncertainties, allowing BO to balance exploration (sampling in uncertain regions) and exploitation (sampling in regions likely to contain the optimum).

The BO process iteratively refines its surrogate model based on the collected data, updating its beliefs about the objective function. It then employs an acquisition function, such as Expected Improvement (EI) [105] or Upper Confidence Bound (UCB) [161], to guide the selection of the next sampling point. By systematically focusing on areas that are expected to yield the most informative data, the process efficiently optimizes the objective function. Such process can be summarized as Algorithm 1.

Algorithm 1 Pseudo-code for BO

Require: N(total number of calls to objective function), GP model trained on initial dataset $n = n_0 : \left(\mathbf{x}^{(i)}, y_i\right)_{1 \leq i \leq n}$ where $\mathbf{x}^{(i)}$ are input parameters at iteration i and y_i is objective function value corresponding to $\mathbf{x}^{(i)}$

1: **while** $n \leq N$ **do**
2: Choose $\mathbf{x}^{(n+1)}$ that maximizes the acquisition function.
3: Update the GP model by conditioning on $\left(\mathbf{x}^{(n+1)}, y_{n+1}\right)$
4: $n \leftarrow n + 1$
5: **end while**

Simultaneous perturbation stochastic approximation

Simultaneous Perturbation Stochastic Approximation (SPSA) is a powerful optimization algorithm used to find the minimum or maximum of an objective function [1], especially when the function is noisy, black-box, or computationally expensive to evaluate. SPSA is particularly well-suited for optimization problems where gradient information is unavailable or too costly to compute.

SPSA was introduced by James Spall [158] as an iterative, gradient-free optimization technique. It belongs to the class of stochastic approximation algorithms, which aim to estimate the gradient of the objective function using noisy measurements. What sets SPSA apart is its simplicity and efficiency.

The key idea behind SPSA is to use simultaneous perturbations of the input variables to estimate the gradient. Instead of computing the gradient directly, SPSA perturbs each

[1] For the SPSA algorithm to be effectively applied, the assumption regarding the objective function is that it must be differentiable almost everywhere within the domain where optimization is conducted. This requirement is crucial because the algorithm approximates the gradient by evaluating the objective function at perturbed points around the current estimate.

input variable in two opposite directions simultaneously and evaluates the objective function at these perturbed points. By analyzing the differences in function values, SPSA approximates the gradient in a noisy environment. The control variable is then updated as

$$\lambda_{k+1} = \lambda_k - a_k \mathbf{g}_k(\lambda_k), \tag{2.26}$$

$$\mathbf{g}_k\left(\lambda_k\right) = \frac{\widehat{H}\left(X|\lambda_k + c_k\mathbf{\Delta}_k\right) - \widehat{H}\left(X|\lambda_k - c_k\mathbf{\Delta}_k\right)}{2c_k} \begin{bmatrix} \Delta_{k,1}^{-1} \\ \Delta_{k,2}^{-1} \\ \vdots \\ \Delta_{k,n_d}^{-1} \end{bmatrix}, \tag{2.27}$$

where k is the iteration number,

$$a_k = \frac{a}{(A+k+1)^\alpha}, \quad c_k = \frac{c}{(k+1)^\gamma}, \tag{2.28}$$

and a, A, α, c and γ are hyper-parameters of the algorithm (see [159] for recommended values). $\Delta_{k,i}$ is the i-th component of the Δ_k vector and is independently generated from a zero-mean probability distribution with finite inverse moments. A simple and valid choice is $\Delta_{k,i} \sim$ Bernoulli(0.5).

One of the notable advantages of SPSA is its ability to converge quickly, often requiring fewer function evaluations compared to traditional gradient-based methods. This makes it suitable for scenarios where function evaluations are expensive or where analytical gradients are not available.

Nelder-Mead nonlinear simplex

Nelder-Mead nonlinear simplex (NMNS) [112] is also a popular optimization algorithm used to find the minimum or maximum of a black-box objective function whose gradient information is not available. Slight adjustments to the algorithm's parameters have the potential to enhance its optimization capabilities when dealing with noisy functions [14].

At its core, NMS algorithm uses a geometric shape known as a simplex, which is a multi-dimensional analog of a triangle in two dimensions or a tetrahedron in three dimensions. The simplex is iteratively modified to explore the parameter space of the objective function and seek an optimal solution. Here's a high-level overview of how the NMS algorithm works:

- **Initialization**: The algorithm starts with an initial set of points (a simplex) in the parameter space. A simplex is a geometric shape that can be a triangle in 2D, a tetrahedron in 3D, or a higher-dimensional polytope in higher dimensions.
- **Reflection**: The algorithm reflects the worst point (highest function value) through the centroid of the remaining points to produce a new candidate point.
- **Expansion**: If the reflected point is better (has a lower function value) than the second-worst point, the algorithm attempts to expand further along that direction by reflecting again.
- **Contraction**: If the reflected point is not better than the second-worst point but better than the worst point, the algorithm contracts the reflected point toward the centroid to explore the space in that direction.
- **Shrink**: If none of the above operations lead to a better point, the algorithm contracts the entire simplex toward the best point (lowest function value), reducing the size of the simplex.
- **Termination Criteria**: The algorithm iteratively performs the above steps until a termination criterion is met. Common termination criteria include a maximum number of iterations, a small change in the function value, or a small change in the size of the simplex.

In contrast to contemporary optimization techniques, the NMS heuristic has the potential to reach a non-stationary point, unless the problem meets stricter conditions than

those required by modern methods [130]. Despite this limitation, the NMS algorithm is a valuable optimization tool and is widely used in various fields, including engineering, physics, economics, and machine learning, for optimizing a wide range of objective functions.

2.3.2 Entropy Gradient Estimation

Two-stage optimization methods often require precise entropy estimators to ensure the validity of the optimization process. However, our primary concern is not the precise numerical value of the entropy associated with the system or model. Instead, our focus lies on determining the optimal control variable λ, which either maximizes or minimizes the entropy. With this fresh perspective in mind, a more direct approach to entropy optimization involves estimating the gradient of entropy w.r.t. λ. Subsequently, this gradient information can be leveraged in optimization techniques like stochastic gradient descent or ascent algorithms to efficiently navigate the parameter space and identify the optimal value of λ.

We start this subsection by analyzing the difficulty in estimating the gradient of entropy $\nabla_\lambda H(X|\lambda)$. Suppose that the realization of X is obtained by a sampling path $\mathrm{x} = g(\lambda, \epsilon)$, where ϵ is the source noise with a PDF $\pi(\epsilon)$. By reparameterization trick [106], this gradient can be written as

$$\begin{aligned}\nabla_\lambda H(X|\lambda) &= -\nabla_\lambda \mathbb{E}_{\pi(\epsilon)}[\log p(g(\lambda,\epsilon)|\lambda)] \\ &= -\mathbb{E}_{\pi(\epsilon)}[\nabla_\lambda \log p(g(\lambda,\epsilon)|\lambda)] \\ &= -\mathbb{E}_{\pi(\epsilon)}[\nabla_\lambda \log p(\mathrm{x}|\lambda)|_{\mathrm{x}=g(\lambda,\epsilon)} + \nabla_\mathrm{x} \log p(\mathrm{x}|\lambda)|_{\mathrm{x}=g(\lambda,\epsilon)} \nabla_\lambda g(\lambda,\epsilon)],\end{aligned} \tag{2.29}$$

where first term in the last line is zero [176]. Since $p(\mathrm{x}|\lambda)$ usually does not have an analytical form as we mentioned before, score estimation techniques are required to approximate $\nabla_\mathrm{x} \log p(\mathrm{x}|\lambda)$.

Nested Monte Carlo methods

Recall Section 2.2.1, nested Monte Carlo methods can be used to address the intractability of $p(\mathrm{x}|\lambda)$ if X has a semi-implicit probability distribution. The semi-implicit distribution is defined as

$$\mathrm{x} \sim p(\mathrm{x}|\mathrm{z}, \lambda), \quad \mathrm{z} \sim p_{\mathrm{z}}(\mathrm{z}), \tag{2.30}$$

where x is a realization of X and z is a realization of the latent variable Z. Employing the tractability of $p(\mathrm{x}|\mathrm{z}, \lambda)$, we obtain a lower bound and upper bound estimations of $p(\mathrm{x}|\lambda)$ respectively.

Theorem 2.2. *The entropy $H(X|\lambda)$ can be lower bounded by*

$$H(X|\lambda) \geq \widehat{H}_{\mathrm{LB}}(X|\lambda) = -\mathbb{E}\Big[\log \frac{1}{N+1} \sum_{i=0}^{N} p(\mathrm{x}|\mathrm{z}^{(i)}, \lambda)\Big], \tag{2.31}$$

and upper bounded by

$$H(X|\lambda) \leq \widehat{H}_{\mathrm{UB}}(X|\lambda) = -\mathbb{E}\Big[\log \frac{1}{N} \sum_{i=1}^{N} p(\mathrm{x}|\mathrm{z}^{(i)}, \lambda)\Big], \tag{2.32}$$

where the expectation is taken over $\mathrm{x}, \mathrm{z}^{(0:N)} \sim p(\mathrm{x}|\mathrm{z}^{(0)}, \lambda) \prod_{i=1}^{N} p_{\mathrm{z}}(\mathrm{z}^{(i)})$.

Proof. The proof closely resembles the one presented in Lemma 1 of the paper by Foster et al. [49] and Theorem 1 in the work by Foster et al. [48]. As such, we choose to omit it here. □

Given that these two bounds do not incorporate any computationally challenging terms, we can focus on optimizing them by gradient ascent or descent as a means to seek the near optimal solutions for entropy optimization problems. Specifically, when aiming for entropy maximization, our objective is to maximize the lower bound; conversely, when striving for entropy minimization, our goal is to minimize the upper bound. Similar methods have been used for variational inference [178], Bayesian experimental design [48] and mutual information estimation [129].

Plug-in methods with density estimation

A naïve approach to estimating the intractable score function is by employing the plug-in density estimators. More specifically, we can estimate the intractable density $\hat{p}(\mathrm{x}|\lambda) \approx p(\mathrm{x}|\lambda)$ first and then use it to approximate the score function by $\nabla_{\mathrm{x}} \log \hat{p}(\mathrm{x}|\lambda) = \nabla_{\mathrm{x}} \log p(\mathrm{x}|\lambda)$. See Section 2.2.2 for various choices of plug-in estimators.

Stein gradient estimator

The Stein gradient estimator, first introduced by Li & Turner [90], provides a direct approximation to the intractable score function. For convenience, we omit the control variable and use $q(\mathrm{x})$ to represent the marginal likelihood $p(\mathrm{x}|\lambda)$. Given i.i.d. samples $\mathrm{x}^{(i)} \in \mathbb{R}^{d\times 1}, i = 1, ..., N$ drawn from $q(\mathrm{x})$, the stein gradient estimator aims to approximate the matrix of the realizations of score function $\mathbf{G} = (\mathbf{s}(\mathrm{x}^{(1)}), ..., \mathbf{s}(\mathrm{x}^{(N)}))^{\mathrm{T}} \in \mathbb{R}^{N\times d}$ with $\mathbf{s}(\mathrm{x}) = \nabla_{\mathrm{x}} \log q(\mathrm{x})$. The resulting approximation is denoted as $\widehat{\mathbf{G}} = (\hat{\mathbf{s}}(\mathrm{x}^{(1)}), ..., \hat{\mathbf{s}}(\mathrm{x}^{(N)}))^{\mathrm{T}} \in \mathbb{R}^{N\times d}$.

This method is motivated by the following Stein's identity.

Theorem 2.3 ([162, 90, 92])**.** *Suppose $q(\mathrm{x})$ is a smooth density function supported on $\mathbb{R}^d$ and $\mathbf{h}(\mathrm{x}) = (h_1(\mathrm{x}), ..., h_{d'}(\mathrm{x}))^{\mathrm{T}} \in \mathbb{R}^{d'\times 1}$ is a smooth vector-valued function satisfying the boundary condition:*

$$\lim_{\mathrm{x}\to\infty} q(\mathrm{x})\mathbf{h}(\mathrm{x}) = 0. \tag{2.33}$$

Then we have

$$\mathbb{E}_q[\mathbf{h}(\mathrm{x})\mathbf{s}(\mathrm{x})^{\mathrm{T}} + \nabla_{\mathrm{x}}\mathbf{h}(\mathrm{x})] = 0, \tag{2.34}$$

where $\nabla_{\mathrm{x}}\mathbf{h}(\mathrm{x}) = (\nabla_{\mathrm{x}} h_1(\mathrm{x}), ..., \nabla_{\mathrm{x}} h_{d'}(\mathrm{x}))^{\mathrm{T}} \in \mathbb{R}^{d'\times d}$.

We denote Monte Carlo estimate of the expectation on the left hand side of Stein's

identity as

$$\frac{1}{N}\mathbf{H}\mathbf{G} + \overline{\nabla_{\mathrm{x}}\mathbf{h}} = \frac{1}{N}\sum_{i=i}^{N}\Big(\mathbf{h}(\mathrm{x}^{(i)})\mathbf{s}(\mathrm{x}^{(i)})^{\mathrm{T}} + \nabla_{\mathrm{x}}\mathbf{h}(\mathrm{x}^{(i)})\Big), \tag{2.35}$$

where $\mathbf{H} = (\mathbf{h}(\mathrm{x}^{(1)}), ..., \mathbf{h}(\mathrm{x}^{(N)})) \in \mathbb{R}^{d' \times N}$ and $\overline{\nabla_{\mathrm{x}}\mathbf{h}} = \frac{1}{N}\sum_{i=i}^{N}\nabla_{\mathrm{x}}\mathbf{h}(\mathrm{x}^{(i)}) \in \mathbb{R}^{d' \times d}$. Finally, minimizing the sum of squared residuals with an l_2 regularization term yields an approximation to $\mathbf{G}$:

$$\begin{aligned}\widehat{\mathbf{G}}_V^{\mathrm{Stein}} &= \underset{\widehat{\mathbf{G}} \in \mathbb{R}^{N \times d}}{\arg\min} \left\| \frac{1}{N}\mathbf{H}\widehat{\mathbf{G}} + \overline{\nabla_{\mathrm{x}}\mathbf{h}} \right\|_F^2 + \frac{\eta}{N^2}\|\widehat{\mathbf{G}}\|_F^2 \\ &= -(\mathbf{K} + \eta \boldsymbol{I})^{-1}\langle \nabla, \mathbf{K} \rangle,\end{aligned} \tag{2.36}$$

where $\|\cdot\|$ represents the Frobenius norm, $\eta \geq 0$, $\mathbf{K} = \mathbf{H}^{\mathrm{T}}\mathbf{H}$, $\mathbf{K}_{ij} = \mathcal{K}\left(\mathrm{x}^{(i)}, \mathrm{x}^{(j)}\right) = \mathbf{h}(\mathrm{x}^{(i)})^{\mathrm{T}}\mathbf{h}(\mathrm{x}^{(j)})$, $\langle \nabla, \mathbf{K} \rangle = N\mathbf{H}^{\mathrm{T}}\overline{\nabla_{\mathrm{x}}\mathbf{h}}$, $\langle \nabla, \mathbf{K} \rangle_{ij} = \sum_{k=1}^{N}\nabla_{\mathrm{x}_j^{(k)}}\mathcal{K}\left(\mathrm{x}^{(i)}, \mathrm{x}^{(k)}\right)$. In practice, one typically chooses $\mathcal{K}$ as the RBF kernel [173], in which case $\mathbf{h}(\mathrm{x}) = \mathcal{K}(\mathrm{x}, \cdot)$ and $d' = +\infty$.

Score matching methods

Score matching provides parametric methods for score function estimation. Let $\hat{\mathbf{s}}(\mathrm{x}; \theta)$ be a parametric approximation to the true score function $\mathbf{s}(\mathrm{x}) = \nabla_{\mathrm{x}} \log q(\mathrm{x})$, where $\theta \in \Theta$ are the model parameters. Score matching aims to minimize the expected l_2 error

$$L(\theta) = \frac{1}{2}\mathbb{E}_q[\|\hat{\mathbf{s}}(\mathrm{x}; \theta) - \mathbf{s}(\mathrm{x})\|_2^2]. \tag{2.37}$$

However, Eq. (2.37) can not be directly optimized as the true score function $\mathbf{s}(\mathrm{x})$ is not accessible in closed-form. Several approaches have been proposed to address this challenge.

If X has a semi-implicit probability distribution, i.e., its density conditional on a latent variable $q(\mathrm{x}|\mathrm{z})$ is known, the objective in Eq. (2.37) is equivalent to

$$L_{\mathrm{D}}(\theta) = \frac{1}{2}\mathbb{E}_{p_{\mathrm{z}}(\mathrm{z})q(\mathrm{x}|z)}[\|\hat{\mathbf{s}}(\mathrm{x}; \theta) - \nabla_{\mathrm{x}} q(\mathrm{x}|\mathrm{z})\|_2^2]. \tag{2.38}$$

This variant of score matching is known as **denoising score matching** [174].

Another variant, introduced by Hyvärinen [75], offers a different reformulation of the objective as follows:

$$L_{\mathrm{H}}(\theta) = \mathbb{E}_q[\mathrm{tr}(\nabla_{\mathrm{x}}\hat{\mathbf{s}}(\mathrm{x};\theta)) + \frac{1}{2}\|\hat{\mathbf{s}}(\mathrm{x};\theta)\|_2^2]. \tag{2.39}$$

We refer to this method as **Hyvärinen score matching**. It's important to note that while this method does not require either an explicit or semi-implicit probability distribution for X, it does come with the limitation that the computational cost of computing the trace of the Hessian matrix grows linearly with the dimensionality of X. Consequently, Hyvärinen score matching is often more suitable for simple, shallow models or situations involving low-dimensional data.

To mitigate the above difficulty, Song et al. [157] have proposed the **sliced score matching**. This approach bypasses the computation of the trace of the Hessian matrix by projecting the scores onto random vectors before comparing them, which yields the following objective:

$$L_{\mathrm{S}}(\theta) = \mathbb{E}_{p_{\mathbf{v}}}\mathbb{E}_q[\mathbf{v}^{\mathrm{T}}\nabla_{\mathrm{x}}\hat{\mathbf{s}}(\mathrm{x};\theta)\mathbf{v} + \frac{1}{2}(\mathbf{v}^{\mathrm{T}}\hat{\mathbf{s}}(\mathrm{x};\theta))_2^2], \tag{2.40}$$

where $\mathbf{v} \sim p_{\mathbf{v}}$ is an independent random direction that satisfies $\mathbb{E}_{p_{\mathbf{v}}}[\mathbf{v}^{\mathrm{T}}\mathbf{v}] > 0$ and $\mathbb{E}_{p_{\mathbf{v}}}[\|\mathbf{v}\|_2^2] < \infty$.

Once we get θ^* by optimizing the score matching objective, $\hat{\mathbf{s}}(\mathrm{x};\theta^*)$ can be used to estimate the intractable score function $\mathbf{s}(\mathrm{x})$.

2.3.3 Applications

Implicit variational inference

Variational inference (VI) is employed when we want to estimate the posterior distribution of latent variables Z in a probabilistic model, given observed data X. Instead of directly

calculating the exact posterior distribution, VI seeks to approximate it with a parameterized distribution $q(\mathrm{z}|\psi)$ that can be directly sampled from. The variational parameter ψ is optimized by maximizing the following evidence lower bound (ELBO) [24]

$$\mathrm{ELBO} = \mathbb{E}_{z\sim q(\mathrm{z}|\psi)}[\log p(\mathrm{x}, \mathrm{z})] + H(Z|\psi), \tag{2.41}$$

where $H(Z|\psi) = -\mathbb{E}_{\mathrm{z}\sim q(\mathrm{z}|\psi)}[\log q(\mathrm{z}|\psi)]$.

Traditional VI typically chooses $q(\mathrm{z}|\psi)$ from a family of distributions that are tractable [138]. While this choice enables efficient optimization for $q(\mathrm{z}|\psi)$, it may impose limitations on its expressive power. Recent developments in variational inference [178, 166, 179] have explored implicit models for $q(\mathrm{z}|\psi)$ which are more flexible and expressive but come at the cost of tractability. Concretely, rather than treating ψ as the parameter of a tractable density, implicit VI regards it as the parameter of the tractable sampling path that generates $\mathrm{z} \sim q(\mathrm{z}|\psi)$, and the sampling process proceeds as

$$\mathrm{z} = f(\epsilon, \psi), \quad \epsilon \sim \pi(\epsilon), \tag{2.42}$$

where f is a tractable mapping and π is the base distribution of the input noise. The generator of a GAN [58] is a typical example of such implicit model. The entropy optimization techniques mentioned in this section provide attractive solutions for maximizing ELBO with implicit posterior models, which enable more flexible options for VI.

Maximum entropy modeling with implicit distributions

Maximum entropy (ME) modeling is a probabilistic modeling approach used to estimate probability distributions from incomplete data. The central idea behind ME modeling - known as the ME principle [78] - is to find the most uniform (the maximum entropy) probability distribution that satisfies a set of constraints derived from the data. We denote $q(\mathrm{z}|\psi)$ as the parametric model of the probability distribution for a random variable Z. Then a typical

ME modeling problem can be formulated as

$$\begin{aligned} \underset{\psi}{\text{maximize}} \quad & H(Z|\psi) \\ \text{subject to} \quad & \mathbb{E}_{\mathrm{z}\sim q(\mathrm{z}|\psi)}[T(\mathrm{z})] = 0, \end{aligned} \tag{2.43}$$

where $H(Z|\psi) = -\mathbb{E}_{\mathrm{z}\sim q(\mathrm{z}|\psi)}[\log q(\mathrm{z}|\psi)]$ and $T(\mathrm{z}) = (T_1(\mathrm{z}), ..., T_m(\mathrm{z})) \in \mathbb{R}^m$ is the vector of known statistics. The above constrained optimization problem can be transformed into a series of unconstrained sub-problems through methods like the penalty method and the augmented Lagrangian method. These unconstrained sub-problems are solved iteratively. In penalty method, the unconstrained sub-problem to be solved in each iteration can be expressed as:

$$\underset{\psi}{\text{minimize}} \quad -H(Z|\psi) + \frac{\sigma}{2}\|\mathbb{E}_{\mathrm{z}\sim q(\mathrm{z}|\psi)}[T(\mathrm{z})]\|_2^2, \tag{2.44}$$

while the augmented Lagrangian method uses the following sub-objective in each iteration:

$$\underset{\psi}{\text{minimize}} \quad -H(Z|\psi) + \lambda^{\mathrm{T}}\mathbb{E}_{\mathrm{z}\sim q(\mathrm{z}|\psi)}[T(\mathrm{z})] + \frac{\mu}{2}\|\mathbb{E}_{\mathrm{z}\sim q(\mathrm{z}|\psi)}[T(\mathrm{z})]\|_2^2, \tag{2.45}$$

where σ, λ and μ are updated after each iteration.

The ME modeling using tractable parametric models has been well explored [127, 123, 93]. With gradient optimization methods in this section, these parametric models can be extended to implicit distributions [91], leading to more flexible modeling.

Entropy-regularized GANs

Generative Adversarial Networks (GANs) [58] have seen remarkable success in generating realistic data, but they also face several challenges that researchers are actively working to solve. A prominent challenge within GANs is mode collapse, where the generator only produces a limited set of similar samples while ignores other modes in the data distribution. To mitigate this issue, a common approach is to add the neg-entropy of the generator as a regularization term to the generator's loss [90, 39]. Specifically, assume π is the base

distribution of the input noise, $q(\mathrm{z}|\psi)$ is the implicit probability distribution of the generator $\mathrm{z} = f(\epsilon, \psi), \epsilon \sim \pi(\epsilon)$ and $\mathcal{J}_{\mathrm{gen}}(\psi)$ is the original generator's loss. Then the new generator's loss is given by

$$\tilde{\mathcal{J}}_{\mathrm{gen}}(\psi) = \mathcal{J}_{\mathrm{gen}}(\psi) - \lambda H(Z|\psi), \tag{2.46}$$

where $H(Z|\psi) = -\mathbb{E}_{\mathrm{z}\sim q(\mathrm{z}|\psi)}[\log q(\mathrm{z}|\psi)]$ and λ is the hyper-parameter. Since the neg-entropy is the only intractable term, the entropy optimization techniques in this section can be naturally applied to optimize the new generator's loss.

Maximum Entropy reinforcement learning

Entropy regularization has also found applications in the field of reinforcement learning, resulting in the development of a framework known as the maximum entropy reinforcement learning (MERL) [185, 168, 137, 64]. By augmenting the original objective with the entropy of the policy, MERL encourages the policy to explore widely and capture multiple modes. To illustrate how entropy optimization techniques are applied in this context, we consider the soft actor-critic (SAC) algorithm proposed by Haarnoja et al. [64] as an exemplar. In SAC, the policy parameters ψ can be learned by minimizing objective:

$$\mathcal{J}(\psi) = -\mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\psi(\cdot|s)}[Q(s,a) - \log Z(s)] - \mathbb{E}_{s\sim\mathcal{D}}[H(\pi_\psi(\cdot|s))], \tag{2.47}$$

where s and a represent the state and the action respectively within the framework of reinforcement learning, $\mathcal{D}$ represents the distribution of states that have been sampled previously and Q is the parameterized soft Q-function (defined in [64]) normalized by Z. Note that $\log Z(s)$ does not depend on ψ and can be directly discarded. The first term in Eq. (2.47) is thus tractable. However, it's important to be aware that the second term in Eq. (2.47), which involves the entropy of the policy, typically becomes intractable when we construct the policy distribution $\pi_\psi(\cdot|s)$ implicitly through a sampling process such as:

$$a = f(\epsilon, \psi), \quad \epsilon \sim \pi(\epsilon), \tag{2.48}$$

where f is the parameterized mapping and π is the base distribution of the input noise. In such cases, we would have to resort to the entropy optimization methods for policy training.

Bayesian experimental design with the criterion of expected information gain

Bayesian experimental design (BED) with the criterion of expected information gain (EIG) aims to search for the optimal design variables that maximize the information gained from collected data. In Section 2.4, we will provide an in-depth explanation of the background and principles underlying BED. Simply put, the design variables λ in BED can be learned by maximizing the following EIG:

$$\mathrm{EIG}(\lambda) = \mathbb{E}_{\pi_\theta(\theta) l(y|\theta,\lambda)}[\log l(y|\theta,\lambda)] + H(Y|\lambda), \tag{2.49}$$

where $H(Y|\lambda) = -\mathbb{E}_{p(y|\lambda)}[\log p(y|\lambda)]$. In general BED setups, the parameter prior $\pi_\theta(\theta)$ and likelihood function $l(y|\theta,\lambda)$ are analytically known, rendering the first term in Eq. (2.49) computationally feasible. However, the marginal distribution of the observation $p(y|\lambda) = \mathbb{E}_{\pi_\theta(\theta)}[l(y|\theta,\lambda)]$ usually does not have a closed-form expression, making the second term in Eq. (2.49) (the entropy term) computationally challenging to handle. By leveraging entropy optimization, practitioners can overcome the complexities associated with intractable entropy term and enhance the efficiency of BED in scenarios where such challenges arise.

2.4 Bayesian Experimental Design

In this thesis, Bayesian experimental design (BED) serves as a compelling application scenario for entropy estimation and optimization. Furthermore, BED is a fast growing area of research in its own right. Within the confines of this study, we place significant emphasis on BED as another central topic. In this section, we offer a comprehensive overview of the foundational

concepts and methodologies involved in BED. In Section 2.4.1, we embark on a survey of the model setups, coupled with an exploration of the background on Bayesian statistics and posterior inference. We then turn to discuss the diverse selection of design criteria, commonly referred to as the utility functions, in Section 2.4.2. The intricate connections between BED and entropy optimization are explored in Section 2.4.3. Finally, Section 2.4.4 offers an extensive review of the existing literature on BED methodologies, providing a comprehensive understanding of the current state of research in this field.

2.4.1 Parametric statistical models and Bayesian inference

Before we delve into the modern Bayesian methodologies for optimal experimental design, it is crucial to establish a solid foundation by introducing two fundamental components that constitute the bedrock of BED. Firstly, we give an introduction of the parametric statistical models that describe the natural processes under investigation in experiments. Following this, we acquaint readers with the foundational principles of Bayesian inference, which serve as the basis for constructing design criteria.

Parametric statistical models

In the realms of natural sciences, engineering, and social sciences, the utilization of mathematical models is extensively prevalent. These models are instrumental in facilitating a deeper comprehension and enabling precise predictions regarding the behavior of systems under study. The spectrum of these models is broad, encompassing varieties such as dynamical systems, statistical models, differential equations, and game-theoretic models. A pivotal aspect in the construction of these models is the incorporation of systemic and observed noise, which is a standard practice aimed at accounting for the inherent uncertainty present within the system. This inclusion effectively categorizes these diverse models under the umbrella of

statistical modeling.

The development of a statistical model typically encompasses two fundamental components: firstly, the determination of the model's structure, which is predominantly guided by expert knowledge within the specific domain of study. This stage is crucial as it lays down the theoretical framework upon which the model operates. Secondly, the estimation of the model's parameters, represented as θ. This step is critical for ensuring the model's accuracy in representing and predicting system behaviors. Usually, these parameters are determined and refined from the empirical data, denoted as y, which are gathered post-experimentation.

Furthermore, such statistical models often encompass parameters that can be controlled during experimental processes, such as experimental temperatures, pressure, sampling times, and locations, commonly referred to as design variables, denoted as λ. The selection of these design variables is critical, as it influences the efficacy of parameter identification within the model. The data collected under different experimental conditions can vary in the extent of information they provide about the model parameters. For instance, inappropriately low experimental temperatures might inhibit chemical reactions, yielding data that are not conducive to parameter identification. In such scenarios, the data collected offers minimal amount of information for parameter estimation. Consequently, the primary objective of BED is to identify the most informative set of design variables that facilitate optimal parameter identification.

In this context, the parametric statistical model formulated are typically represented as a probability distribution with the following conditional probability density function:

$$l(y|\theta, \lambda). \tag{2.50}$$

This is often accompanied by an analytically tractable sampling path:

$$y = g(\theta, \epsilon, \lambda), \tag{2.51}$$

where $\lambda \in \mathcal{D}$ represents the design variables that can be controlled by users, θ represents the parameters to be inferred from the observed data y and ϵ denotes the base model noises generated from a known distribution $\pi_\epsilon(\epsilon)$.

A quintessential example of such a parametric statistical model is a linear model incorporating Gaussian observation noise. This model assumes the empirical data to follow a Gaussian distribution $l(y|\theta, \lambda) = \text{Normpdf}(y; \theta^{\text{T}}\lambda, \sigma^2\mathbb{I})$. Here, σ denotes the standard deviation of the observation noise. The corresponding sampling path for this model is described by $g(\theta, \epsilon, \lambda) = \theta^{\text{T}}\lambda + \epsilon$, , with ϵ following a normal distribution $\epsilon \sim \mathbb{N}(0, \sigma^2\mathbb{I})$. This simple yet illustrative example serves as a fundamental reference in the broader context of parametric statistical modeling.

Background on Bayesian inference

Frequentist inference typically provides a single point estimate for the parameters of interest, often computed using methods like maximum likelihood estimation. On the other hand, Bayesian methods treat these parameters as random variables and provide probability distributions, specifically the posterior distribution. This distribution encompasses a spectrum of plausible parameter values, enabling the quantification of uncertainty in the inference process. In the context of BED, this ability to quantify uncertainty is paramount, as the design criteria in BED are fundamentally centered around reducing prediction uncertainty related to these parameters.

In Bayesian statistics, a prior distribution $\pi_\theta(\theta)$ is formulated to represent our initial knowledge or beliefs about the parameters of the parametric statistical model. This prior distribution quantifies the degree of belief in different values of the parameter θ before any

data is observed. Then, the posterior distribution is defined via Bayes' theorem

$$q(\theta|y,\lambda) = \frac{\pi_\theta(\theta)l(y|\theta,\lambda)}{p(y|\lambda)}, \tag{2.52}$$

where $p(y|\lambda) = \int \pi_\theta(\theta)l(y|\theta,\lambda)d\theta$ is the marginal distribution of observed data. The main difficulty in computing posterior distribution lies in the computational intractability of $p(y|\lambda)$. For explicit models where the likelihood functions are analytically known, we note that $q(\theta|y,\lambda)$ is also analytically known up to a constant, i.e., $q(\theta|y,\lambda) \propto \pi_\theta(\theta)l(y|\theta,\lambda)$. In these cases, importance sampling (IS) [167], Markov Chain Monte Carlo (MCMC) [5] and variational inference (VI) [50, 24] are popular methods. Bayesian inference for implicit models where the likelihood functions can not be directly evaluated has become a growing research topic known as likelihood-free inference. One popular class of work within this domain is Approximate Bayesian computation (ABC) [15, 154] methods. More recently, the conditional density models constructed with artificial neural networks (ANN) provide another class of likelihood-free inference methods, including SNPE [119, 99, 60], SNL [121] and SRE[72, 43].

2.4.2 Utility Functions

The utility functions play a critical role in defining the quality of experimental designs. Specifically, a utility function $u(\lambda, y)$ quantifies the worth of a specific design λ of the experiment when it results in an observed data y. As experimental designs must be chosen before conducting the experiment and observing any data, the BED aims to maximize the expected utility function $U(d)$, which is the expectation of the utility function $u(\lambda, y)$ taken over the distribution of observed data $p(y|\lambda)$:

$$\max_{\lambda\in\mathcal{D}} U(\lambda) = \mathbb{E}_{y\sim p(y|\lambda)}[u(\lambda, y)]. \tag{2.53}$$

The choice of utility functions is heavily influenced by the specific objectives of the experiments. Common objectives within BED include but not limit to minimizing bias,

decreasing random variation, enhancing the precision of parameter estimates (or another relevant measure), making predictions about future observations, and model discrimination. The focus of this thesis is primarily on achieving highly accurate parameter estimation. To achieve this objective, the utility functions are typically formulated as functionals of the posterior distribution, a concept detailed in [142]. Within the scope of this thesis, we delve into two particularly significant utility functions, namely, the *Bayesian D-posterior precision utility* and the *Kullback–Leibler divergence based utility*.

Bayesian D-posterior precision utility

A well-known example of such a utility function is the one referred to as the Bayesian D-posterior precision [143]:

$$u_D(\lambda, y) = \frac{1}{\det(\mathrm{cov}(\theta|y, \lambda))}, \tag{2.54}$$

where $\mathrm{cov}(\theta|y, \lambda)$ represents the posterior covariance matrix and det computes the determinant of the matrix. Such utility function tends to result in experimental designs that yield small posterior covariance. As the Bayesian D-posterior precision can be straightforwardly estimated using the posterior computation methods discussed in the previous subsection, it has found widespread application in BED literature [42, 65, 36]. However, employing this utility function may not be suitable when the posterior distributions exhibit substantial deviations from Gaussian characteristics, such as when they are multi-modal. In such instances, the covariance alone may not effectively capture the uncertainty of the posterior.

Kullback–Leibler divergence based utility

The above issue can be alleviated by another commonly used utility function - the Kullback–Leibler divergence (KLD) between the posterior and prior distributions:

$$u_{\mathrm{KLD}}(\lambda, y) = \mathbb{E}_{q(\theta|y,\lambda)}[\log q(\theta|y, \lambda) - \log \pi_\theta(\theta)]. \tag{2.55}$$

This utility function measures the information gain about the parameters obtained from an experiment with design λ and observed data y. It is worth noting that the resulting expected utility function $U_{\mathrm{KLD}}(\lambda) = \mathbb{E}_{y\sim p(y|\lambda)}[u_{\mathrm{KLD}}(\lambda, y)]$ is equivalent to the mutual information [33] between θ and y under design λ in information theory community. Unlike the Bayesian D-posterior precision, the KLD based utility function can effectively measure the information gain for posterior distributions exhibiting either linear or nonlinear dependencies between variables. This flexibility allows it to handle a wide range of posterior distributions, including multi-modal ones. Furthermore, research in [118] has demonstrated that adaptive design of experiments based on the KLD based utility function can lead to consistent and efficient parameter estimates under mild modelling conditions. Despite these desirable properties, it is worth noting that the KLD based utility function presents certain computational challenges due to its double intractability: 1. Approximations are often necessary for the posterior distributions unless the prior distribution and likelihood function possess conjugate properties. 2. The integral depicted in Eq. (2.55) typically lacks a closed-form solution. As a result, BED that relies on the KLD based utility used to be confined to specific scenarios in which either analytical solutions or simplifying assumptions were required to make the computation of the KLD feasible [25, 89]. In the following subsections, we will delve into more recent approaches that extend the utility of KLD-based BED to a broader range of applications.

Limitations of Bayesian D-posterior precision utility vs. KLD based utility: an illustrative example

In this analysis, we revisit the toy problem outlined in [6] to underscore the limitations inherent to the Bayesian D-posterior precision, especially when dealing with strongly non-Gaussian posteriors. The generative model in question is represented as

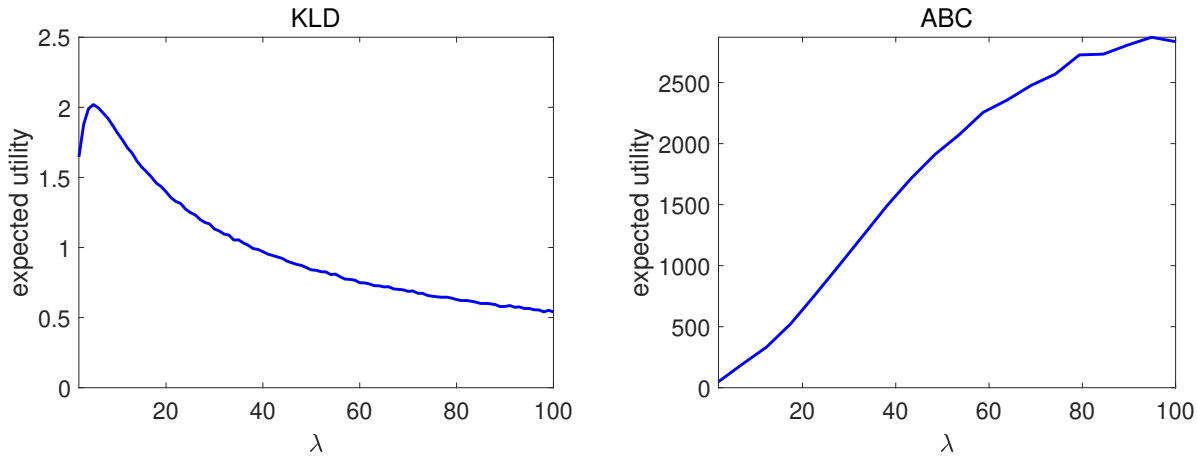
$$y = G(\theta, \lambda)(1 + \epsilon_1) + \epsilon_2, \quad G(\theta, \lambda) = \frac{1}{B(2, \lambda)}\theta(1 - \theta)^{\lambda - 1},$$

where $B(\cdot, \cdot)$ denotes the beta function, ϵ_1 and ϵ_2, are both normally distributed with a mean of 0 and a variance of 0.05^2. We have the luxury of an available likelihood in this instance, expressed as

$$p(y|\theta, \lambda) = N(G(\theta, \lambda), 0.05^2(1 + G^2(\theta, \lambda)).$$

We assume a uniform prior distribution over the $[0, 1]$ interval, and the design variable is selected from the $[2, 100]$ range. The utility of this toy problem is dual-pronged. Initially, the availability of the likelihood permits an accurate evaluation of the expected utility function, providing a benchmark to assess the validity of our approximation. Moreover, the distinctly non-Gaussian nature of the posterior in this scenario accentuates the disparity between the KLD-based utility and the Bayesian D-posterior precision utility.

To illuminate the distinctions between the KLD-based expected utility function and the Bayesian D-posterior precision utility function, we initiate our analysis by estimating the KLD-based expected utility function employing a nested Monte Carlo method with a large sample size, thereby approximating the exact value of the expected utility function. Figure 2.1 (a) presents a plot of this utility against the design parameter λ, clearly indicating the optimal design corresponds to $\lambda = 5$. For comparative analysis, Figure 2.1 (b) also displays the expected D-posterior precision utility estimated by ABC based method as a function of the design parameter λ. Intriguingly, this utility exhibits an almost monotonically increasing trend with λ, culminating in an optimal solution at the upper limit, $\lambda = 100$.



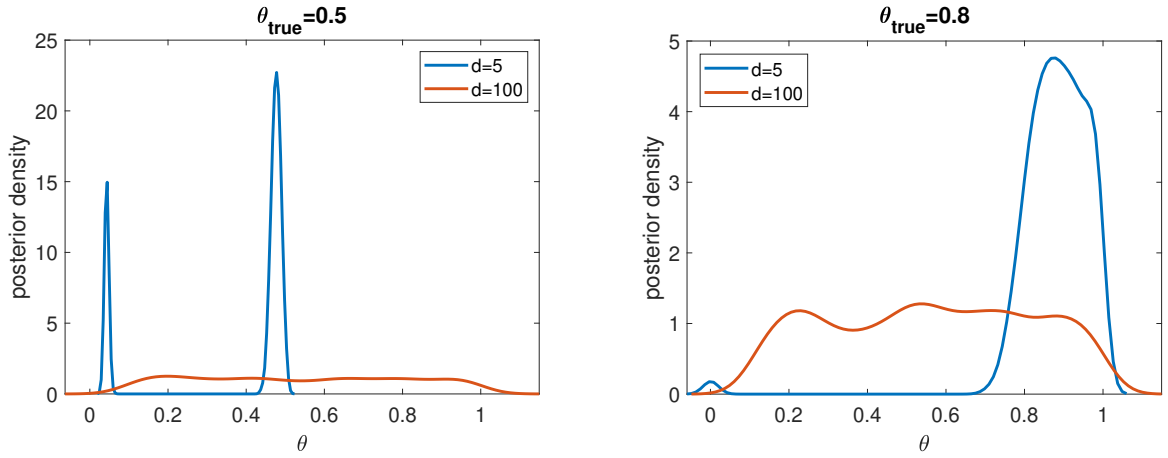
(a) (b)

Figure 2.1: (a) The expected utility plotted against the design parameter λ in the KLD method. (b) The expected utility plotted against the design parameter λ in the ABC method.

To evaluate the effectiveness of the two design strategies, we conducted numerical tests under the obtained two designs: $\lambda = 5$ and $\lambda = 100$. We collected data and applied Bayesian inference in scenarios where the true value of θ was 0.5 and 0.8. The resulting posterior distributions are displayed in Fig. 2.2. A closer look at these distributions reveals a significant departure from Gaussian behavior. For $\lambda = 5$, the posterior distributions are noticeably bimodal, with one mode closely aligning with the true θ value. Conversely, at $\lambda = 100$, the posteriors are less defined and informative. These findings underscore the KLD based utility's superiority in addressing problems characterized by non-Gaussian posterior distributions.

2.4.3 Connect Bayesian experimental design to entropy optimization

The KLD-based utility function has garnered prominence in design criteria, attributed to its theoretical and practical appeals. In this subsection, we delve deeper into its intricate connections with entropy optimization, expanding upon the preceding discussion. We delineate



(a) (b)

Figure 2.2: The posterior distributions for $\theta_{true} = 0.5$ (a) and $\theta_{true} = 0.8$ (b), obtained under the two experimental conditions $\lambda = 5$ and $\lambda = 100$.

how the BED problem, when approached through the lens of KLD-based utility, transforms into entropy optimization challenges. Consequently, the insights and methods delineated in Section 2.3 on entropy optimization become instrumental in navigating and resolving the intricacies of BED problems efficiently. The quintessence of these insights, analyses, and subsequent discussions is encapsulated in the ensuing propositions and remarks.

Proposition 2.4. *The KLD based expected utility function equals to the negative expected posterior entropy up to a constant, i.e.,*

$$U_{\text{KLD}}(\lambda) = -\mathbb{E}_{p(y|\lambda)}[H(q(\theta|y,\lambda))] + \text{const.} \tag{2.56}$$

Proof. By the definition of the KLD based utility function in Eq. (2.55), we have

$$\begin{aligned} U_{\mathrm{KLD}}(\lambda) &= \mathbb{E}_{p(y|\lambda)}[u_{\mathrm{KLD}}(\lambda, y)] \\ &= \mathbb{E}_{p(y|\lambda)}[\mathbb{E}_{q(\theta|y,\lambda)}[\log q(\theta|y,\lambda) - \log \pi_\theta(\theta)]] \\ &= \mathbb{E}_{p(y|\lambda)q(\theta|y,\lambda)}[\log q(\theta|y,\lambda)] - \mathbb{E}_{p(y|\lambda)q(\theta|y,\lambda)}[\log \pi_\theta(\theta)] \\ &= -\mathbb{E}_{p(y|\lambda)}[H(q(\theta|y,\lambda))] - \mathbb{E}_{\pi_\theta(\theta)l(y|\theta,\lambda)}[\log \pi_\theta(\theta)] \\ &= -\mathbb{E}_{p(y|\lambda)}[H(q(\theta|y,\lambda))] - \mathbb{E}_{\pi_\theta(\theta)}[\log \pi_\theta(\theta)] \\ &= -\mathbb{E}_{p(y|\lambda)}[H(q(\theta|y,\lambda))] + H(\pi_\theta(\theta)). \end{aligned} \tag{2.57}$$

Note that $H(\pi_\theta(\theta))$ is independent of λ, we thus have $U_{\mathrm{KLD}}(\lambda) = -\mathbb{E}_{p(y|\lambda)}[H(q(\theta|y,\lambda))] +$ const. □

Remark 2.4. *In practice, it is more efficient to optimize the rewritten KLD based expected utility function using unified optimization with variational estimators compared to alternative entropy optimization approaches. Concretely, this approach aims to maximize the following variational lower bound on* $U_{\mathrm{KLD}}(\lambda)$ *[13, 48] w.r.t.* (λ, ϕ)*:*

$$\widehat{U}_{\mathrm{KLD}}(\lambda, \phi) = \mathbb{E}_{\pi_\theta(\theta)l(y|\theta,\lambda)}[\log q_\phi(\theta|y,\lambda)], \tag{2.58}$$

where $q_\phi(\theta|y,\lambda)$ *is a family of probability distributions parameterized by* ϕ*. Maximizing this objective requires estimating the gradients* $\nabla_\lambda \widehat{U}_{\mathrm{KLD}}(\lambda, \phi)$ *and* $\nabla_\phi \widehat{U}_{\mathrm{KLD}}(\lambda, \phi)$*, which can be obtained by the reparameterization trick [106] utilizing only samples* $\theta^{(i)} \sim \pi_\theta(\theta)$*,* $y^{(i)} \sim l(y|\theta^{(i)}, \lambda)$ *for* $i = 1, ..., N$*. In contrast, multiple* θ*'s need to be sampled from each posterior* $q(\theta|y^{(i)}, \lambda)$ *when estimating* $H(q(\theta|y,\lambda))$ *or* $\nabla_\lambda H(q(\theta|y,\lambda))$ *with alternative methods.*

Proposition 2.5 ([146])**.** *The KLD based expected utility function can be written as*

$$U_{\mathrm{KLD}}(\lambda) = H(p(y|\lambda)) - \mathbb{E}_{\pi_\theta(\theta)}[H(l(y|\theta,\lambda))]. \tag{2.59}$$

Proof. By using Bayes' Theorem, we have

$$\begin{aligned} U_{\mathrm{KLD}}(\lambda) &= \mathbb{E}_{p(y|\lambda)}[u_{\mathrm{KLD}}(\lambda, y)] \\ &= \mathbb{E}_{p(y|\lambda)}[\mathbb{E}_{q(\theta|y,\lambda)}[\log q(\theta|y,\lambda) - \log \pi_\theta(\theta)]] \\ &= \mathbb{E}_{p(y|\lambda)q(\theta|y,\lambda)}[\log \frac{\pi_\theta(\theta) l(y|\theta,\lambda)}{p(y|\lambda)} - \log \pi_\theta(\theta)] \\ &= \mathbb{E}_{p(y|\lambda)q(\theta|y,\lambda)}[\log l(y|\theta,\lambda) - \log p(y|\lambda)] \\ &= -\mathbb{E}_{p(y|\lambda)}[\log p(y|\lambda)] + \mathbb{E}_{\pi_\theta(\theta)l(y|\theta,\lambda)}[\log l(y|\theta,\lambda)] \\ &= H(p(y|\lambda)) - \mathbb{E}_{\pi_\theta(\theta)}[H(l(y|\theta,\lambda))]. \end{aligned} \tag{2.60}$$

□

Remark 2.5. *In models where the likelihood function, $l(y|\theta,\lambda)$, is explicitly defined and tractable, one can achieve an unbiased estimation of the second term in Eq.* (2.59) *using traditional Monte Carlo methods. Therefore, the primary computational challenge in utilizing this representation pivots on the task of maximizing the entropy of the marginal distribution of the observed data, represented as $H(p(y|\lambda))$.Interestingly, when the entropy of the likelihood, $H(l(y|\theta,\lambda))$, is independent of λ - a situation commonly encountered in models with i.i.d. measurement noises - the expected utility function $U_{\mathrm{KLD}}(\lambda)$ simplifies to $H(p(y|\lambda)) + \mathrm{const}$. Under these circumstances, maximizing $U_{\mathrm{KLD}}(\lambda)$ is equivalent to maximizing $H(p(y|\lambda))$.*

Proposition 2.6 ([6])**.** *Let y and y' be two independent samples generated from $l(y|\theta,\lambda)$. Now define $z = y - y'$ with probability distribution $p(z|\theta,\lambda)$, and we then have*

$$U_{\mathrm{KLD}}(\lambda) \geq -H(\mathbb{E}_{\pi_\theta(\theta)}[p(z|\theta,\lambda)]) + \frac{dim(y)}{2}\log 2 + H(p(y|\lambda)), \tag{2.61}$$

where $dim(y)$ is the dimensionality of y.

Proof. From Shannon's entropy power inequality [34], we obtain,

$$\exp(2H(p(z|\theta,\lambda))/dim(y)) \geq 2\exp(2H(l(y|\theta,\lambda))/dim(y)),$$

which implies that

$$H(l(y|\theta,\lambda)) \leq H(p(z|\theta,\lambda)) - \frac{dim(y)}{2}\log 2. \tag{2.62}$$

Taking expectation over $\pi_\theta(\theta)$ on both sides of Eq. (2.62) yields,

$$\begin{aligned} &\mathbb{E}_{\pi_\theta(\theta)}[H(l(y|\theta,\lambda))] \\ \leq &\mathbb{E}_{\pi_\theta(\theta)}[H(p(z|\theta,\lambda))] - \frac{dim(y)}{2} log2 \\ \leq &H(\mathbb{E}_{\pi_\theta(\theta)}[p(z|\theta,\lambda)]) - \frac{dim(y)}{2} log2, \end{aligned} \tag{2.63}$$

where the last inequality is due to the concavity of the entropy [34]. Finally using Eq. (2.59), we have

$$U_{\mathrm{KLD}}(\lambda) \geq -H(\mathbb{E}_{\pi_\theta(\theta)}[p(z|\theta,\lambda)]) + \frac{dim(y)}{2}\log 2 + H(p(y|\lambda)), \tag{2.64}$$

□

Remark 2.6. *The above proposition can be seen as an extension of Eq.* (2.59) *to BED applications without tractable likelihood. Concretely, Eq.* (2.61) *provides a lower bound estimate for* $U_{\mathrm{KLD}}(\lambda)$*, formulated as the difference between the two entropies. Hence, the task of maximizing the expected utility function* $U_{\mathrm{KLD}}(\lambda)$ *can be reframed as maximizing this lower bound - a clear entropy optimization problem only encompassing two entropies. In contrast, direct use of Eq.* (2.59) *as the objective implies an optimization of expected entropy, necessitating the manipulation of multiple entropies computationally.*

2.4.4 Review of literature on methodologies in Bayesian experimental design

This subsection provides a thorough exploration of the existing literature on Bayesian experimental design (BED). Particularly, our focus is on the BED methods grounded in the KLD based utility. As previously outlined, the quest for optimal experimental designs can be encapsulated in two distinct but related objectives. The first pertains to minimizing the expected

entropy of the posterior distribution $q(\theta|y,\lambda)$, expounded in Proposition 2.4. In contrast, the second objective converges on maximizing the entropy of the marginal likelihood $p(y|\lambda)$, illuminated in Proposition 2.5. A third perspective emerges when considering models lacking a closed-form likelihood function. In these instances, the focus transitions to maximizing the expectation of the log likelihood ratio, expressed as $\frac{l(y|\theta,\lambda)}{p(y|\lambda)}$. This perspective is thoroughly detailed in the seminal work by Kleinegesse et al. [82]. Consequently, methodologies in BED can be systematically organized into three principal categories, each aligning with the aforementioned perspectives. These are: estimating the posterior distribution, estimating the marginal likelihood and estimating the likelihood ratio.

Estimating the posterior distribution

When we view BED as a task of minimizing the expected entropy of the posterior distribution, a pivotal query emerges: how can one precisely estimate this posterior distribution?

Markov Chain Monte Carlo (MCMC) methods, as explored in [5], have been the cornerstone of Bayesian statistics for the estimation of posterior distributions. However, it is important to note that MCMC does not produce a probability density function (PDF) for the posterior. Instead, it yields a collection of samples from this distribution. This nuance implies that, within the context of BED, an auxiliary step is necessary: converting these MCMC samples into a posterior PDF. A prevalent methodology for this, proposed in [71], leverages Gaussian Mixture Models [70] to approximate the posterior PDF using the MCMC samples.

The Laplace approximation presents an alternative strategy. It approximates the posterior distribution with a Gaussian form, subsequently offering a closed-form solution for the posterior entropy. BED methods that utilize the Laplace approximation [89, 28, 96, 141, 95] offer computational advantages. However, a significant limitation of this line

of work is its reliance on a strong structural assumption about the shape of the posterior distribution. This assumption can restrict its applicability, as it may not be suitable for posterior distributions that deviate substantially from a unimodal and symmetric Gaussian shape, limiting its generalizability.

More recently, ideas from amortized variational inference have been introduced to address the challenges posed by the intractability of posterior distributions in BED problems. Approaches inspired by these ideas, such as those discussed in [13] and [49], follow a two-step process. First, they learn a parametric amortized approximation $q_\phi(\theta|y,\lambda)$ to the posterior $q(\theta|y,\lambda)$. Then, this learned approximation is utilized to estimate the posterior entropy. Unlike MCMC and Laplace approximation running separate inference for each y, the amortized approximation based methods use a shared parametric model to approximate the posterior distribution, spreading the cost of inference across all y's and making the overall inference process more efficient.

Estimating the marginal likelihood

Estimating the posterior densities can be avoided by reframing BED as a problem of maximizing the entropy of the marginal likelihood. This shift, while promising, brings with it the challenge of estimating the often intractable marginal likelihood $p(y|\lambda)$.

The entropy of this marginal likelihood can be effectively computed using the Nested Monte Carlo (NMC) estimator. This methodology employs an inner Monte Carlo iteration specifically to estimate $p(y|\lambda)$, as supported by various studies [144, 109, 74, 133]. Rigorous theoretical analyses [135, 184, 17] have shed light on the efficiency and accuracy of the NMC estimator. They have established that the estimator is asymptotically unbiased. Its rate of convergence is governed by $O(\sqrt{N^{-1}+cM^{-2}})$, where N and M denote the number of samples utilized for the outer and inner Monte Carlo estimations, respectively. Nonetheless,

it is important to note – and as will be delved into in chapter 5 – that the computational expense associated with the NMC estimator scales exponentially in relation to the true value of the expected utility function.

Alternatively, the amortized variational technique finds utility once more in this context. In their work, [49] also proposes a "forward" variational approach for estimating the entropy of the marginal likelihood. Distinguishing itself from the approach of acquiring the variational posterior approximation in the aforementioned posterior entropy perspective, this method instead aims to learn an amortized approximation $p_\psi(y|\lambda)$ to the intractable marginal likelihood $p(y|\lambda)$. Following this, the approximation is deployed to estimate the entropy of the marginal likelihood, yielding an upper bound estimation of this entropy. The decision to opt for either the variational posterior or the marginal likelihood approximation is contingent upon the specific nature of the problem at hand. Typically, in contexts characterized by a high-dimensional θ juxtaposed against a low-dimensional y, the task of acquiring the variational approximation for the marginal likelihood is less cumbersome compared to its posterior counterpart. Such ease of adaptability renders the "forward" variational estimator particularly compelling in these scenarios.

Estimating the likelihood ratio

In an effort to expand BED to encompass implicit models, a fresher perspective has surfaced, viewing $U_{\text{KLD}}(\lambda)$ as the expectation of a likelihood ratio:

$$U_{\text{KLD}} = \mathbb{E}_{\pi_\theta(\theta)l(y|\theta,\lambda)}[\log w(y,\theta|\lambda)], \tag{2.65}$$

where $w(y,\theta|\lambda) = \frac{l(y|\theta,\lambda)}{p(y|\lambda)}$ is the density ratio between likelihood and marginal likelihood functions, commonly referred to as the likelihood ratio. Given this new representation, the key now turns to estimate the likelihood ratio.

Techniques for estimating the likelihood ratio have been previously investigated in the context of mutual information [163] and divergence functionals [113] estimation. In the realm of BED, the study by [82] stands out as a pioneering effort that seeks to design Bayesian experiments for implicit models through the estimation of likelihood ratios. This research harnesses the Likelihood-Free Inference by Ratio Estimation (LFIRE) approach [165] to approximate the likelihood ratio, subsequently integrating it into Eq. (2.65) to obtain the approximation of $U_{\mathrm{KLD}}(\lambda)$. Their subsequent works [81, 83] introduce the application of MINE [19] and various variational mutual information estimators, like NWJ [113] and its JSD proxy [73], for utility function optimization. Remarkably, a deeper inspection reveals that these methodologies are essentially rooted in the likelihood ratio estimation principle, as the optimal variational networks they employ equal to specific functionals of the likelihood ratio.

BED methods for implicit models

The approaches outlined above, with the exception of those founded on likelihood ratio estimation, often require the likelihood function to be explicitly defined and tractable. Yet, the rise of implicit models introduces distinct challenges in BED, leading to the development of a variety of BED strategies tailored specifically for these models.

The genesis of BED for implicit models can be traced back to the contributions of [32]. They utilized moment closure [88] to approximate the intractable likelihood functions commonly found in epidemiological studies. Price et al. [132] later utilized rejection Approximate Bayesian Computation (ABC) for posterior sampling and harnessed histogram binning for the subsequent estimation of the expected utility function. Ao et al. [6] proposed a lower bound approximation of the expected utility function as the difference between two entropies, calculated using a k-Nearest Neighbors (k-NN) entropy estimator. In parallel, Overstall et

al. [115] explored the use of multivariate Gaussian process emulators for likelihood function approximation, coupled with a copula-based technique for marginal likelihood approximation.

Optimization

Our previous discourse has underscored that the optimization of design variables in BED fundamentally revolves around entropy optimization, at least in part. Analogous to the categorization in entropy optimization (see Section 2.3 for detailed discussions), the existing methodologies within BED can be broadly classified into two main groups: two-stage optimization and gradient estimation.

Early efforts in BED predominantly employed the two-stage approach, treating the expected utility estimators as black-box functions and then applying a separate optimization technique to maximize them. Both general-purpose algorithms and those tailored for BED have been utilized in the optimization stage. For instance, [74] employed Simultaneous perturbation stochastic approximation (SPSA) and Nelder-Mead nonlinear simplex (NMS) to optimize the approximate expected utility function. The coordinate exchange algorithm was adopted by [103] and [116]. Other studies, such as [82], [49], and [6], explored Bayesian optimization (BO). Evolutionary algorithms [180], like the genetic algorithms presented in [67] and the Induced Natural Selection Heuristic (INSH) introduced by [131], have also been employed.

On the other hand, gradient estimation based methods are garnering attention in BED. Taking advantage of the modern automatic differentiation frameworks, variational estimators of the expected utility function, as presented in [81, 83, 48], can now seamlessly compute gradients with respect to design variables. This facilitates simultaneous updates to both the design variables and the associated variational parameters, often leveraging gradient descent. A distinct trajectory focuses on the direct estimation of the gradients of the expected utility

function with respect to design variables. An exemplar of this is the work by [57], which employs Multi-Level Monte Carlo (MLMC) [139] techniques to estimate gradients for the expected utility function. Notably, while the broader domain of entropy optimization has delved deeply into the gradient estimation based methods, their applications within BED remains an area ripe for further exploration.

Chapter Three

Entropy Estimation via Uniformization

3.1 Introduction

Entropy, a fundamental concept in information theory, has found applications in various fields such as physics, statistics, signal processing, and machine learning. For example, in the statistics and data science contexts, various applications rely critically on the estimation of entropy, including goodness-of-fit testing [171, 59], sensitivity analysis [11], parameter estimation [136, 177], and Bayesian experimental design [146, 6].

In this work we focus on the continuous version of entropy that takes the form,

$$H(X) = -\int \log[p_{\mathrm{x}}(\mathrm{x})]p_{\mathrm{x}}(\mathrm{x})d\mathrm{x}, \tag{3.1}$$

where $p_{\mathrm{x}}(\mathrm{x})$ is the probability density function (PDF) of random variable X. Despite the rather simple definition, entropy only admits an analytical expression for a limited family of distributions and needs to be evaluated numerically in general. When the distribution of interest is analytically available, in principle its entropy can be estimated by numerical integration schemes such as the Monte Carlo method. However, in many real-world applications, the distribution of interest is not analytically available, and one has to estimate the entropy

from the realizations drawn from the target distribution, which makes it difficult or even impossible to directly compute the entropy via numerical integration.

Entropy estimation has attracted considerable attention from various communities in the last a few decades, and numerous methods have been developed to directly estimate entropy from realizations. In this work we only consider non-parametric approaches which do not assume any parametric model of the target distribution, and those methods can be broadly classified into two categories. The first class of methods, are known as the plug-in estimators, which first estimate the underlying probability density, and then compute the integral in Eq. (3.1) using numerical integration or Monte Carlo (see [18] for a detailed description). Some examples of density estimation approaches that have been studied for plug-in methods are kernel density estimator [79, 66, 107, 128], histogram estimator [63, 66] and field-theoretic approach [31]. A major limitation of this type of methods is that they rely on an effective density estimation, which is a difficult problem in its own right, especially when the dimensionality of the problem is high. A different strategy is to directly estimate the entropy from the independent samples of the random variable. Popular methods falling in this category include the sample-spacing [104] and the k-nearest neighbors (k-NN) [84, 85] based estimators. The latter is particularly appealing among the existing estimation methods thanks to its theoretical and computational advantages and has been widely used in practical problems. Efforts have been constantly devoted to extending and improving the k-NN methods, and some recent variants and extensions of the methods are [53, 97, 20]. It is also worth mentioning that there are many other types of direct entropy estimators available. For example, Ariel and Louzoun [10] decoupled the target entropy to a sum of the entropy of marginals, which is estimated using one-dimensional methods, and the entropy of copula, which is estimated recursively by splitting the data along statistically dependent dimensions. Kandasamy et al. [80] suggested a leave-one-out technique for the von Mises expansion based estimator [47]. We also note that in certain applications the main purpose is to minimize

or maximize the quantity of entropy, and in this case entropy gradient estimation strategies [176, 91] have been explored to avoid direct entropy estimation.

It is well known that, entropy estimation becomes increasingly more difficult as the dimensionality grows, and such difficulty is mainly due to the *estimation bias*, which decays very slowly with respect to sample size for high-dimensional problems. For example in many popular approaches including the k-NN method [84], the estimation bias decays at the rate of $O(N^{-\gamma/d})$ where N is the sample size, d is the dimensionality, and γ is a positive constant [87, 80, 54, 160]. As a result, very few, if not none, of the existing entropy estimation methods can effectively handle high-dimensional problems without making strong assumptions about the smoothness of the underlying distribution [80]. Indeed, the well-known minimax bias results (e.g., [68, 22]) indicate that without the strong smoothness assumption [80], the curse of dimensionality is unavoidable. However, efforts can still be made to reduce the difference between the actual estimation bias and the theoretical bound.

The main goal of this work is to provide an effective entropy estimation approach which can achieve faster bias decaying rate than traditional k-NN methods under mild smoothness assumption, and thus can effectively deal with high-dimensional problems. The method presented here consists of two main ingredients. First, we propose two truncated k-NN estimators based on those by [84] and [85] respectively, and also provide the bounds of the estimation bias in these estimators. Interestingly our theoretical results suggest that the estimators achieve *zero bias* for uniform distributions, while there is no such a result for any existing k-NN based estimators, according to the bias analysis available to date [54, 153, 21]. This property offers the possibility to significantly improve the performance of entropy estimation by mapping the data points toward a uniform distribution, a procedure that we refer to as *uniformization*. Therefore the second main ingredient of the method is to conduct the uniformization of the data points, with the normalizing flow (NF) technique [138, 122]. Simply speaking, NF constructs a sequence of invertible and differentiable mappings that

transform a simple base distribution such as standard Gaussian into a more complicated distribution whose density function may not be available. Specifically we use the Masked Autoregressive Flow [120], an NF algorithm originally developed for density estimation, combined with the probability integral transform, to push the original data points towards the uniform distribution. We then estimate the entropy of the resulting near-uniform data points with the proposed truncated k-NN estimators, and derive that of the original ones accordingly (by adding an entropic correction term due to the transformation). Therefore, by combining the truncated k-NN estimators and the normalizing flow model, we are able to decode a complex high-dimensional distribution represented by the realizations, and obtain an accurate estimation of its entropy.

The rest of the paper is organized as follows. In Section 3.2, we describe the traditional k-NN based methods of entropy estimation and their convergence properties. In Section 3.3, we introduce the truncated k-NN estimators for distributions with compact support, and then show how to combine these new estimators with the NF-based uniformization procedure to estimate the entropy of general distributions. Numerical examples and applications are presented in Section 3.4 and Section 3.5 respectively to demonstrate the effectiveness of the proposed methods. Finally, in Section 3.7, we summarize our findings and discuss some future research directions.

3.2 k-NN Based Entropy Estimation

We provide a brief introduction to two commonly used k-NN based entropy estimators in this section. We start with the original k-NN entropy estimator proposed in [84], where the k-th nearest neighbor is contained in the smallest possible closed ball. Next, we introduce a popular variant of the k-NN estimator proposed in [85] - the Kraskov-Stögbauer-Grassberger

(KSG) estimator, and this method uses the smallest possible hyper-rectangle to cover at least k points. We finally discuss some theoretical analysis of estimation errors in the estimators.

3.2.1 Kozachenko-Leonenko Estimator

Recall the definition of entropy in Eq. (3.1). Given a density estimator $\widehat{p_{\mathrm{x}}}(\mathrm{x})$ for $p_{\mathrm{x}}(\mathrm{x})$ and a set of N i.i.d. samples $S = \{\mathrm{x}^{(i)}\}_{i=1}^{N}$ drawn from $p_{\mathrm{x}}(\mathrm{x})$, the entropy of the random variable X can be estimated as follows:

$$\widehat{H}(X) = -N^{-1} \sum_{i=1}^{N} \log \widehat{p_{\mathrm{x}}}(\mathrm{x}^{(i)}). \tag{3.2}$$

The Kozachenko-Leonenko (KL) estimator depends on a local uniformity assumption to obtain the estimate $\widehat{p_{\mathrm{x}}}(\mathrm{x})$. For each $x^{(i)}$, one first identifies the k-nearest neighbors (in terms of the p-norm distance) of it, and defines the smallest closed ball covering all these k neighbors as:

$$B(\mathrm{x}^{(i)}, \epsilon_i(k)/2) = \{\mathrm{x} \in \mathbb{R}^d \mid \|\mathrm{x} - \mathrm{x}^{(i)}\|_p \leq \epsilon_i(k)/2\},$$

where $\epsilon_i(k)$ be twice the distance between $\mathrm{x}^{(i)}$ and its k-th nearest neighbor among the set S. We shall refer to the closed ball $B(\mathrm{x}^{(i)}, \epsilon_i(k)/2)$ as a *cell* centered at $\mathrm{x}^{(i)}$, and let q_i be the mass of the cell $B(\mathrm{x}^{(i)}, \epsilon_i(k)/2)$, i.e.,

$$q_i(\epsilon_i(k)) = \int_{\mathrm{x} \in B(\mathrm{x}^{(i)}, \epsilon_i(k)/2)} p_{\mathrm{x}}(\mathrm{x}) d\mathrm{x}.$$

It can be derived that the expectation value of $\log q_i$ over $\epsilon_i(k)$ is given by

$$\mathbb{E}(\log q_i) = \psi(k) - \psi(N), \tag{3.3}$$

where $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ with $\Gamma(x)$ being the Gamma function [85]. KL estimator then assumes that the density is constant in $B(x^{(i)}, \epsilon_i(k)/2)$, which gives

$$q_i(\epsilon_i(k)) \approx c_d \epsilon_i(k)^d p_{\mathrm{x}}(\mathrm{x}^{(i)}), \tag{3.4}$$

where d is the dimension of X and

$$c_d = \Gamma(1+\frac{1}{p})^d/\Gamma(1+\frac{d}{p}),$$

is the volume of the d-dimensional unit ball [1] with respect to p-norm. Combining (3.3) and (3.4) one can get an estimate of the log-density at each sample point,

$$\log \widehat{p}_{\mathrm{x}}(\mathrm{x}^{(i)}) = \psi(k) - \psi(N) - \log c_d - d\log \epsilon_i(k). \tag{3.5}$$

Plugging the above estimates for $i = 1, ..., N$ into (3.2) yields the KL estimator:

$$\widehat{H}_{\mathrm{KL}}(X) = -\psi(k) + \psi(N) + \log c_d + \frac{d}{N}\sum_{i=1}^{N} \log \epsilon_i(k). \tag{3.6}$$

3.2.2 KSG Estimator

As is mentioned earlier, the Kraskov-Stögbauer-Grassberger (KSG) estimator [85] is an important variant of $\hat{H}_{\mathrm{KL}}$. Unlike KL estimator that is based on closed balls, KSG estimator uses hyper-rectangles to form the cells at each data point. Namely one chooses the ∞-norm as the distance metric (i.e $p = \infty$), and as a result the cell $B(x^{(i)}, \epsilon_i(k)/2)$ becomes a hyper-cube with side length $\epsilon_i(k)$. Next, we allow the hyper-cube to become a hyper-rectangle: i.e., the cells admit different side lengths along different dimensions. Specifically, for $j = 1, ..., d$, we define $\epsilon_{i,j}(k)$ to be twice of the distance between $x^{(i)}$ and its k-th nearest neighbor along dimension j, and the cell centered at $\mathrm{x}^{(i)}$ covering its k-nearest neighbors becomes

$$\begin{aligned} B(\mathrm{x}^{(i)}, \epsilon_{i,1:d}(k)/2) = \{\mathrm{x} = (\mathrm{x}_1, ..., \mathrm{x}_d) \,|\, &|\mathrm{x}_j - \mathrm{x}_j^{(i)}| \le \epsilon_{i,j}(k)/2, \\ &\text{for } j = 1, ..., d\}, \end{aligned} \tag{3.7}$$

where $\epsilon_{i,1:d}(k) = (\epsilon_{i,1}(k), ..., \epsilon_{i,d}(k))$. This change leads to a different formula for computing the mass of the cell $B(\mathrm{x}^{(i)}, \epsilon_{i,1:d}/2)$,

$$\mathbb{E}(\log q_i) \approx \psi(k) - \frac{d-1}{k} - \psi(N). \tag{3.8}$$

[1] "Unit ball" here refers to a ball whose diameter, rather than radius, is 1.

It is worth noting that the equality in Eq. (3.3) is replaced by approximate equality in Eq. (3.8), because a uniform density within the rectangle has to be assumed to obtain Eq. (3.8) (see Lemma 2 in 4.1.2 for details). Using a similar local assumption as Eq. (3.4), the KSG estimator is derived as,

$$\widehat{H}_{\mathrm{KSG}}(X) = -\psi(k) + \psi(N) + \frac{d-1}{k} + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\log \epsilon_{i,j}(k). \tag{3.9}$$

We note that the KSG method was actually developed in the context of estimating mutual information [85], and has been reported to outperform the KL estimator in a wide range of problems [54]. As has been shown above, it is straightforward to extend it to entropy estimation, and our numerical experiments also suggest that it has competitive performance as an entropy estimator, which will be demonstrated in Section 3.4.

3.2.3 Convergence Analysis

Another important issue is to analyze the estimation errors in these entropy estimators and especially how they behave as the sample size increases. In most of the k-NN based estimators including the two mentioned above, the variance is generally well controlled, decaying at a rate of $O(N^{-1})$ with N being the sample size, while the main issue lies on the estimation bias. In fact, the bias of estimator $\widehat{H}_{\mathrm{KL}}$ has been well studied, but that of $\widehat{H}_{\mathrm{KSG}}$ receives very little attention. Previous results related to the former are listed as follows. The original [84] paper established the asymptotic unbiasedness for $k = 1$ while [152] obtained the same result for general k. For distributions with unbounded support, [170] proved that the bias bound decays at a rate of $O(\frac{1}{\sqrt{N}})$ for $d = 1$. [54] generalized it to higher dimensions, obtaining a bias bound of $O(N^{-\frac{1}{d}})$ up to polylogarithmic factors. For distributions compactly supported, usually densities satisfying the β-Hölder condition are considered. [21] gave a quick-and-dirty upper bound of bias, $O(N^{-\beta})$, for a simple class of univariate densities supported on $[0, 1]$ and bounded away from zero. [153] proved the bias is around $O(N^{-\frac{\beta}{d}})$ ($\beta \in (0, 2]$) for general

d with some additional conditions on the boundary of support. We reinstate that all these works obtained a variance bound of $O(N^{-1})$.

It should be noted that the bias bounds given by previous studies typically depend on some properties of target densities, such as smoothness parameter and Hessian matrix, providing insights that these estimators perform well on certain distributions. This motivates the idea that one can transform the given data points toward a desired distribution for a more accurate entropy estimation, which is detailed in next section.

3.3 Uniformizing Mapping Based Entropy Estimation

In this section, we present the proposed approach in detail. As is mentioned earlier, it consists of two main ingredients: a truncated version of the k-NN entropy estimators, and a transformation that can map data points toward a uniform distribution.

3.3.1 Truncated KL/KSG Estimators

For compactly supported distributions, a significant source of bias comes from the boundary of the support, where the k-NN cells are constructed including areas outside of the support of the distribution density [153]. Intuitively speaking, incorrectly including such areas results in an underestimate of the densities, leading to bias in the estimator. We thus propose a method to reduce the estimation bias by excluding the areas outside of the distribution support, and remarkably the resulting estimator enjoy certain convergence properties which enable us to design the NF based estimation approach. The only additional requirement for using these estimators is that the bound of support of density should be specified. For the purposes of analysis, without loss of generality, we suppose the target density is supported on the unit

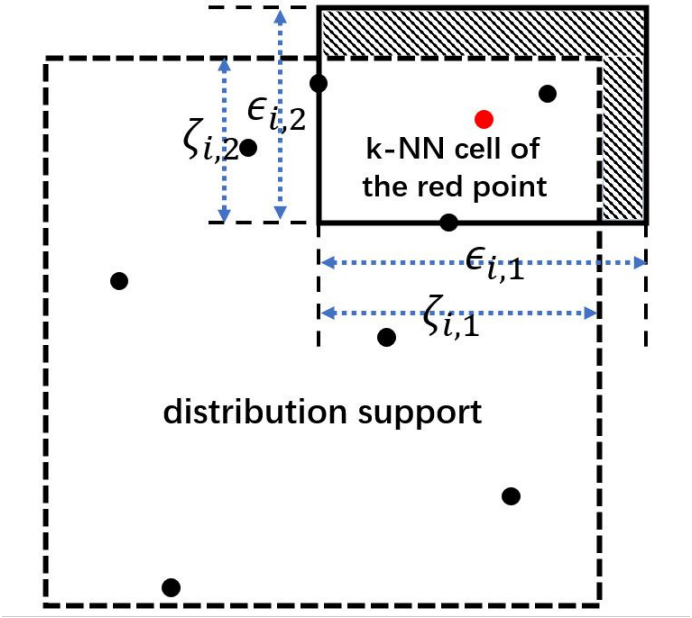


Figure 3.1: The schematic illustration of the truncated KSG estimator with $k = 3$. The shaded area is that removed from the k-NN cell.

cube $\mathcal{Q} := [0,1]^d$ in $\mathbb{R}^d$. The procedure of our method is as follows: we first determine all the cells using either KL or KSG, then examine whether each k-NN cell covers area out of the distribution support, and if so, truncate the cell at the boundary to exclude such area (see Fig. 3.1 for a schematic illustration). Mathematically the truncated KL (tKL) estimator (with ∞-norm), is given by

$$\widehat{H}_{\mathrm{tKL}}(X) = -\psi(k) + \psi(N) + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\log \xi_{i,j}(k), \tag{3.10}$$

where

$$\xi_{i,j}(k) = \min\{\mathrm{x}_j^{(i)} + \epsilon_i(k)/2, 1\} - \max\{\mathrm{x}_j^{(i)} - \epsilon_i(k)/2, 0\};$$

and the truncated KSG (tKSG) esitmator is given by

$$\widehat{H}_{\mathrm{tKSG}}(X) = -\psi(k) + \psi(N) + (d-1)/k + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\log \zeta_{i,j}(k), \tag{3.11}$$

where

$$\zeta_{i,j}(k) = \min\{\mathrm{x}_j^{(i)} + \epsilon_{i,j}(k)/2, 1\} - \max\{\mathrm{x}_j^{(i)} - \epsilon_{i,j}(k)/2, 0\}.$$

Next we shall theoretically analyze the biases and variances of the truncated estimators. Our analysis relies on some assumptions on the density function p_{x}, which are summarized as below:

Assumption 3.1. *The distribution p_{x} satisfies:*

(a) p_{x} is continuous and supported on $\mathcal{Q}$;

(b) p_{x} is bounded away from 0, i.e., $C_1 = \inf_{\mathrm{x}\in\mathcal{Q}} p_{\mathrm{x}}(\mathrm{x}) > 0$;

(c) The gradient of p_{x} is uniformly bounded on the interior of its support $\mathcal{Q}^o$, i.e., $C_2 = \sup_{\mathrm{x}\in\mathcal{Q}^o} ||\nabla p_{\mathrm{x}}(\mathrm{x})||_1 < \infty$.

First we consider the bias of estimator $\widehat{H}_{\mathrm{tKL}}$ and the following theorem states that, the bias in $\widehat{H}_{\mathrm{tKL}}$ is bounded and vanishes at the rate of $O(N^{-\frac{1}{d}})$.

Theorem 3.1. *Under Assumption 3.1 and for any finite k and d, the bias of the truncated KL estimator is bounded by*

$$\big|\mathbb{E}[\widehat{H}_{\mathrm{tKL}}(X)] - H(X)\big| \leq \frac{C_2}{C_1^{1+1/d}}\big(\frac{k}{N}\big)^{\frac{1}{d}}.$$

The variance of the truncated KL estimator is bounded by

$$\mathrm{Var}[\widehat{H}_{\mathrm{tKL}}(X)] \leq C\frac{1}{N},$$

for some $C > 0$.

Proof. We provide a skeleton proof here, where the complete proof including the notations is detailed in 4.1.3 and 4.1.4. The definitions of new notations can be found in 4.1.1.

Proof of the bias bound for the truncated KL estimator proceeds as follows.

1. Show that

$$\mathbb{E}[\widehat{H}_{tKL}(X)] = -\mathbb{E}\big[\log\frac{P(\overline{B}(\mathrm{x};\epsilon_k/2))}{\mu(\overline{B}(\mathrm{x};\epsilon_k/2))}\big]. \tag{3.12}$$

2. Bound the following difference by

$$\left| \log p(\mathrm{x}) - \log \frac{P(\overline{B}(\mathrm{x}; \epsilon_k/2))}{\mu(\overline{B}(\mathrm{x}; \epsilon_k/2))} \right| \leq \frac{C_2}{2C_1} \epsilon_k. \tag{3.13}$$

3. Note that $H(X) = -\mathbb{E}(\log p(x))$, and using Eq. (3.12), Eq. (3.13) and the upper bound of $\mathbb{E}(\epsilon_k)$ obtained from Lemma 4.4, we can derive that the bias $\mathbb{E}[\widehat{H}_{tKL}(X)]$ is bounded by

$$\left| \mathbb{E}[\widehat{H}_{tKL}(X)] - H(X) \right| \leq \frac{C_2}{C_1^{1+1/d}} \left(\frac{k}{N}\right)^{\frac{1}{d}}. \tag{3.14}$$

Proof of the variance bound for the truncated KL estimator proceeds as follows.

1. Let $\alpha_i = \sum_{j=1}^{d} \log \xi_{i,j}(k)$ and let α_i^* (for $i = 2, ..., N$) be the estimators with sample $\mathrm{x}^{(1)}$ removed. Then, by the Efron-Stein inequality [44],

$$\mathrm{Var}[\widehat{H}_{tKL}(X)] = \mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N} \alpha_i\right] \leq 2N\mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N} \alpha_i - \frac{1}{N}\sum_{i=2}^{N} \alpha_i^*\right)^2\right]. \tag{3.15}$$

2. Let $\mathbb{1}_{E_i}$ be the indicator function of the event $E_i = \{\epsilon_k(\mathrm{x}^{(1)}) \neq \epsilon_k^*(\mathrm{x}^{(1)})\}$, where $\epsilon_k^*(\mathrm{x}^{(1)})$ is twice the k-NN distance of $\mathrm{x}^{(1)}$ when α_i^* are used. Then we show that

$$N^2\left(\frac{1}{N}\sum_{i=1}^{N} \alpha_i - \frac{1}{N}\sum_{i=2}^{N} \alpha_i^*\right)^2 \leq (1 + C_{k,d})\left(\alpha_1^2 + 2\sum_{i=2}^{N} \mathbb{1}_{E_i}(\alpha_i^2 + \alpha_i^{*2})\right), \tag{3.16}$$

where $C_{k,d}$ is a constant.

3. Since α_i and α_i^* are identically distributed, we only need to derive the upper bounds of the following three expectations: $\mathbb{E}[\alpha_1^2]$, $(N-1)\mathbb{E}[\mathbb{1}_{E_2}\alpha_2^2]$ and $(N-1)\mathbb{E}[\mathbb{1}_{E_2}\alpha_2^{*2}]$.

4. Finally we obtain the bound of the variance of $\widehat{H}_{tKL}(X)$

$$\mathrm{Var}[\widehat{H}_{tKL}(X)] \leq C\frac{1}{N}, \tag{3.17}$$

for some $C > 0$.

□

Note that $C_2 = 0$ when p_{x} is uniform on $\mathcal{Q}$, and the following corollary follows directly:

Corollary 3.1. *Under the assumption in Theorem 3.1, if X is uniformly distributed on $\mathcal{Q}$, then the truncated KL estimator is unbiased.*

This corollary is the theoretical foundation of the proposed method, as it suggests that if one can transform the data points into a uniform distribution, the tKL method can yield an unbiased estimate. In reality, it is usually impossible to map the data point exactly into a uniform distribution to achieve the unbiased estimate. To this end, Theorem 3.1 suggests that, as long as the transformed samples are close to a uniform distribution in the sense that C_2 is small, the transformation can still significantly reduce the bias. Since the main contribution of the mean-square estimation error comes from the bias (as the variance decays at the rate of $O(N^{-1})$), reducing the bias therefore leads much more accurate estimation of the entropy.

We next consider the bias of the tKSG estimator. The second theorem shows that the expectation of $\widehat{H}_{\mathrm{tKSG}}$ has the same limiting behavior up to a polylogarithmic factor in N.

Theorem 3.2. *Under Assumption 3.1 and for any finite k and d, the bias of the truncated KSG estimator is bounded by*

$$\left|\mathbb{E}[\widehat{H}_{\mathrm{tKSG}}(X)] - H(X)\right| \leq C\frac{(\log N)^{k+2}}{C_1^{k+1}N^{\frac{1}{d}}}$$

for some $C > 0$. The variance of the truncated KSG estimator is bounded by

$$\mathrm{Var}[\widehat{H}_{\mathrm{tKSG}}(X)] \leq C'\frac{(\log N)^{k+2}}{N},$$

for some $C' > 0$.

Proof. Again, we only provide a skeleton proof here, with the complete details given in 4.1.5 and 4.1.6. The definitions of new notations can be found in 4.1.1.

Proof of the bias bound for the truncated KSG estimator proceeds as follows.

1. Suppose that $\widetilde{P}$, $\widetilde{p}$, and $\widetilde{q}_{\epsilon_k^{\mathrm{x}_1},\ldots,\epsilon_k^{\mathrm{x}_d}}(\mathrm{x})$ are defined as in Lemma 4.2 with $l = p(\mathrm{x})^{-\frac{1}{d}}$, and by Lemma 4.2 and the fact that $\sum_{j=1}^{d} \log \zeta_{i,j}(k)$ are identically distributed, we have

$$\mathbb{E}[\widehat{H}_{tKSG}(X)] = \mathop{\mathbb{E}}_{\mathrm{x}\sim pP}\mathbb{E}\big[\log \zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\big] - \mathop{\mathbb{E}}_{\mathrm{x}\sim p\widetilde{P}}\mathbb{E}\big[\log\big(p(\mathrm{x})\epsilon_k^{\mathrm{x}_1}\cdots\epsilon_k^{\mathrm{x}_d}\big)\big]. \tag{3.18}$$

2. We separate the d-dimensional unit cube $\mathcal{Q}$ into two subsets, $\mathcal{Q} = \mathcal{Q}_1 + \mathcal{Q}_2$, where $\mathcal{Q}_1 := [\frac{a_N}{2}, 1-\frac{a_N}{2}]^d$, $a_N = \big(\frac{2k\log N}{C_1 N}\big)^{\frac{1}{d}}$, and $\mathcal{Q}_2 = \mathcal{Q} - \mathcal{Q}_1$.

3. Note that $H(X) = -\mathbb{E}(\log p(x))$, and we can then decompose the bias into three terms according to the above separation of unit cube:

$$\begin{aligned}
&\big|\mathbb{E}[\widehat{H}_{tKSG}(X)] - H(X)\big| \\
=&\left|\mathop{\mathbb{E}}_{\mathrm{x}\sim pP}\mathbb{E}\big[\log\big(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\big)\big] - \mathop{\mathbb{E}}_{\mathrm{x}\sim p\widetilde{P}}\mathbb{E}\big[\log\big(\epsilon_k^{\mathrm{x}_1}\cdots\epsilon_k^{\mathrm{x}_d}\big)\big]\right| \\
\le& I_1 + I_2 + I_3,
\end{aligned} \tag{3.19}$$

with

$$\begin{aligned}
I_1 &= \left|\mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_2}\mathop{\mathbb{E}}_{P:\epsilon_k<a_N}\big[\log\big(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\big)\big]\right| + \left|\mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_2}\mathop{\mathbb{E}}_{\widetilde{P}:\epsilon_k<a_N}\big[\log\big(\epsilon_k^{\mathrm{x}_1}\cdots\epsilon_k^{\mathrm{x}_d}\big)\big]\right|, \\
I_2 &= \left|\mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_1}\mathop{\mathbb{E}}_{P:\epsilon_k<a_N}\big[\log\big(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\big)\big] - \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_1}\mathop{\mathbb{E}}_{\widetilde{P}:\epsilon_k<a_N}\big[\log\big(\epsilon_k^{\mathrm{x}_1}\cdots\epsilon_k^{\mathrm{x}_d}\big)\big]\right|, \\
I_3 &= \left|\mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}}\mathop{\mathbb{E}}_{P:\epsilon_k\ge a_N}\big[\log\big(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\big)\big]\right| + \left|\mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}}\mathop{\mathbb{E}}_{\widetilde{P}:\epsilon_k\ge a_N}\big[\log\big(\epsilon_k^{\mathrm{x}_1}\cdots\epsilon_k^{\mathrm{x}_d}\big)\big]\right|,
\end{aligned} \tag{3.20}$$

where $\mathop{\mathbb{E}}_{P:\epsilon_k<a_N}$ means taking expectation under the probability measure P over $\epsilon_k^{\mathrm{x}_j} < a_N, j = 1, ..., d$.

4. Finally, by bounding the three terms separately, we obtain

$$\big|\mathbb{E}[\widehat{H}_{\mathrm{tKSG}}(X)] - H(X)\big| \le C\frac{(\log N)^{k+2}}{C_1^{k+1}N^{\frac{1}{d}}}, \tag{3.21}$$

for some $C > 0$.

Proof of variance bound for the truncated KSG estimator proceeds as fllows.

1. Let $\beta_i = \sum_{j=1}^{d} \log \zeta_{i,j}(k)$, and define β_i^* (for $i = 2, ..., N$) to be the estimators with sample $\mathrm{x}^{(1)}$ removed. Next we show that $(N-1)\mathbb{E}[\mathbb{1}_{E_2}\beta_2^2]$ and $(N-1)\mathbb{E}[\mathbb{1}_{E_2}\beta_2^{*2}]$ are of the same order as $\mathbb{E}[\beta_1^2]$. As such we only need to prove that $\mathbb{E}[\beta_1^2] = O((\log N)^{k+2})$, which is done in Steps 2 and 3.

2. Separate $\mathbb{E}\big[\beta_1^2\big]$ into two parts,

$$\mathbb{E}\big[\beta_1^2\big] = \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}P:\epsilon_k<a_N} \mathbb{E}\big[\beta_1^2\big] + \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}P:\epsilon_k\geq a_N} \mathbb{E}\big[\beta_1^2\big], \tag{3.22}$$

   where $a_N = \big(\frac{2k\log N}{C_1 N}\big)^{\frac{1}{d}}$.

3. By bounding the two parts separately, we obtain the bound of the expectation of β_1^2

$$\mathbb{E}[\beta_1^2] \leq C_9(\log N)^{k+2}, \tag{3.23}$$

   for some $C_9 > 0$.

4. With the above bound, we can obtain the bound of the variance of $\widehat{H}_{tKSG}(X)$

$$\mathrm{Var}[\widehat{H}_{tKSG}(X)] \leq C'\frac{(\log N)^{k+2}}{N}, \tag{3.24}$$

   for some $C' > 0$.

□

As one can see from Theorem 3.2, while the uniform distribution leads to zero bias for $\widehat{H}_{\mathrm{tKL}}$, we can not obtain the same result for $\widehat{H}_{\mathrm{tKSG}}$, which means no theoretical justification for mapping the data points toward a uniform distribution for this estimator. That said, the tKSG estimator and Theorem 3.2 are still useful, and the reason for that is two-fold. First as is mentioned earlier, no existing result on the bound of bias is available for the KSG estimator to the best of our knowledge, and to this end our analysis on tKSG is the first known bias bound for this type of estimators, and may provide useful information for understanding the

convergence property of them. More importantly, our numerical experiments demonstrate that mapping the data points toward a uniform distribution does significantly improve the performance of tKSG as well. In fact, we have found that tKSG can achieve the same or slightly better results than tKL on the transformed samples in our test cases.

3.3.2 Estimating Entropy via Transformation

As is mentioned earlier, based on the interesting convergence properties of the truncated estimators in particularly tKL, we want to estimate the entropy of a given set of samples by mapping them toward a uniform distribution. To implement this idea, an essential question to ask is that, how the entropy of the transformed samples relates to that of the original ones. Proposition 3.1 provides an answer to this question.

Proposition 3.1 ([76]). *Let f be a mapping: $\mathcal{R}^d \to \mathcal{R}^d$, X be random variable defined on $\mathcal{R}^d$ following distribution p_{x}, and $Z = f(X)$. If f is bijective and differentiable, we have*

$$H(X) = H(Z) + \int p_{\mathrm{z}}(\mathrm{z}) \log \left| \det \frac{\partial f^{-1}(\mathrm{z})}{\partial \mathrm{z}} \right| d\mathrm{z}, \tag{3.25}$$

where $p_{\mathrm{z}}(\mathrm{z})$ is the distribution of Z.

Therefore given a data set $S = \{\mathrm{x}^{(i)}\}_{i=1}^N$ and a mapping $Z = f(X)$, from Eq. (3.25) we can construct an entropy estimator of X as,

$$\widehat{H}(X) = \widehat{H}(Z) + \frac{1}{n} \sum_{i=1}^{n} \log \left| \det \frac{\partial f^{-1}(\mathrm{z}^{(i)})}{\partial \mathrm{z}} \right|, \tag{3.26}$$

where $\widehat{H}(Z)$ is an entropy estimator of Z (either tKL or tKSG) based on the transformed samples $S_Z = \{\mathrm{z}^{(i)} = f(\mathrm{x}^{(i)})\}_{i=1}^n$.

We refer to such a mapping $f(\cdot)$ as a uniformizing mapping (UM) and the resulting methods as UM based entropy estimators where the main procedure is outlined in Algorithm 2.

A central question in the implementation of Algorithm 2 is obviously how to construct a UM which can push the samples toward a uniform distribution, which is discussed in next section.

The bias of the UM based estimators rely on the property of the UM (or equivalently the NF), on which we make the following assumption:

Assumption 3.2. *Let $S = \{\mathrm{x}^{(i)}\}_{i=1}^{N}$ be the set of i.i.d samples used to construct the UM and p_{z}^{S} be the resulting density of Z in Eq. (3.26). Denote $C_2^N = \sup_{\mathrm{z}\in\mathcal{Q}^o} ||\nabla p_{\mathrm{z}}^{S}(\mathrm{z})||_1$, and assume that C_2^N satisfies: (1) $C_2^N \xrightarrow[N\to\infty]{\mathbb{P}} 0$; (2) There exist a positive integer M and a positive real number $\bar{C} < 1$ such that:*

$$\forall N > M, \quad C_2^N \leq \bar{C}, \, a.s.$$

Based on Theorem 3.1 and Theorem 3.2, we can obtain the bias bounds and the MSE bounds of the UM based estimators.

Corollary 3.2. *Suppose that the density function of the original distribution is differentiable and the UM satisfies Assumption 3.2. The bias of UM-tKL estimator is bounded by*

$$\left|\mathbb{E}[\widehat{H}_{\mathrm{UM-tKL}}(X)] - H(X)\right| \leq C_{\mathrm{UM-tKL}}^{N}\left(\frac{k}{N}\right)^{\frac{1}{d}}, \tag{3.27}$$

where $\lim_{N\to\infty} C_{\mathrm{UM-tKL}}^{N} = 0$. The MSE of UM-tKL estimator is bounded by

$$\mathbb{E}[(\widehat{H}_{\mathrm{UM-tKL}}(X) - H(X))^2] \leq C_1\frac{1}{N} + D_{UM-tKL}^{N}\left(\frac{k}{N}\right)^{\frac{2}{d}}, \tag{3.28}$$

where C_1 is a positive constant and $\lim_{N\to\infty} D_{UM-tKL}^{N} = 0$.

Proof. See 4.2.1. □

Corollary 3.3. *Suppose that the density function of the original distribution is differentiable and the UM satisfies Assumption 3.2. The bias of UM-tKSG estimator is bounded by*

$$\left|\mathbb{E}[\widehat{H}_{\mathrm{UM-tKSG}}(X)] - H(X)\right| \leq C_{UM-tKSG}\frac{(\log N)^{k+2}}{N^{\frac{1}{d}}}, \tag{3.29}$$

where $C_{UM-tKSG} = C\frac{(1+\bar{C})\left((1+\bar{C})^d+1\right)}{(1-\bar{C})^{k+1}}$ *and* C *is a positive constant. The MSE of UM-tKSG estimator is bounded by*

$$\mathbb{E}[(\widehat{H}_{\mathrm{UM-tKSG}}(X) - H(X))^2] \leq C_2\frac{(\log N)^{k+2}}{N} + D^N_{UM-tKSG}\frac{(\log N)^{2(k+2)}}{N^{\frac{2}{d}}}, \tag{3.30}$$

where C_2 *is a positive constant and* $D^N_{UM-tKSG} = \left(C\frac{(1+\bar{C})\left((1+\bar{C})^d+1\right)}{(1-\bar{C})^{k+1}}\right)^2$.

Proof. See 4.2.2. □

These corollaries demonstrate the consistency of the proposed methods. Notably, for the UM-tKL estimator, the bias bound factor $C^N_{\mathrm{UM-tKL}}$ approaches zero as the sample size increases, suggesting that the transformation effectively reduces estimation bias. However, the specific rate of improvement remains uncertain. Furthermore, it is unclear whether the bias bound factor for the UM-tKSG estimator similarly converges to zero with increasing sample sizes. These issues warrant further investigation, which we intend to pursue in future research.

Algorithm 2 UM based entropy estimator

Input: a set of i.i.d samples: $S_X = \{\mathrm{x}^{(i)}\}$;

Output: an entropy estimate $\widehat{H}(X)$;

- compute a uniformizing map $f(\cdot)$;
- let $S_Z = \{\mathrm{z}^{(i)} = f(\mathrm{x}^{(i)}),\ i = 1, ..., n\}$;
- estimate $\widehat{H}(Z)$ from S_Z using Eq. (3.10) or Eq. (3.11);
- compute $\widehat{H}(X)$ using Eq. (3.26).

3.3.3 Constructing UM via Normalizing Flow

We discuss in this section how to construct a UM via the NF method. First since the image of f is $[0,1]^d$, we assume that f is in the form of $f = \Phi \circ g$ where $g : \mathcal{R}^d \to \mathcal{R}^d$ is learned and $\Phi : \mathcal{R}^d \to [0,1]^d$ is prescribed. Recall that p_{z} is the distribution of $Z = f(X)$ with X following p_{x}, and we want the function g by minimize the Kullback-Leibler divergence (KLD) between p_{z} and the uniform distribution p_{u}:

$$\min_{g \in \Omega} D(p_{\mathrm{z}}|p_{\mathrm{u}}) := \min_{g \in \Omega} \int p_{\mathrm{z}}(\mathrm{z}) \log \left[\frac{p_{\mathrm{z}}(\mathrm{z})}{p_{\mathrm{u}}(\mathrm{z})} \right] d\mathrm{z}, \tag{3.31}$$

where $z = \Phi \circ g(x)$ and Ω is a suitable function space. Solving Eq. (3.31) directly poses some computational difficulty as the calculation involves the function Φ, the choice of which may affect the computational efficiency. To simplify the computation, we recall the following proposition:

Proposition 3.2 ([122]). *Let $T : \mathcal{Y} \to \mathcal{Z}$ be a bijective and differentiable transformation, $p_{\mathrm{z}}(\mathrm{z})$ be the distribution obtained by passing $p_{\mathrm{y}}(\mathrm{y})$ through T, and $\pi_{\mathrm{z}}(\mathrm{z})$ be the distribution obtained by passing $\pi_{\mathrm{y}}(\mathrm{y})$ through T. Then the equality*

$$D(\pi_{\mathrm{y}}(\mathrm{y})||p_{\mathrm{y}}(\mathrm{y})) = D(\pi_{\mathrm{z}}(\mathrm{z})||p_{\mathrm{z}}(\mathrm{z})) \tag{3.32}$$

holds.

We now construct the mapping Φ with the cumulative distribution function of the standard normal distribution, a technique known as the probability integral transform, yielding, for a given $\mathrm{y} \in R^d$,

$$\Phi(\mathrm{y}) = (\phi_1(\mathrm{y}_1), ..., \phi_d(\mathrm{y}_d)), \ \phi_i(\mathrm{y}_i) = \frac{1}{2}(1 + \mathrm{erf}(\frac{\mathrm{y}}{\sqrt{2}})),$$

where $\mathrm{erf}(\cdot)$ is the error function. It should be clear that if y follows a standard normal distribution, $\mathrm{z} = \Phi(\mathrm{y})$ follows a uniform distribution in $[0,1]^d$, and vice versa. Now applying

Proposition 3.2, we can show that Eq. (3.31) is equivalent to

$$\min_{g\in\Omega} D(p_{\mathrm{y}}(\mathrm{y})|q(\mathrm{y})), \tag{3.33}$$

where $\mathrm{y} = g(\mathrm{x})$ follows distribution $p_{\mathrm{y}}(\cdot)$ and $q(\cdot)$ is the standard normal distribution. Now assume that $g(\cdot)$ is invertible and let its inverse be $h = g^{-1}$. We also assume that both g and h are differentiable. Applying Proposition 3.2 to Eq. (3.33) with $T = h$, we find that Eq. (3.33) is equivalent to

$$\min_{h\in\Omega^{-1}} D(p_{\mathrm{x}}(\mathrm{x})|q_{\mathrm{h}}(\mathrm{x})), \tag{3.34}$$

where $\Omega^{-1} = \{g^{-1}|g \in \Omega\}$ and q_{h} is the distribution obtained by passing q through the mapping h:

$$q_{\mathrm{h}}(\mathrm{x}) = q\left(h^{-1}(\mathbf{x})\right)\left|\det\left(\frac{\partial h^{-1}}{\partial \mathbf{x}}\right)\right|. \tag{3.35}$$

Eq. (3.34) essentially says that we want to push a standard normal distribution q toward a target distribution p_{x}, and therefore solving Eq. (3.34) falls naturally into the framework of NF. Specifically, NF aims to build such a mapping h by composing multiple simple mappings: $h = h_1 \circ ... \circ h_K$. Each h_k needs to be a diffeomorphism: namely it is invertible and both it and its inverse are differentiable, which ensures that their composition h is also a diffeomorphism. Next by plugging in the data, we can rewrite Eq. (3.34) as a maximum likelihood problem:

$$\max_{h=(h_1,...,h_K)} E_{p_{\mathrm{x}}}[\log q_{\mathrm{h}}(\mathrm{x})] :\approx \frac{1}{N}\sum_{i=1}^{N} \log q_{\mathrm{h}}(x^{(i)}). \tag{3.36}$$

As is mentioned earlier, the intermediate mapping h_i is usually taken to be of a simple parametrized form and so that its gradient and inverse are easy to compute. Once $h_1, ..., h_K$ are computed, the function g can be obtained as

$$g = (h_1 \circ \cdots \circ h_K)^{-1} = h_K^{-1} \circ \cdots \circ h_1^{-1}, \tag{3.37}$$

and recall that in Eq. (3.26) in Section 3.3.2 we also need the det-Jacobian of mapping g^{-1}

(i.e., h), which can be calculated as,

$$\det \frac{\partial g^{-1}(\mathrm{y})}{\partial \mathrm{y}} = \det \frac{\partial h_1(\mathrm{y}_1)}{\partial \mathrm{y}_1} \circ \cdots \circ \det \frac{\partial h_K(\mathrm{y}_K)}{\partial \mathrm{y}_K}, \tag{3.38}$$

where $\mathrm{y}_K = \mathrm{y}$, $\mathrm{y}_0 = \mathrm{x}$ and $\mathrm{y}_{k-1} = h_k(\mathrm{y}_k)$ for $k = 1, ..., K$.

The NF methods depend critically on the component layers, the choice of which has to be balanced between computational efficiency and representing flexibility. In this paper, we use a special version of NF, the Masked Autoregressive Flow (MAF) [120] that is originally designed for density estimation. Since the purpose of MAF is to estimate the density p_{x}, it is specifically designed to efficiently evaluate the inverse mappings, which is thus particularly useful for our application. We note, however, our method does not rely on any specific implementation of NF.

Once the mapping $h(\cdot)$ (or equivalently $g^{-1}(\cdot)$) is obtained, it can be inserted directly into Algorithm 1 to estimate the sought entropy. In practice, the samples are split into two sets, where one of them is used to construct the UM and the other is used to estimate the entropy.

3.4 Numerical Experiments

Before diving into the applications, we conduct several numerical comparisons of the proposed estimators using mathematical examples. The code for reproducing these examples can be found in https://github.com/ziq-ao/NFEE.

3.4.1 An Illustrating Example for the Truncated Estimators

Here we use a toy example to demonstrate the improvement of the truncated estimators over the naïve version. Specifically, the test example is an independent multivariate Beta distributions $B(b, b)$ with dimensionality d and shape parameter b. In the numerical experiments, the dimensionality is varied from 1 to 40 and the parameter b takes three values 1, 1.5 and 2. In each setup, we generate 1000 samples from the distribution and use KL, KSG, tKL and tKSG to estimate the entropy. All experiments are repeated 100 times and the Root-mean-square-error (RMSE) of estimates are computed. In Fig. 3.2, we plot the RMSE (on a logarithmic scale) against the dimensionality d. From this figure, we can see that the truncated methods (blue lines) significantly outperform the naïve ones (red lines) in all cases, indicating that the truncation technique can improve the performance of the KL/KSG estimators for compactly supported distributions.

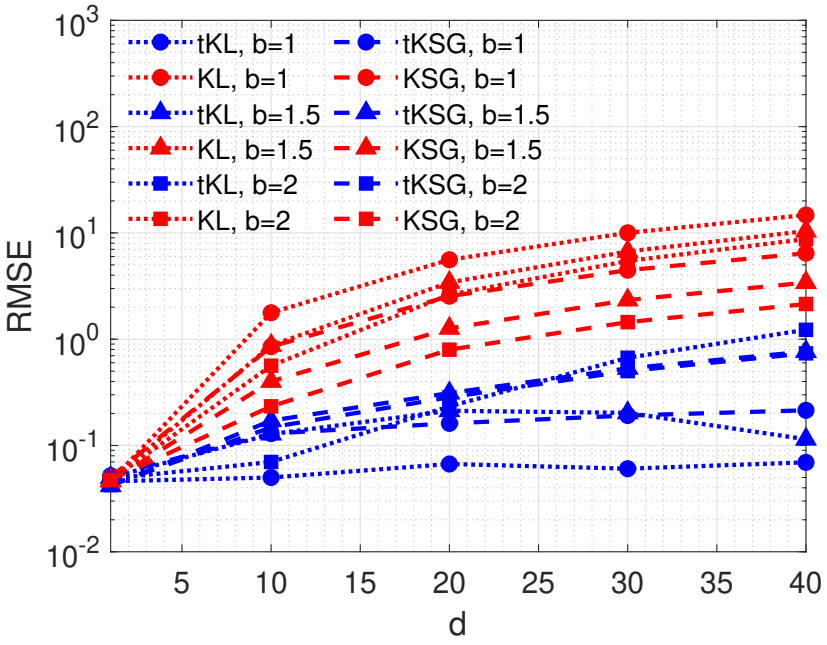


Figure 3.2: truncated estimators vs non-truncated estimators for multidimensional Beta distributions with various shape parameters b.

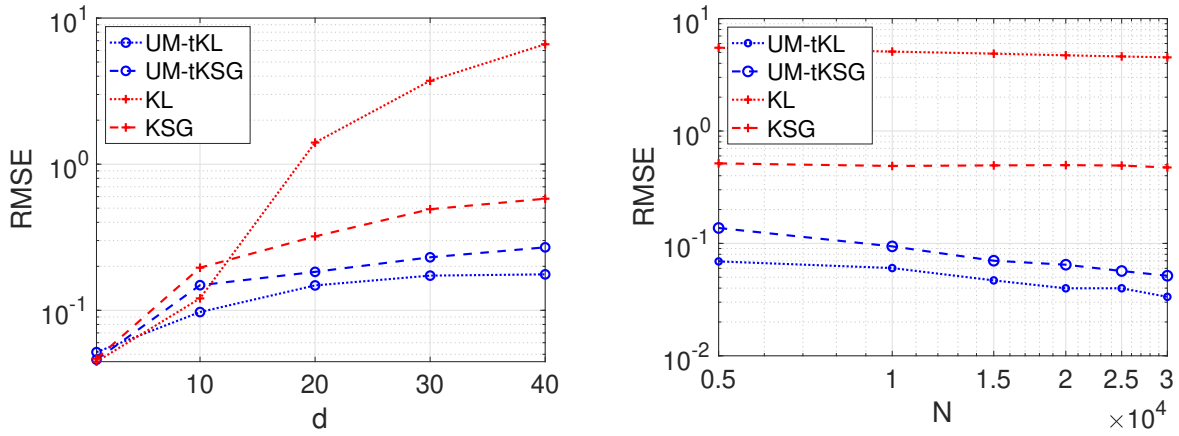


Figure 3.3: Left: RMSE plotted against the dimensionality d. Right: RMSE (on a logarithmic scale) plotted against the sample size N.

3.4.2 Multivariate Normal Distribution

To validate the idea of UM based entropy estimator, a natural question to ask is that how it works with a perfect NF transformation, that yields exactly normally distributed samples. To answer this question, we first conduct the numerical tests with the standard multivariate normal distribution, corresponding to the situation that one has done a perfect NF (in this case the function g in Section 3.3.3 is chosen to be identity map).

Specifically we test the four methods: KL, KSG, UM-tKL and UM-tKSG, and we conduct two sets of tests: in the first one we fix the sample size to be 1000 and vary the dimensionality, while in the second one we fix the dimensionality to be 40 and vary the sample size. All the tests are repeated 100 times and the RMSE of the estimates are calculated. In Fig. 3.3 (left), we plot the RMSE (on a logarithmic scale) as a function of the dimensionality. One can see from this figure that, as the dimensionality increases, the estimation error in KL and KSG grows significantly faster than that in the two UM based ones, with the error in KL being particularly large. Next in Fig. 3.3 (right) we plot the RMSE against the sample size N (note that the plot is on a log-log scale) for $d = 40$, which shows that for this high-dimensional

case, the two UM based estimators yield much lower and faster-decaying RMSE than those two estimators on the original samples. Overall these results support the theoretical findings in Section 3.3.1 that the estimation error can be significantly reduced by mapping the target samples toward a uniform distribution.

3.4.3 Multivariate Rosenbrock Distribution

In this example we shall see how the proposed method performs when NF is included. Specifically our example is the Rosenbrock type of distributions – the standard Rosenbrock distribution is 2-D and widely used as a testing example for various of statistical methods. Here we consider two high-dimensional extensions of the 2-D Rosenbrock [117]: the hybrid Rosenbrock (HR) and the even Rosenbrock (ER) distributions. The details of the two distributions including their density functions are provided in 3.6.2. The Rosenbrock distribution is strongly non-Gaussian, and that can be demonstrated by Fig. 3.4 (left) which shows the samples drawn from 2-D Rosenbrock. As a comparison, Fig. 3.4 (right) shows the samples that have been transformed toward a uniform distribution and used in entropy estimation.

In this example we compare the performance of seven estimators: in addition to the four used in the previous example, we include an estimator only using NF (details in SI) as well as two state-of-the-art entropy estimators: CADEE [10] and the von-Mises based estimator [80]. First we test how the estimators scale with respect to dimensionality, where the sample size is taken to be $N = 500d$. With each method, the experiment is repeated 20 times and the RMSE is calculated. The RMSE against the dimensionality d for both test distributions is plotted in Figs. 3.5 (a) and (b). One can observe here that in most cases, the UM based methods (especially UM-tKSG) offer the best performance. An exception is that CADEE performs better in low dimensional cases for ER, but its RMSE grows much higher than that of the UM methods in the high-dimensional regime ($d > 15$). Our second

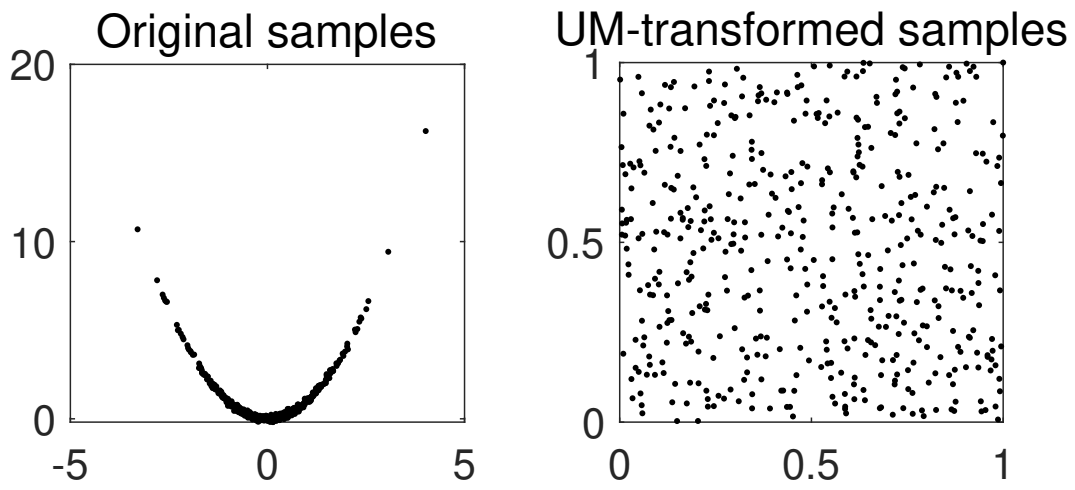


Figure 3.4: Left: the original samples drawn from a 2-D Rosenbrock distribution; Right: the UM-transformed samples used in the entropy estimation.

experiment is to fix the dimensionality at $d = 10$ and vary the sample size, where the RMSE is plotted against the sample size for both HR and ER in Figs. 3.5 (c) and (d). The figures show clearly that the RMSE of the UM based estimators decays faster than other methods in both examples, with the only exception being CADEE in the small sample ($\leq 10^4$) regime of ER. It is also worth noting that, though it is not justified theoretically, UM-tKSG seems to perform slightly better than UM-tKL in all the cases.

3.4.4 Multivariate Rosenbrock Distribution with Discontinuous Density

Recall that Corollaries 3.2 and 3.3 assume the differentiability of the original density functions, which is often not satisfied by practice. Thus, it is also of interest to examine the performance of the proposed methods for distributions with discontinuous densities. To this end, we modify the multivariate Rosenbrock distributions studied in Section 3.4.3, so that their densities are discontinuous on the boundaries of their supports (see 3.6.2 for the details), and repeat the

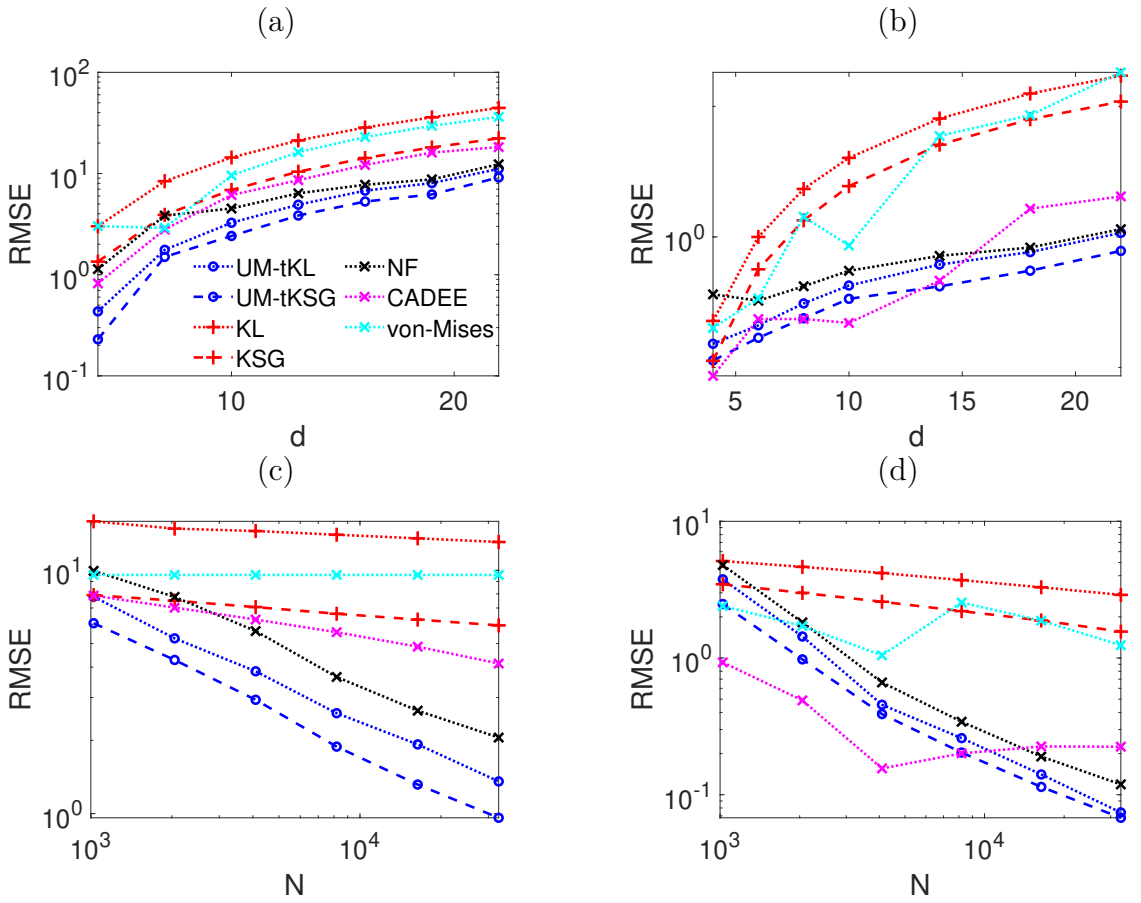


Figure 3.5: Top: RMSE vs. dimensionality for HR (a) and ER (b); Bottom: RMSE vs. sample size for HR (c) and ER (d).

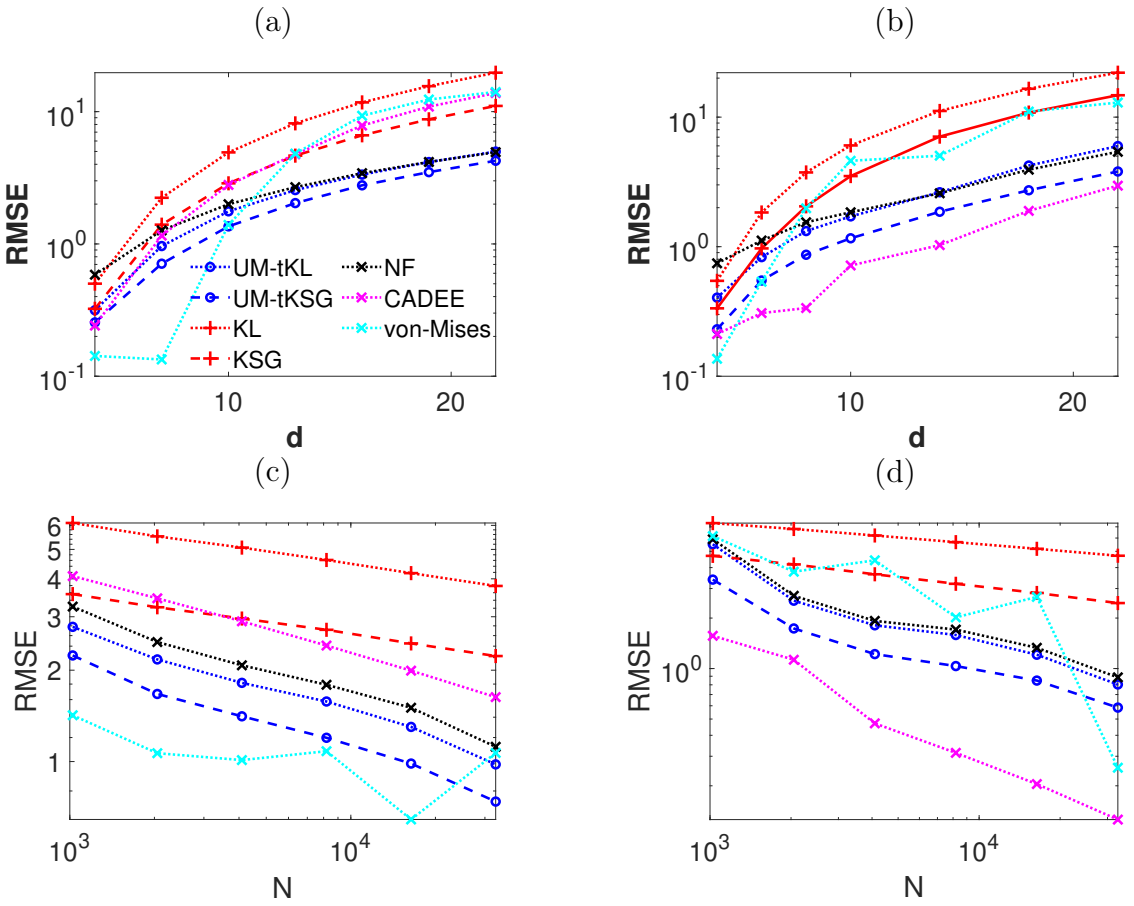


Figure 3.6: Top: RMSE vs. dimensionality for modified HR (a) and ER (b); Bottom: RMSE vs. sample size for modified HR (c) and ER (d).

comparisons conducted in Section 3.4.3. The results are shown in Figs. 3.6. For the modified HR (in Fig. 3.6 (a) and (c)), only the von-Mises estimator achieves a smaller RMSE than the UM based ones in the low-dimensional regime (d$\leq$10), while the UM based estimators perform the best in the high-dimensional regime. For modified ER (in Fig. 3.6 (b) and (d)), the UM based estimators are inferior to CADEE but outperform any other methods in most cases.

3.5 Application Examples

In this section, we consider two applications involving entropy estimation, in which our methods are compared with the existing ones.

3.5.1 Application to Entropy Rate Estimation

Our first application example is to estimate the differential entropy rate of a continuous-valued time series. Shannon entropy rate [149] measures the uncertainty of a stochastic process $\mathcal{X} = \{X_i\}_{i\in\mathbb{N}}$. For a stationary process, it is defined as,

$$\bar{H}(\mathcal{X}) = \lim_{t\to\infty} H(X_t \mid X_{t-1}, ..., X_1), \tag{3.39}$$

where $H(\cdot \mid \cdot)$ is the conditional entropy of two random variables. In this example, we consider the stochastic processes that satisfy the following two assumptions:

- First $\mathcal{X}$ is a conditionally stationary process of order p: there exists a fixed positive integer p such that, for any integer $t > p$, the conditional density function of X_t given $X_{t-1} = x_{t-1}, ..., X_{t-p} = x_{t-p}$ satisfies

  $$p(X_t = x_t \mid X_{t-1} = x_{t-1}, ..., X_{t-p} = x_{t-p}) = f(x_t \mid x_{t-1}, ..., x_{t-p}), \tag{3.40}$$

  where f is a fixed conditional density function independent from t.

- Second $\mathcal{X}$ is a Markov process of order p: there exists a positive integer p such that, for any integer $t > p$,

  $$\begin{aligned} &p(X_t = x_t \mid X_{t-1} = x_{t-1}, ..., X_1 = x_1) \\ &\qquad = p(X_t = x_t \mid X_{t-1} = x_{t-1}, ..., X_{t-p} = x_{t-p}.) \end{aligned} \tag{3.41}$$

Under these assumptions, the entropy rate of $\mathcal{X}$ can be calculated as,

$$\bar{H} = H(X_t \mid X_{(t-1):(t-p)}) = H(X_{t:(t-p)}) - H(X_{(t-1):(t-p)}), \tag{3.42}$$

where $X_{t:(t-p)} = (X_t, X_{t-1}, ..., X_{t-p})$ and so on. Note here that t can be taken to be any integer $> p$, and for simplicity we can take it to be $t = p + 1$, and as a result Eq. (3.42) is simplified to,

$$\bar{H} = H(X_t \mid X_{(t-1):(t-p)}) = H(X_{(p+1):1}) - H(X_{p:1}).$$

Suppose that we have a T-step (with $T > p$) observation of $\mathcal{X}$: $\{x_t\}_{t=1}^T$, and we can compute its entropy rate as follows [35]:

$$\hat{H} = \widehat{H}(X_{(p+1):1}) - \widehat{H}(X_{p:1}),$$

where $\widehat{H}(X_{(p+1):1})$ and $\widehat{H}(X_{p:1})$ are estimated with a desired estimator from the observation $\{x_t\}_{t=1}^T$.

In this example, we consider three autoregressive models of orders 3, 7 and 15 respectively, which are given by

$$AR(3) : X_t = -1.35 + 0.5X_{t-1} + 0.4X_{t-2}^2 - 0.3X_{t-3} + \epsilon_t, \tag{3.43a}$$

$$AR(7) : X_t = -1.35 + 0.5X_{t-1} + 0.3X_{t-5}^2 - 0.3X_{t-7} + \epsilon_t, \tag{3.43b}$$

$$\begin{aligned} AR(15) : X_t = -1.35 + 0.5X_{t-1} + 0.05(X_{t-5} + X_{t-6} + X_{t-7})^2 \\ -0.005(X_{t-11} + X_{t-12} + X_{t-13})^2 - 0.1X_{t-15} + \epsilon_t, \end{aligned} \tag{3.43c}$$

where $\epsilon_t \sim \mathcal{N}(0, (0.03)^2)$ is white noise. Fig. 3.7 shows the simulated snapshots of the three models. We implemented the procedure described above to estimate the entropy rate of these three models where the entropy is estimated with the seven estimators used in Section 3.4. On the other hand, since the conditional density functions are analytically available in this example, the entropy rate can also be directly estimated via the standard Monte Carlo integration, which will be used as the *ground truth*. We apply the aforementioned entropy

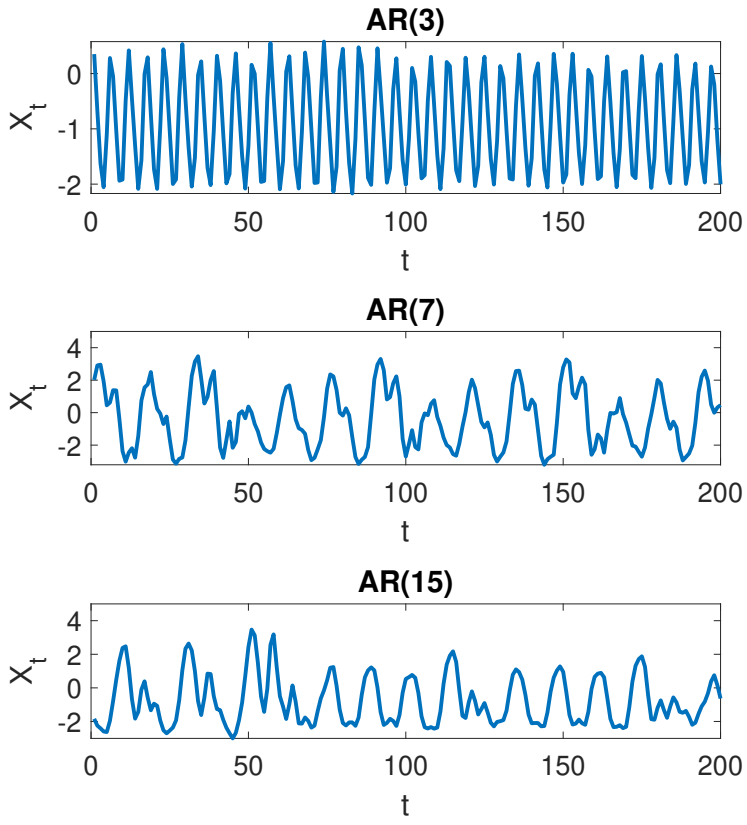


Figure 3.7: Snapshots of the simulated time series.

estimators to compute the entropy rate with a simulated sequence of $10,000$ steps. With each method, 20 repeated trials are conducted and the RMSE is calculated. The results are reported in Table 3.1, from which we make the following observations. The performance of the von-Mises estimator appears to be the best for the $AR(3)$ model, however, all estimators yield very small Root Mean Squared Error (RMSE) suggesting that this problem is not particularly challenging. For the $AR(7)$ model, the UM-based methods have smaller RMSE than the others, and for the $AR(15)$ model, the two UM-based methods and KSG perform better than the other three. Overall, UM-KSG results in the smallest RMSE for both $AR(7)$ and $AR(15)$.

Method	UM-tKL	UM-tKSG	KL	KSG	NF	CADEE	von-Mises
AR(3)	0.029	0.051	0.027	0.032	0.12	0.31	**0.016**
AR(7)	0.67	**0.43**	1.23	0.90	0.95	2.40	0.70
AR(15)	1.15	**0.68**	1.51	0.98	1.61	4.14	1.42

Table 3.1: RMSE of entropy rate estimations based on entropy estimators for the autoregressive model. The smallest (best) RMSE value is shown in bold.

3.5.2 Application to Optimal Experimental Design

In this section, we apply entropy estimation to an optimal experimental design (OED) problem. Simply put, the goal of OED is to determine the optimal experimental conditions (e.g., locations of sensors) that maximize certain utility function associated with the experiments. Mathematically let $\lambda \in \mathcal{D}$ be design parameters representing experimental conditions, θ be the parameter of interest, and Y be the observed data. An often used utility function is the entropy of the data Y, resulting in the so-called maximum entropy sampling method (MES) [146]:

$$\max_{\lambda \in \mathcal{D}} U(\lambda) := H(Y|\lambda), \tag{3.44}$$

and therefore evaluating $U(\lambda)$ becomes an entropy estimation problem. This utility function is equivalent to the mutual entropy criterion under certain conditions [150]. This formulation is particularly useful for problems with expensive or intractable likelihoods, as the likelihoods are not needed if the utility function is computed via entropy estimation. A common application of OED is to determine the observation times for stochastic processes so that one can accurately estimate the model parameters and here we provide such an example, arising from the field of population dynamics.

Specifically we consider the Lotka-Volterra (LV) predator-prey model [98, 175]. Let x and y be the populations of prey and predator respectively, and the LV model is given by

$$\dot{x} = ax - xy, \quad \dot{y} = bxy - y,$$

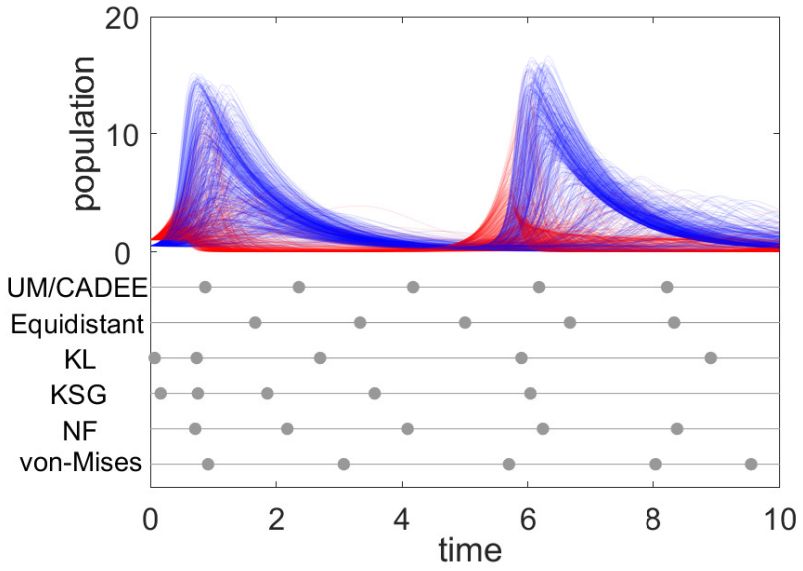


Figure 3.8: Top: some sample data paths of (x, y); Bottom: the optimal observation times obtained by the eight methods.

where a and b are respectively the growth rates of the prey and the predator. In practice, often the parameters a and b are not known and need to be estimated from the population data. In a Bayesian framework, one can assign a prior distribution on a and b, and infer them from measurements made on the population (x, y). Here we assume that the prior for both a and b is a uniform distribution $U[0.5, 4]$. In particular we assume that the pair $(x+\epsilon_x, y+\epsilon_y)$, where $\epsilon_x, \epsilon_y \sim \mathcal{N}(0, 0.01)$ are independent observation noises, is measured at $d = 5$ time points located within the interval $[0, 10]$, and the goal is to determine the observation times for the experiments. As is mentioned earlier, we shall determine the observation times using the MES method. Namely, the design parameter in this example is $\lambda = (t_1, ..., t_d)$, the data Y is the pair $(x + \epsilon_x, y + \epsilon_y)$ measured at $t_1, ..., t_d$, and we want to find λ that maximizes the entropy $H(Y|\lambda)$.

A common practice in such problems is not to optimize the observation times directly and instead parametrize them using the percentiles of a prescribed distribution to reduce

Method	UM-tKL	UM-tKSG	CADEE	Equidistant	KL	KSG	NF	von-Mises
NMC	**-1.45**			-2.73	-1.65	-1.56	-1.48	-1.81
(SE)	**(0.0073)**			(0.0074)	(0.0072)	(0.0076)	(0.0072)	(0.0049)
RMSE	**0.73**	**0.48**	0.86	—	3.60	1.05	0.88	1.31

Table 3.2: The reference entropy values of the observation time placements obtained by using all the methods. The smallest (best) entropy value is shown in bold.

the optimization dimensionality [143]. Here we use a Beta distribution, resulting in two distribution parameters to be optimized (see [143] and 3.6.4 for further details). We solve the resulting optimization problem with a grid search where the entropy is evaluated by the seven aforementioned estimators each with 10,000 samples. We plot in Fig. 3.8 the optimal observation time placements computed with the seven aforementioned estimators, as well as the equidistant placement for a comparison purpose. Also shown in the figure are some sample paths of the population (x, y) where we can see that the population samples are generally subject to larger variations near the two ends and relative smaller ones in the middle. Regarding the optimization results, we see that the optimal time placements obtained by the two UM based estimators and CADEE are the same, while they are different from the results of other methods. To validate the optimization results, we compute a reference entropy value for the optimal placement obtained by each method, using Nested Monte Carlo (NMC) (see [144] and 3.6.5 for details) with a large sample size ($10^5 \times 10^5$), and show the results in Table 3.2. Note that though the NMC can produce a rather accurate entropy estimate, it is too expensive to use directly in this OED problem. Using the reference values as the ground truth, we can further compute the RMSE of these estimates (over 20 repetitions), which are also reported in Table 3.2. From the table one observes that the placement of observation times computed by the two UM methods and CADEE yields the largest entropy values, which indicates that these three methods clearly outperform all the other estimators in this OED problem. Moreover, from the RMSE results we can see that the UM based

methods (especially UM-tKSG) yield smaller RMSE than CADEE, suggesting that they are more statistically reliable than CADEE.

3.6 Further details of the numerical examples

3.6.1 Implementation details of the estimators

The setup of MAF: We use a MAF built by 10 autoregressive layers [56] for Hybrid Rosenbrock distribution and one built by 5 autoregressive layers for Even Rosenbrock distribution and the application of experimental design. Each layer has two hidden layers of 50 units and tanh nonlinearities. In each experiment, half of the samples are used to train the MAF model and the other half are used to estimate the entropy.

The implementation of CADEE and non-Mises estimator: The two estimators are implemented using the code provided by [10] and [80] with the default parameters.

3.6.2 The two multivariate Rosenbrock distributions

Hybrid Rosenbrock Distribution. The density of the hybrid Rosenbrock distribution is given by

$$\pi(\mathbf{x}) \propto \exp\left\{-a(x_1-\mu)^2 - \sum_{j=1}^{n_2}\sum_{i=2}^{n_1} b_{j,i}(x_{j,i}-x_{j,i-1}^2)^2\right\}, \tag{3.45}$$

where the dimensionality of $\mathbf{x}$ is $d=(n_1-1)n_2+1$. The variable $x_{j,1}=x_1$ for $j=1,...,n_2$. The normalization constant of Eq. (3.45) is

$$\frac{\sqrt{a}\prod_{i=2,j=1}^{n_1,n_2}\sqrt{b_{j,i}}}{\pi^{d/2}}. \tag{3.46}$$

In this experiment, we set $\mu=1.0$, $a=1.0$, $b_{j,i}=0.1$ for all i and j, $n_1=4$ and n_2

ranging from 1 to 7. This setting forms a class of distributions with dimensions ranging from 4 to 22.

Even Rosenbrock Distribution. The density of the even Rosenbrock distribution is given by

$$\pi(\mathbf{x}) \propto \exp\left\{-\sum_{i=1}^{d/2}\left[(x_{2i-1}-\mu_{2i-1})^2 - c_i\left(x_{2i}-x_{2i-1}^2\right)^2\right]\right\}, \tag{3.47}$$

where the dimensionality d must be an even number. The normalization constant for Eq. (3.47) is

$$\frac{\prod_{i=1}^{d/2}\sqrt{c_i}}{\pi^{d/2}}. \tag{3.48}$$

In this experiment, we set $\mu_{2i-1}=0$, $c_i=12.5$ for $i=1,...,d/2$ with d ranging from 2 to 22. This setting forms a class of distributions with dimensions ranging from 2 to 22.

Hybrid Rosenbrock Distribution with Discontinuous Density. The density of the hybrid Rosenbrock distribution with discontinuous density is given by

$$\pi(\mathbf{x}) = \text{unifpdf}(x_1, \mu, \sqrt{\frac{1}{8a}}) \times \prod_{j=1}^{n_2}\prod_{i=2}^{n_1}\text{unifpdf}(x_{j,i}, x_{j,i-1}^2, \sqrt{\frac{1}{8b}}) \tag{3.49}$$

where $\text{unifpdf}(x,\alpha,\beta)$ is the pdf of the continuous uniform distribution on the interval $[\alpha-\beta,\alpha+\beta]$, evaluated at the values in x, and where the dimensionality of $\mathbf{x}$ is $d=(n_1-1)n_2+1$. The variable $x_{j,1}=x_1$ for $j=1,...,n_2$.

In this experiment, we set $\mu=1.0$, $a=1.0$, $b_{j,i}=0.1$ for all i and j, $n_1=4$ and n_2 ranging from 1 to 7. This setting forms a class of distributions with dimensions ranging from 4 to 22.

Even Rosenbrock Distribution with Discontinuous Density. The density of the even Rosenbrock distribution with discontinuous density is given by

$$\pi(\mathbf{x}) = \prod_{i=1}^{d/2}\left[\text{unifpdf}(x_{2i-1}, \mu_{2i-1}, 0.5) \times \text{unifpdf}(x_{2i}, x_{2i-1}^2, c_i)\right], \tag{3.50}$$

where the dimensionality d must be an even number.

In this experiment, we set $\mu_{2i-1} = 0$, $c_i = 0.025$ for $i = 1, ..., d/2$ with d ranging from 2 to 22. This setting forms a class of distributions with dimensions ranging from 2 to 22.

3.6.3 Entropy estimator only using NF

In this section we describe a simplified version of the proposed method, which estimate the entropy only using NF (without the truncated entropy estimators). To start with, we recall Eq. (3.25) in Section 3.3.2,

$$H(X) = H(Z) + \int p_{\mathrm{z}}(\mathrm{z}) \log \left| \det \frac{\partial f^{-1}(\mathrm{z})}{\partial \mathrm{z}} \right| d\mathrm{z}. \tag{3.51}$$

The main idea of this simplified method is to assume that the transformed random variable Z exactly follows a uniform distribution and as a result $H(Z) = 0$. Therefore the entropy of X is estimated as,

$$\hat{H}_{NF}(X) = \frac{1}{n} \sum_{i=1}^{n} \log \left| \det \frac{\partial f^{-1}(\mathrm{z}^{(i)})}{\partial \mathrm{z}} \right|, \tag{3.52}$$

where $\mathrm{z}^{(i)} = f(\mathrm{x}^{(i)})$. A limitation of this method is quite obvious – the transformed random variable Z is usually not uniformly distributed and simply taking its entropy to be zero will undoubtedly introduce bias, which is demonstrated by the numerical examples in the previous subsections. It should also be noted that, while not in the context of entropy estimation, an NF based approach has been used for maximum entropy modeling [94].

3.6.4 The Beta scheme for parametrizing the observation times

In the optimal experimental design (OED) example, we use a lower dimensional parameterization scheme to reduce the dimensionality of the optimization problem [143]. In particular we use the Beta scheme [143] to allocate the placements of the observation times. Specifically,

let $Q(\cdot, \alpha, \beta)$ be the quantile function of the beta distribution with shape parameters α and β, and the d observation times $\lambda = (t_1, ..., t_d)$ in the time interval $[0, T]$ are allocated as,

$$t_i = T \cdot Q(\frac{i}{d+1}, \alpha, \beta), \quad i = 1, ..., d. \tag{3.53}$$

As such the d-dimensional variable λ is parametrized by $\alpha > 0$ and $\beta > 0$.

3.6.5 Nested Monte Carlo

Here we describe the Nested Monte Carlo (NMC) approach that is used to estimate the entropy in the experimental design example. Recall that the entropy of interest is $H(Y)$ (here for simplicity we omit the design parameter λ):

$$H(Y) = \int \log p(y) p(y) dy, \tag{3.54}$$

which can be estimated via Monte Carlo (MC):

$$H(Y) \approx -\frac{1}{M} \sum_{i=1}^{M} \log p(y^{(i)}), \tag{3.55}$$

where $y^{(i)}$ are drawn from $p(y)$. A difficulty here is that we do not have an explicit expression of $p(y)$. Note however that in this example the likelihood $p(y|\theta)$ and the prior $p(\theta)$ are available and we can therefore write

$$p(y) = \int p(y|\theta) p(\theta) d\theta. \tag{3.56}$$

It follows that $p(y)$ can also be estimated via MC:

$$p(y^{(i)}) \approx \frac{1}{N} \sum_{j=1}^{N} p(y^{(i)}|\theta^{(j)}), \tag{3.57}$$

where $\theta^{(j)}$ are drawn from $p(\theta)$. Combining Eq. (3.57) and Eq. (3.55), we obtain an estimator of $H(Y)$, which is referred to as the NMC method [144]. In particular, Eq. (3.57) is usually referred to as the inner MC and Eq. (3.55) is referred to as the outer one. Since the theoretical

results in [144, 135] show that the mean squared error of NMC estimator decays at a rate of $O(\frac{1}{M} + \frac{1}{N})$, we can obtain an accurate evaluation of $H(Y)$ with a sufficiently large number of samples, and in the numerical example we use $M = N = 1 \times 10^5$. We emphasize that such a large number of samples is not computationally feasible to use in the experimental design procedure, and thus in the example we have to resort to other entropy estimation methods.

3.7 Conclusion

In summary, we have presented a uniformization based entropy estimator, and also provided some theoretical analysis of it. We believe the proposed entropy estimator can be useful for a wide range of real-world applications. Some improvements and extensions of the method are possible. First while our theoretical results provide some justification for the method, further analysis is needed to establish the convergence rate and understand the estimation bias. Additionally, the method may be extended to estimate other density functionals, such as the Renyi entropy and the Kullback-Leibler divergence. Finally in this work the proposed method is demonstrated only with synthetic data, and it is therefore sensible to further examine the method with real-world data sets. We will explore these research problems in future studies.

Chapter Four

Convergence Analyses for Entropy Estimation via Uniformization

4.1 Convergence analyses for the truncated KL and KSG estimators

In this section, we present proofs for Theorems 3.1 and 3.2, closely following the framework in [153] and [54] concerning finite-sample analysis of fixed k nearest neighbor entropy estimators. These studies established bias bounds of roughly $O((\frac{1}{N})^{\gamma/d})$ (γ is some positive constant) and variance bounds of roughly $O(\frac{1}{N})$ for the entropy estimator $\widehat{H}_{KL}$, under some mild assumptions. Extending this framework, we demonstrate that our proposed estimators $\widehat{H}_{tKL}$ and $\widehat{H}_{tKSG}$ exhibit similar bias and variance bounds. Notably, our analysis draws an intriguing connection between the bias bound of $\widehat{H}_{tKL}$ and the gradient of the density function.

4.1.1 Definitions and assumptions

In this section, we introduce some notations and assumptions that the proofs rely on. As is mentioned in the main paper, we only consider distributions with densities supported on the unit cube in $\mathbb{R}^d$. Let $\mathcal{Q} := [0,1]^d$ denote the unit cube in d-dimensional Euclidean space $\mathbb{R}^d$ and P denote an unknown μ-absolutely continuous Borel probability measure, where μ is the Lebesgue measure. Let $p : \mathcal{Q} \to [0,\infty)$ be the density of P.

Definition 4.1 (Twice the k-NN distance for cubes)**.** *Suppose* $\{\mathrm{x}^{(i)}\}_{i=1}^{N-1}$ *is set of* $N-1$ *i.i.d. samples from* P*. We define twice the maximum-norm k-NN distance for cubes by* $\epsilon_k(\mathrm{x}) = 2||\mathrm{x}-\mathrm{x}^*||_\infty$*, where* x^* *is the k-nearest element amongst* $\{\mathrm{x}^{(i)}\}_{i=1}^{N-1}$ *to* x *with respect to* ∞*-norm.*

Definition 4.2 (Twice the k-NN distance for rectangles)**.** *Suppose* $\{\mathrm{x}^{(1')},...,\mathrm{x}^{(k')}\}$ *is set of the k nearest elements amongst* $\{\mathrm{x}^{(i)}\}_{i=1}^{N-1}$ *to* x *with respect to* ∞*-norm. We define twice the k-NN distance in the marginal direction* x_j *by* $\epsilon_k^{\mathrm{x}_j}(\mathrm{x}) = 2|\mathrm{x}_j - \mathrm{x}_j^{*j}|$*, where* x^{*j} *is the k-nearest element amongst* $\{\mathrm{x}^{(1')},...,\mathrm{x}^{(k')}\}$ *in the marginal direction* x_j *to* x*. It should be noted that* $\epsilon_k(\mathrm{x}) = \max\limits_{1\le j\le d} \epsilon_k^{\mathrm{x}_j}(\mathrm{x})$.

Definition 4.3 (Truncated twice the k-NN distance)**.** *Since we only consider densities supported on the unit cube, we define so-called truncated distance for convenience. In the cubic case, we define truncated twice the k-NN distance in the marginal direction* x_j *by* $\xi_k^{\mathrm{x}_j}(\mathrm{x}) = \min\{\mathrm{x}_j + \epsilon_k(\mathrm{x})/2, 1\} - \max\{\mathrm{x}_j - \epsilon_k(\mathrm{x})/2, 0\}$*. In the rectangular case, such distance in the marginal direction* x_j *is defined by* $\zeta_k^{\mathrm{x}_j}(\mathrm{x}) = \min\{\mathrm{x}_j+\epsilon_k^{\mathrm{x}_j}(\mathrm{x})/2, 1\}-\max\{\mathrm{x}_j-\epsilon_k^{\mathrm{x}_j}(\mathrm{x})/2, 0\}$.

Definition 4.4 (r-cell)**.** *We define the* r*-cell centered at* x *by* $B(\mathrm{x}; r) = \{\mathrm{x}' \in \mathbb{R}^d : ||\mathrm{x}'-\mathrm{x}||_\infty < r\}$ *in the cubic case, and by* $B(\mathrm{x}; r_{1:d}) = \bigcap\limits_{j=1}^{d} \{\mathrm{x}' \in \mathbb{R}^d : |\mathrm{x}'_j - \mathrm{x}_j| < r_j\}$ *in the rectangular case.*

Definition 4.5 (Truncated r-cell)**.** *We define the truncated* r*-ball centered at* x *by* $\overline{B}(\mathrm{x}; r) = \mathcal{Q} \cap B(\mathrm{x}; r)$ *in the cubic case, and by* $\overline{B}(\mathrm{x}; r_{1:d}) = \mathcal{Q} \cap B(\mathrm{x}; r_{1:d})$ *in the rectangular case.*

Definition 4.6 (Mass function)**.** *We define the mass of the cell $B(\mathrm{x}; r/2)$ as a function with respect to r, which is given by $p_r(\mathrm{x}) = P(B(\mathrm{x}; r/2))$, and define the mass of the cell $B(\mathrm{x}; r_{1:d}/2)$ as a function with respect to $r_1, ..., r_d$, which is given by $q_{r_1,...,r_d}(\mathrm{x}) = P(B(\mathrm{x}; r_{1:d}/2))$.*

Assumption 4.1. *We make the following assumptions:*

(a) p is continuous and supported on $\mathcal{Q}$;

(b) p is bounded away from 0, i.e., $C_1 = \inf_{\mathrm{x}\in\mathcal{Q}} p(\mathrm{x}) > 0$;

(c) The gradient of p is uniformly bounded on $\mathcal{Q}^o$, i.e., $C_2 = \sup_{\mathrm{x}\in\mathcal{Q}^o} ||\nabla p(\mathrm{x})||_1 < \infty$.

4.1.2 Preliminary lemmas

Here, we present some lemmas that support the proofs of the main results.

Lemma 4.1 ([85])**.** *The expectation of $\log p_{\epsilon_k}(\mathrm{x})$ satisfies*

$$\mathbb{E}[\log p_{\epsilon_k}(\mathrm{x})] = \psi(k) - \psi(N).$$

Lemma 4.2. *Let $\widetilde{P}$ be the probability measure of a uniform distribution supported on a d-dimensional (hyper-)cubic area $S := B(\mathrm{x}; l/2)$, and $\widetilde{p}(\mathrm{x}) = \frac{1}{l^d}, \mathrm{x} \in S$ be the density function. Define $\widetilde{q}_{r_1,...,r_d}(\mathrm{x}) = \widetilde{P}(B(\mathrm{x}; r_1/2, ..., r_d/2))$ and $\widetilde{p}_r(\mathrm{x}) = \widetilde{P}(B(\mathrm{x}; r/2))$. Then, we have*

$$\mathbb{E}[\log \widetilde{q}_{\epsilon_k^{\mathrm{x}_1},...,\epsilon_k^{\mathrm{x}_d}}(\mathrm{x})] = \psi(k) - \frac{d-1}{k} - \psi(N),$$

where $\epsilon_k^{\mathrm{x}_j}, j = 1, ..., d$ are defined as Definition 4.2 after replacing P by $\widetilde{P}$.

Proof. The probability density function for $(\epsilon_k^{\mathrm{x}_1}, ..., \epsilon_k^{\mathrm{x}_d})$ is given by,

$$f_{N,k}(r_1, ..., r_d) = \frac{(N-1)!}{k!(N-k-1)!} \times \frac{\partial^d(\widetilde{q}^k_{r_1,...,r_d})}{\partial r_1 \cdots \partial r_d} \times (1 - \widetilde{p}_{r_{\mathrm{m}}})^{N-k-1}, \tag{4.1}$$

where $\widetilde{p}_r = \widetilde{P}(B(\mathrm{x}; r/2))$, and $r_{\mathrm{m}} = \max_{1\le j\le d} r_j$ [85]. Then we have

$$\begin{aligned}
\mathbb{E}[\log \widetilde{q}_{\epsilon_k^{\mathrm{x}_1},\ldots,\epsilon_k^{\mathrm{x}_d}}(\mathrm{x})] &= \int_0^l \cdots \int_0^l \binom{N-1}{k} \cdot \frac{\partial^d(\widetilde{q}^k_{r_1,\ldots,r_d})}{\partial r_1 \cdots \partial r_d} \cdot (1-\widetilde{p}_{r_{\mathrm{m}}})^{N-k-1} \log \widetilde{q}_{r_1,\ldots,r_d} dr_1 \cdots dr_d \\
&= \int_0^l \cdots \int_0^l \binom{N-1}{k} \cdot \frac{\partial^d\big((\frac{1}{l^d} r_1 \cdots r_d)^k\big)}{\partial r_1 \cdots \partial r_d} \cdot (1-\frac{1}{l^d} r_{\mathrm{m}}^d)^{N-k-1} \log(\frac{1}{l^d} r_1 \cdots r_d) dr_1 \cdots dr_d \\
&= \binom{N-1}{k} k^d \frac{1}{l^d} \int_0^l \cdots \int_0^l (\frac{1}{l^d} r_1 \cdots r_d)^{k-1} (1-\frac{1}{l^d} r_{\mathrm{m}}^d)^{N-k-1} \log(\frac{1}{l^d} r_1 \cdots r_d) dr_1 \cdots dr_d \\
&= \binom{N-1}{k} k^d \int_0^1 \cdots \int_0^1 (u_1 \cdots u_d)^{k-1} (1-u_{\mathrm{m}}^d)^{N-k-1} \log(u_1 \cdots u_d) du_1 \cdots du_d,
\end{aligned} \tag{4.2}$$

where the last equality comes from the change of variables $u_i = \frac{1}{l} r_i, i = 1, ..., d$. Note that the integrand is symmetric under a permutation of the labels $1, ..., d$, and so we have

$$\begin{aligned}
&\mathbb{E}[\log \widetilde{q}_{\epsilon_k^{\mathrm{x}_1},\ldots,\epsilon_k^{\mathrm{x}_d}}(\mathrm{x})] \\
=&dk^d \binom{N-1}{k} \int_0^1 du_d \Big(u_d^{k-1} (1-u_d^d)^{N-k-1} \int_0^{u_d} \cdots \int_0^{u_d} (u_1 \cdots u_{d-1})^{k-1} \log(u_1 \cdots u_d) du_1 \cdots du_{d-1} \Big)
\end{aligned} \tag{4.3}$$

Computing the integral over $u_1, ..., u_{d-1}$ using the symmetry again, we obtain

$$\begin{aligned}
&\int_0^{u_d} \cdots \int_0^{u_d} (u_1 \cdots u_{d-1})^{k-1} \log(u_1 \cdots u_d) du_1 \cdots du_{d-1} \\
=&(d-1) \int_0^{u_d} \cdots \int_0^{u_d} (u_1 \cdots u_{d-1})^{k-1} \log u_1 du_1 \cdots du_{d-1} \\
&\qquad + \log u_m \int_0^{u_d} \cdots \int_0^{u_d} (u_1 \cdots u_{d-1})^{k-1} du_1 \cdots du_{d-1} \\
=&I_1 + I_2,
\end{aligned} \tag{4.4}$$

where I_1 and I_2 represent the first and second terms of the equation respectively. By basic

calculus, we have

$$\begin{aligned} I_1 &= (d-1)\int_0^{u_d} u_1^{k-1}\log u_1 du_1 \Big(\int_0^{u_d}(u_2)^{k-1}du_2\Big)^{d-2} \\ &= (d-1)\big(\frac{1}{k}u_d^k\big)^{d-1}(\log u_d - \frac{1}{k}), \end{aligned} \tag{4.5}$$

and

$$I_2 = \log u_d (\frac{1}{k}u_d^k)^{d-1}, \tag{4.6}$$

which yield $I_1 + I_2 = (\frac{1}{k}u_d^k)^{d-1}\big(d\log u_d - \frac{d-1}{k}\big)$. Plug this into Eq (4.3) and change the variables by $t = u_d^d$, and we finally have

$$\begin{aligned} &\mathbb{E}[\log \widetilde{q}_{\epsilon_k^{\mathrm{x}_1},\ldots,\epsilon_k^{\mathrm{x}_d}}(\mathrm{x})] \\ =&dk\binom{N-1}{k}\int_0^1 u_d^{kd-1}(1-u_d^d)^{N-k-1}\big(d\log u_d - \frac{d-1}{k}\big)du_d \\ =&k\binom{N-1}{k}\int_0^1 t^{k-1}(1-t)^{N-k-1}\big(\log t - \frac{d-1}{k}\big)dt \\ =&\psi(k) - \frac{d-1}{k} - \psi(N). \end{aligned} \tag{4.7}$$

□

Lemma 4.3 (Lemma 3 in [153]). *Suppose p satisfies Assumption (a) and (b). Then, for any $\mathrm{x} \in \mathcal{Q}$ and $r > \big(\frac{k}{C_1 N}\big)^{1/d}$, we have*

$$\mathbb{P}(\epsilon_k(\mathrm{x}) > r) \le e^{-C_1 r^d N}\big(\frac{eC_1 r^d N}{k}\big)^k.$$

Lemma 4.4 (Lemma 4 in [153]). *Suppose p satisfies Assumption (a) and (b). Then, for any $\mathrm{x} \in \mathcal{Q}$ and $\alpha > 0$, we have*

$$\mathbb{E}[\epsilon_k^\alpha(\mathrm{x})] \le (1+\frac{\alpha}{d})\big(\frac{k}{C_1 N}\big)^{\frac{\alpha}{d}}.$$

Lemma 4.5. *Suppose p satisfies Assumption 4.1, then, for any $\mathrm{x} \in \mathcal{Q}$ and array $(r_1, ..., r_d)$ that satisfy*

$$\begin{cases} \mathrm{x}_j + \frac{r_j}{2} \le 1, if\ \mathrm{x}_j \le \frac{1}{2} \\ \mathrm{x}_j - \frac{r_j}{2} \ge 0, if\ \mathrm{x}_j > \frac{1}{2} \end{cases}$$

for $j = 1, ..., d$, we have

$$\left| \frac{\partial^d q_{r_1,...,r_d}(\mathrm{x})}{\partial r_1 \cdots \partial r_d} - \frac{1}{2^{\sum_{j=1}^d \mathbb{1}_j}} p(\mathrm{x}) \right| \leq \frac{1}{2^{\sum_{j=1}^d \mathbb{1}_j + 1}} C_2 r_{\mathrm{m}},$$

and

$$\left| \frac{\partial^u q_{r_1,...,r_d}(\mathrm{x})}{\partial r_1 \cdots \partial r_u} - \frac{1}{2^{\sum_{j=1}^u \mathbb{1}_j}} p(\mathrm{x}) \mu\big(\overline{B}(\mathrm{x}_{u+1:d}; \frac{r_{u+1}}{2}, ..., \frac{r_d}{2})\big) \right| \leq \frac{1}{2^{\sum_{j=1}^u \mathbb{1}_j + 1}} C_2 r_{\mathrm{m}} \mu\big(\overline{B}(\mathrm{x}_{u+1:d}; \frac{r_{u+1}}{2}, ..., \frac{r_d}{2})\big),$$

where $u < d$, $r_{\mathrm{m}} = \max_{1 \leq j \leq d} r_j$ and $\mathbb{1}_j$ is the indicator function admitting the value 1 if the interval $[\mathrm{x}_j - \frac{r_j}{2}, \mathrm{x}_j + \frac{r_j}{2}]$ contains 0 or 1 and 0 otherwisely.

Proof. For the sake of convenience, we only discuss the case when $\mathrm{x} \in [0, \frac{1}{2}]^d$ and $\mathbb{1}_j = 1$ for $j = 1, ..., n \leq u$. The proof for other cases can be obtained by permuting the labels $1, ..., d$. By the definition of $q_{r_1,...,r_d}(\mathrm{x})$, we have

$$\begin{aligned} q_{r_1,...,r_d}(\mathrm{x}) &= \int_{\mathrm{x}_1 - r_1/2}^{\mathrm{x}_1 + r_1/2} \cdots \int_{\mathrm{x}_d - r_d/2}^{\mathrm{x}_d + r_d/2} p(\mathrm{x}'_1, ..., \mathrm{x}'_d) d\mathrm{x}'_d \cdots d\mathrm{x}'_1 \\ &= \int_0^{\mathrm{x}_1 + r_1/2} \cdots \int_0^{\mathrm{x}_n + r_n/2} \int_{\mathrm{x}_{n+1} - \frac{r_{n+1}}{2}}^{\mathrm{x}_{n+1} + \frac{r_{n+1}}{2}} \cdots \int_{\mathrm{x}_d - r_d/2}^{\mathrm{x}_d + r_d/2} p(\mathrm{x}'_1, ..., \mathrm{x}'_d) d\mathrm{x}'_d \cdots d\mathrm{x}'_1, \end{aligned} \tag{4.8}$$

and the partial derivative of it with respect to the first n variables is given by

$$\begin{aligned} &\frac{\partial^n q_{r_1,...,r_d}(\mathrm{x})}{\partial r_1 \cdots \partial r_n} \\ =&\frac{1}{2^n} \int_{\mathrm{x}_{n+1} - \frac{r_{n+1}}{2}}^{\mathrm{x}_{n+1} + \frac{r_{n+1}}{2}} \cdots \int_{\mathrm{x}_d - r_d/2}^{\mathrm{x}_d + r_d/2} p(\mathrm{x}_1 + \frac{r_1}{2}, ..., \mathrm{x}_n + \frac{r_n}{2}, \mathrm{x}'_{n+1}, ..., \mathrm{x}'_d) d\mathrm{x}'_d \cdots d\mathrm{x}'_{n+1}. \end{aligned} \tag{4.9}$$

Next we obtain the partial derivative of $q_{r_1,...,r_d}(\mathrm{x})$ with respect to the first u variables

$$\begin{aligned} &\frac{\partial^u q_{r_1,...,r_d}(\mathrm{x})}{\partial r_1 \cdots \partial r_u} \\ =&\frac{1}{2^u} \int_{\mathrm{x}_{u+1} - r_{u+1}/2}^{\mathrm{x}_{u+1} + r_{u+1}/2} \cdots \int_{\mathrm{x}_d - r_d/2}^{\mathrm{x}_d + r_d/2} p(\mathrm{x}_1 + \frac{r_1}{2}, ..., \mathrm{x}_n + \frac{r_n}{2}, \mathrm{x}_{n+1} \pm \frac{r_{n+1}}{2}, ..., \mathrm{x}_u \pm \frac{r_u}{2}, \mathrm{x}'_{u+1}, ..., \mathrm{x}'_d) \\ &\qquad d\mathrm{x}'_{u+1} \cdots d\mathrm{x}'_d \\ =&\frac{1}{2^u} \int_{\overline{B}(\mathrm{x}_{u+1:d}; \frac{r_{u+1}}{2}, ..., \frac{r_d}{2})} p(\mathrm{x}_1 + \frac{r_1}{2}, ..., \mathrm{x}_n + \frac{r_n}{2}, \mathrm{x}_{n+1} \pm \frac{r_{n+1}}{2}, ..., \mathrm{x}_u \pm \frac{r_u}{2}, \mathrm{x}'_{u+1}, ..., \mathrm{x}'_d) \\ &\qquad d\mathrm{x}'_{u+1} \cdots d\mathrm{x}'_d, \end{aligned} \tag{4.10}$$

where the notation $p(..., x \pm \frac{r}{2}, ...) = p(..., x + \frac{r}{2}, ...) + p(..., x - \frac{r}{2}, ...)$.

Finally, we have

$$\begin{aligned}
&\left| \frac{\partial^u q_{r_1,...,r_d}(\mathrm{x})}{\partial r_1 \cdots \partial r_u} - \frac{1}{2^{\sum_{j=1}^u \mathbb{1}_j}} p(\mathrm{x}) \mu\big(\overline{B}(\mathrm{x}_{u+1:d}; \frac{r_{u+1}}{2}, ..., \frac{r_d}{2})\big) \right| \\
\leq & \frac{1}{2^u} \int_{\overline{B}(\mathrm{x}_{u+1:d}; \frac{r_{u+1}}{2}, ..., \frac{r_d}{2})} \Big| p(\mathrm{x}_1 + \frac{r_1}{2}, ..., \mathrm{x}_n + \frac{r_n}{2}, \mathrm{x}_{n+1} \pm \frac{r_{n+1}}{2}, ..., \mathrm{x}_u \pm \frac{r_u}{2}, \mathrm{x}'_{u+1}, ..., \mathrm{x}'_d) \\
& \qquad - 2^{u-n} p(\mathrm{x}) \Big| d\mathrm{x}'_{u+1} \cdots d\mathrm{x}'_d \qquad (4.11) \\
\leq & \frac{2^{u-n}}{2^u} \int_{\overline{B}(\mathrm{x}_{u+1:d}; \frac{r_{u+1}}{2}, ..., \frac{r_d}{2})} C_2 \frac{r_{\mathrm{m}}}{2} d\mathrm{x}'_{u+1} \cdots d\mathrm{x}'_d \\
= & \frac{1}{2^{n+1}} C_2 r_{\mathrm{m}} \mu\big(\overline{B}(\mathrm{x}_{u+1:d}; \frac{r_{u+1}}{2}, ..., \frac{r_d}{2})\big),
\end{aligned}$$

which completes the proof for $u < d$.

Particularly, we have

$$\left| \frac{\partial^d q_{r_1,...,r_d}(\mathrm{x})}{\partial r_1 \cdots \partial r_d} - \frac{1}{2^{\sum_{j=1}^d \mathbb{1}_j}} p(\mathrm{x}) \right| \leq \frac{1}{2^{\sum_{j=1}^d \mathbb{1}_j + 1}} C_2 r_{\mathrm{m}}. \qquad (4.12)$$

□

Lemma 4.6. *Suppose p satisfies Assumption 4.1, then, for any $\mathrm{x} \in \mathcal{Q}$ and r that satisfy*

$$\begin{cases} \mathrm{x}_j + \frac{r}{2} \leq 1, if\ \mathrm{x} \leq \frac{1}{2} \\ \mathrm{x}_j - \frac{r}{2} \geq 0, if\ \mathrm{x} > \frac{1}{2} \end{cases}$$

for $j = 1, ..., d$, we have

$$\left| p_r(\mathrm{x}) - p(\mathrm{x}) \mu\big(\overline{B}(\mathrm{x}; \frac{r}{2})\big) \right| \leq C_2 \frac{r}{2} \overline{B}(\mathrm{x}; \frac{r}{2}),$$

and

$$\left| \frac{dp_r(\mathrm{x})}{dr} - \sum_{j=1}^d \frac{1}{2^{\mathbb{1}_j}} p(\mathrm{x}) \mu\big(\overline{B}(\mathrm{x}_{\hat{j}}; \frac{r}{2})\big) \right| \leq \sum_{j=1}^d \frac{1}{2^{\mathbb{1}_j + 1}} C_2 r \mu\big(\overline{B}(\mathrm{x}_{\hat{j}}; \frac{r}{2})\big),$$

where $m < d$ and $\mathbb{1}_j$ is the indicator function admitting the value 1 if $[\mathrm{x}_j - \frac{r}{2}, \mathrm{x}_j + \frac{r}{2}]$ intersects $[0, 1]$ and 0 otherwiesly.

Proof. By the definition of $p_r(\mathrm{x})$, we have

$$p_r(\mathrm{x}) = \int_{\overline{B}(\mathrm{x};\frac{r}{2})} p(\mathrm{x}'_1, ..., \mathrm{x}'_d) d\mathrm{x}'_d \cdots d\mathrm{x}'_1. \tag{4.13}$$

It then follows that,

$$\begin{aligned}
&\left| p_r(\mathrm{x}) - p(\mathrm{x})\mu\big(\overline{B}(\mathrm{x};\frac{r}{2})\big) \right| \\
\leq &\int_{\overline{B}(\mathrm{x};\frac{r}{2})} \big| p(\mathrm{x}'_1, ..., \mathrm{x}'_d) - p(\mathrm{x}) \big| d\mathrm{x}'_d \cdots d\mathrm{x}'_1 \\
\leq &\int_{\overline{B}(\mathrm{x};\frac{r}{2})} C_2 \frac{r}{2} d\mathrm{x}'_d \cdots d\mathrm{x}'_1 \\
= &C_2 \frac{r}{2} \overline{B}(\mathrm{x};\frac{r}{2}),
\end{aligned} \tag{4.14}$$

which completes proof of the first inequality. For the second inequality, one can easily see that

$$p_r(\mathrm{x}) = q_{r,...,r}(\mathrm{x}). \tag{4.15}$$

Now using Lemma 4.5, we obtain

$$\begin{aligned}
&\left| \frac{dp_r(\mathrm{x})}{dr} - \sum_{j=1}^{d} \frac{1}{2^{\mathbb{1}_j}} p(\mathrm{x}) \mu\big(\overline{B}(\mathrm{x}_{\hat{j}};\frac{r}{2})\big) \right| \\
\leq &\sum_{j=1}^{d} \left| \frac{\partial q_{r_1,...,r_d}(\mathrm{x})}{\partial r_j}\Big|_{r_{1:d}=r} - \frac{1}{2^{\mathbb{1}_j}} p(\mathrm{x}) \mu\big(\overline{B}(\mathrm{x}_{\hat{j}};\frac{r}{2})\big) \right| \\
\leq &\sum_{j=1}^{d} \frac{1}{2^{\mathbb{1}_j+1}} C_2 r \mu\big(\overline{B}(\mathrm{x}_{\hat{j}};\frac{r}{2})\big).
\end{aligned} \tag{4.16}$$

□

4.1.3 Proof of bias bound for the truncated KL estimator

In this proof, we establish that the bias of the truncated KL estimator is upper bounded by $\frac{C_2}{C_1^{1+1/d}} \big(\frac{k}{N}\big)^{\frac{1}{d}}$, where C_1 and C_2 are defined in Assumption 4.1. This formulation precisely quantifies the influence of the gradient of the density function on the bias of the estimator.

Proof. Note that $\sum_{j=1}^{d} \log \xi_{i,j}$ are identically distributed, then we have

$$\begin{aligned}
\mathbb{E}[\widehat{H}_{tKL}(X)] &= -\psi(k) + \psi(N) + \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\big[\sum_{j=1}^{d}\log \xi_{i,j}\big] \\
&= -\psi(k) + \psi(N) + \mathbb{E}\big[\sum_{j=1}^{d}\log \xi_k^{\mathrm{x}_j}(\mathrm{x})\big] \\
&= -\mathbb{E}[\log p_{\epsilon_k}(\mathrm{x})] + \mathbb{E}[\log \mu(B(\mathrm{x}; \xi_k^{\mathrm{x}_1}/2, ..., \xi_k^{\mathrm{x}_d}/2))] \\
&= -\mathbb{E}\big[\log \frac{P(B(\mathrm{x};\epsilon_k/2))}{\mu(B(\mathrm{x}; \xi_k^{\mathrm{x}_1}/2, ..., \xi_k^{\mathrm{x}_d}/2))}\big] \\
&= -\mathbb{E}\big[\log \frac{P(\overline{B}(\mathrm{x};\epsilon_k/2))}{\mu(\overline{B}(\mathrm{x};\epsilon_k/2))}\big],
\end{aligned} \tag{4.17}$$

where the third equality is from Lemma 4.1 and the fifth equality is due to the fact that p is supported on $\mathcal{Q}$. Note that

$$C_1 \leq \frac{P(\overline{B}(\mathrm{x};\epsilon_k/2))}{\mu(\overline{B}(\mathrm{x};\epsilon_k/2))} \leq \sup_{\mathrm{x}\in\mathcal{Q}} p(\mathrm{x}) < \infty, \tag{4.18}$$

and we have

$$\begin{aligned}
&\left|\log p(\mathrm{x}) - \log \frac{P(\overline{B}(\mathrm{x};\epsilon_k/2))}{\mu(\overline{B}(\mathrm{x};\epsilon_k/2))}\right| \\
\leq &\frac{1}{C_1}\left|p(\mathrm{x}) - \frac{P(\overline{B}(\mathrm{x};\epsilon_k/2))}{\mu(\overline{B}(\mathrm{x};\epsilon_k/2))}\right| \\
\leq &\frac{1}{C_1\mu(\overline{B}(\mathrm{x};\epsilon_k/2))}\int_{\overline{B}(\mathrm{x};\epsilon_k/2)} |p(\mathrm{x}) - p(\mathrm{x}')|d\mathrm{x}' \\
\leq &\frac{1}{C_1\mu(\overline{B}(\mathrm{x};\epsilon_k/2))}\int_{\overline{B}(\mathrm{x};\epsilon_k/2)} C_2||\mathrm{x} - \mathrm{x}'||_\infty d\mathrm{x}' \\
\leq &\frac{C_2}{2C_1}\epsilon_k.
\end{aligned} \tag{4.19}$$

Finally, using Lemma 4.4, the bias bound of $\mathbb{E}[\widehat{H}_{tKL}(X)]$ can be obtained by

$$\begin{aligned}
&\big|\mathbb{E}[\widehat{H}_{tKL}(X)] - H(X)\big| \\
\leq &\mathop{\mathbb{E}}_{\mathrm{x}\sim p}\mathbb{E}\Big[\left|\log p(\mathrm{x}) - \log \frac{P(\overline{B}(\mathrm{x};\epsilon_k/2))}{\mu(\overline{B}(\mathrm{x};\epsilon_k/2))}\right|\Big] \\
\leq &\frac{C_2}{2C_1}\mathop{\mathbb{E}}_{\mathrm{x}\sim p}\mathbb{E}[\epsilon_k] \\
\leq &\frac{C_2}{C_1^{1+1/d}}\big(\frac{k}{N}\big)^{\frac{1}{d}},
\end{aligned} \tag{4.20}$$

which completes the proof. □

4.1.4 Proof of variance bound for the truncated KL estimator

In this proof, we demonstrate that the variance of the truncated KL estimator is upper bounded by $C\frac{1}{N}$, where C being a positive constant. This finding suggests that the variance of the truncated KL estimator diminishes linearly relative to the sample size. This behavior is consistent with previously observed results for the non-truncated KL estimator, indicating a similar decay rate in variance as the sample size increases.

Proof. For the sake of convenience, we define $\alpha_i = \sum_{j=1}^{d} \log \xi_{i,j}$. We then define $\alpha_i', i = 1, ..., N$ as the estimators after $\mathrm{x}^{(1)}$ is resampled and $\alpha_i^*, i = 2, ..., N$ as the estimators after $\mathrm{x}^{(1)}$ is removed. Then, by the Efron-Stein inequality [44],

$$\begin{aligned}
\mathrm{Var}[\widehat{H}_{tKL}(X)] &= \mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N}\alpha_i\right] \\
&\leq \frac{N}{2}\mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N}\alpha_i - \frac{1}{N}\sum_{i=1}^{N}\alpha_i'\right)^2\right] \\
&\leq N\mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N}\alpha_i - \frac{1}{N}\sum_{i=2}^{N}\alpha_i^*\right)^2 + \left(\frac{1}{N}\sum_{i=1}^{N}\alpha_i' - \frac{1}{N}\sum_{i=2}^{N}\alpha_i^*\right)^2\right] \\
&= 2N\mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N}\alpha_i - \frac{1}{N}\sum_{i=2}^{N}\alpha_i^*\right)^2\right].
\end{aligned} \tag{4.21}$$

Let $\mathbb{1}_{E_i}$ be the indicator function of the event $E_i = \{\epsilon_k(\mathrm{x}^{(1)}) \neq \epsilon_k^*(\mathrm{x}^{(1)})\}$, where $\epsilon_k^*(\mathrm{x}^{(1)})$ is twice the k-NN distance of $\mathrm{x}^{(1)}$ when α_i^* are used. Then,

$$N\left(\frac{1}{N}\sum_{i=1}^{N}\alpha_i - \frac{1}{N}\sum_{i=2}^{N}\alpha_i^*\right) = \alpha_1 + \sum_{i=2}^{N}\mathbb{1}_{E_i}(\alpha_i - \alpha_i^*). \tag{4.22}$$

By Cauchy-Schwarz inequality, we have

$$\begin{aligned}
N^2\left(\frac{1}{N}\sum_{i=1}^{N}\alpha_i - \frac{1}{N}\sum_{i=2}^{N}\alpha_i^*\right)^2 &\leq \left(1 + \sum_{i=2}^{N}\mathbb{1}_{E_i}\right)\left(\alpha_1^2 + \sum_{i=2}^{N}\mathbb{1}_{E_i}(\alpha_i - \alpha_i^*)^2\right) \\
&\leq (1 + C_{k,d})\left(\alpha_1^2 + \sum_{i=2}^{N}\mathbb{1}_{E_i}(\alpha_i - \alpha_i^*)^2\right) \\
&\leq (1 + C_{k,d})\left(\alpha_1^2 + 2\sum_{i=2}^{N}\mathbb{1}_{E_i}(\alpha_i^2 + \alpha_i^{*2})\right),
\end{aligned} \tag{4.23}$$

where $C_{k,d}$ is the constant such that x^1 is amongst the k-nearest neighbors of at most $C_{k,d}$ other samples. Note that α_i and α_i^* are identically distributed, we only need to bound

$$\mathbb{E}[\alpha_1^2], \tag{4.24a}$$

$$(N-1)\mathbb{E}[\mathbb{1}_{E_2}\alpha_2^2], \tag{4.24b}$$

$$(N-1)\mathbb{E}[\mathbb{1}_{E_2}\alpha_2^{*2}]. \tag{4.24c}$$

Bound of (4.24a)**:**

We separate (4.24a) into two parts,

$$\mathbb{E}\big[\alpha_1^2\big] = \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}P:\epsilon_k<a_N}\big[\alpha_1^2\big] + \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}P:\epsilon_k\geq a_N}\big[\alpha_1^2\big], \tag{4.25}$$

where $a_N = \left(\frac{2k\log N}{C_1 N}\right)^{\frac{1}{d}}$.

First, we consider the bound of the first term in Eq (4.25). For any $\mathrm{x}\in\mathcal{Q}$,

$$\begin{aligned}&\mathop{\mathbb{E}}_{P:\epsilon_k<a_N}\big[\alpha_1^2\big]\\ =&\int_0^{a_N} f_{N,k}(r)\big[\log\left(\xi_k^{\mathrm{x}_1}\cdots\xi_k^{\mathrm{x}_d}\right)\big]^2 dr.\end{aligned} \tag{4.26}$$

where $f_{N,k}(r) = k\begin{pmatrix}N-1\\ k\end{pmatrix}\cdot\frac{dp_r}{dr}\cdot p_r^{k-1}\cdot(1-p_r)^{N-k-1}$ [85]. Note that for sufficiently large N,

$$\begin{aligned}&\int_0^{a_N}\left[\log\left(\xi_k^{\mathrm{x}_1}\cdots\xi_k^{\mathrm{x}_d}\right)\right]^2 dr\\ \leq&\int_0^{a_N}\big[\log\left(\frac{r}{2}\cdots\frac{r}{2}\right)\big]^2 dr\\ \leq& C_3\frac{(\log N)^3}{N^{1/d}},\end{aligned} \tag{4.27}$$

for some $C_3>0$, we now focus on bounding $f_{N,k}(r)$. By basic calculus, we can see that

$$k\begin{pmatrix}N-1\\ k\end{pmatrix}\cdot p_r^{k-1}\cdot(1-p_r)^{N-k-1}\leq C_4 N, \tag{4.28}$$

for some $C_4 > 0$ and $p_r \in (0,1)$. Also, by Lemma 4.6, we have $\frac{dp_r}{dr} \leq C_5 \frac{\log N}{N}$ for some $C_5 > 0$ and $r < a_N$. Therefore, the pdf term can be bounded by

$$f_{N,k}(r) \leq C_4 C_5 \log N. \tag{4.29}$$

Combining Eq (4.27) and Eq (4.29), we can bound Eq (4.26) by:

$$\mathop{\mathbb{E}}_{P:\epsilon_k < a_N} \left[\alpha_1^2\right] \leq C_3 C_4 C_5 \frac{(\log N)^4}{N^{1/d}} \leq C_6, \tag{4.30}$$

for some $C_6 > 0$. Thus, the first term in Eq (4.25) is bounded by

$$\mathop{\mathbb{E}}_{\mathrm{x} \in \mathcal{Q}P:\epsilon_k < a_N} \mathop{\mathbb{E}}_{P:\epsilon_k < a_N} \left[\alpha_1^2\right] \leq C_6. \tag{4.31}$$

Now we consider the second term in Eq (4.25). For $\epsilon_k \geq a_N$ and sufficiently large N, we have

$$\begin{aligned} \left[\log\left(\xi_k^{\mathrm{x}_1} \cdots \xi_k^{\mathrm{x}_d}\right)\right]^2 &\leq \left[\log\left(\epsilon_k/2 \cdots \epsilon_k/2\right)\right]^2 \\ &\leq d^2 \left[\log\left(\frac{a_N}{2}\right)\right]^2 \\ &\leq C_7 (\log N)^2, \end{aligned} \tag{4.32}$$

for some $C_7 > 0$. Using Lemma 4.3 and Eq (4.32), the second term in Eq (4.25) can be bounded by

$$\begin{aligned} \mathop{\mathbb{E}}_{\mathrm{x} \in \mathcal{Q}P:\epsilon_k \geq a_N} \mathop{\mathbb{E}}_{P:\epsilon_k \geq a_N} \left[\alpha_1^2\right] &= \mathop{\mathbb{E}}_{\mathrm{x} \in \mathcal{Q}P:\epsilon_k \geq a_N} \left[\left[\log\left(\xi_k^{\mathrm{x}_1} \cdots \xi_k^{\mathrm{x}_d}\right)\right]^2\right] \\ &\leq C_7 (\log N)^2 \cdot P(\epsilon_k \geq a_N) \\ &\leq C_8 \frac{(\log N)^{k+2}}{N^{2k}}, \end{aligned} \tag{4.33}$$

for some $C_8 > 0$.

Combining Eq (4.31) and Eq (4.33), the expectation of α_1^2 is bounded by

$$\mathbb{E}[\alpha_1^2] \leq C_9, \tag{4.34}$$

for some $C_9 > 0$.

Bound of (4.24b)**:**

Since the event E_2 is equivalent to the event that $\mathrm{x}^{(1)}$ is amongst the k-NN of $\mathrm{x}^{(2)}$, $\mathbb{E}[\mathbb{1}_{E_2}] = \mathbb{P}\{\mathrm{x}^{(1)} \in B(\mathrm{x}^{(2)}; \epsilon_k(\mathrm{x}^{(2)})\} = \frac{k}{N-1}$. Additionally, since E_2 is independent of $\epsilon_k(\mathrm{x}^{(2)})$, (4.24b) is therefore bounded as

$$(N-1)\mathbb{E}[\mathbb{1}_{E_2}\alpha_2^2] \leq (N-1)\mathbb{E}[\mathbb{1}_{E_2}]\mathbb{E}[\alpha_2^2] \leq kC_9, \tag{4.35}$$

where the second inequality is from Eq (4.34).

Bound of (4.24c)**:**

Using the independence between E_2 and $\epsilon_k^*(\mathrm{x}^{(2)})$ (twice the k-NN distance of $\mathrm{x}^{(2)}$ after $\mathrm{x}^{(1)}$ is removed), we can bound (4.24c) as

$$(N-1)\mathbb{E}[\mathbb{1}_{E_2}\alpha_2^{*2}] \leq (N-1)\mathbb{E}[\mathbb{1}_{E_2}]\mathbb{E}[\alpha_2^{*2}] \leq kC_{10}, \tag{4.36}$$

for some $C_{10} > 0$, where the second inequality is obtained from Eq (4.34) when the sample size is reduced to $N-1$.

Finally we obtain the bound of the variance of $\widehat{H}_{tKL}(X)$

$$\mathrm{Var}[\widehat{H}_{tKL}(X)] \leq C_{11}\frac{1}{N}, \tag{4.37}$$

for some $C_{11} > 0$. □

4.1.5 Proof of bias bound for the truncated KSG estimator

In this proof, the bias of the truncated KSG estimator is bounded by $C\frac{(\log N)^{k+2}}{C_1^{k+1}N^{\frac{1}{d}}}$, where C is a positive constant and C_1 is defined in Assumption 4.1. To the best of our knowledge, this result represents the first convergence analysis for estimators related to the KSG method.

Proof. We separate the d-dimensional unit cube $\mathcal{Q}$ into two subsets, $\mathcal{Q} = \mathcal{Q}_1 + \mathcal{Q}_2$, where $\mathcal{Q}_1 := [\frac{a_N}{2}, 1 - \frac{a_N}{2}]^d$, $a_N = \big(\frac{2k \log N}{C_1 N}\big)^{\frac{1}{d}}$, and $\mathcal{Q}_2 = \mathcal{Q} - \mathcal{Q}_1$. Suppose that $\widetilde{P}$, $\widetilde{p}$, and $\widetilde{q}_{\epsilon_k^{\mathrm{x}_1}, ..., \epsilon_k^{\mathrm{x}_d}}(\mathrm{x})$ are defined as in Lemma 4.2 with $l = p(\mathrm{x})^{-\frac{1}{d}}$, and by Lemma 4.2 and the fact that $\sum_{j=1}^d \log \zeta_{i,j}$ are identically distributed, we have

$$\begin{aligned}
\mathbb{E}[\widehat{H}_{tKSG}(X)] &= -\psi(k) + \psi(N) + (d-1)/k + \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}\big[\sum_{j=1}^{d} \log \zeta_{i,j}\big] \\
&= \mathop{\mathbb{E}}_{\mathrm{x}\sim p} \mathop{\mathbb{E}}_{P} \big[\log \zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d}\big] - \mathop{\mathbb{E}}_{\mathrm{x}\sim p} \mathop{\mathbb{E}}_{\widetilde{P}} \big[\log \widetilde{q}_{\epsilon_k^{\mathrm{x}_1}, ..., \epsilon_k^{\mathrm{x}_d}}\big] \\
&= \mathop{\mathbb{E}}_{\mathrm{x}\sim p} \mathop{\mathbb{E}}_{P} \big[\log \zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d}\big] - \mathop{\mathbb{E}}_{\mathrm{x}\sim p} \mathop{\mathbb{E}}_{\widetilde{P}} \big[\log \big(p(\mathrm{x}) \epsilon_k^{\mathrm{x}_1} \cdots \epsilon_k^{\mathrm{x}_d}\big)\big].
\end{aligned} \tag{4.38}$$

We decompose the bias into three terms and bound them separately:

$$\begin{aligned}
&\big|\mathbb{E}[\widehat{H}_{tKSG}(X)] - H(X)\big| \\
=&\left| \mathop{\mathbb{E}}_{\mathrm{x}\sim p} \mathop{\mathbb{E}}_{P} \big[\log \big(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d}\big)\big] - \mathop{\mathbb{E}}_{\mathrm{x}\sim p} \mathop{\mathbb{E}}_{\widetilde{P}} \big[\log \big(\epsilon_k^{\mathrm{x}_1} \cdots \epsilon_k^{\mathrm{x}_d}\big)\big] \right| \\
\leq& I_1 + I_2 + I_3,
\end{aligned} \tag{4.39}$$

with

$$\begin{aligned}
I_1 &= \left| \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_2} \mathop{\mathbb{E}}_{P:\epsilon_k < a_N} \big[\log \big(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d}\big)\big] \right| + \left| \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_2} \mathop{\mathbb{E}}_{\widetilde{P}:\epsilon_k < a_N} \big[\log \big(\epsilon_k^{\mathrm{x}_1} \cdots \epsilon_k^{\mathrm{x}_d}\big)\big] \right|, \\
I_2 &= \left| \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_1} \mathop{\mathbb{E}}_{P:\epsilon_k < a_N} \big[\log \big(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d}\big)\big] - \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_1} \mathop{\mathbb{E}}_{\widetilde{P}:\epsilon_k < a_N} \big[\log \big(\epsilon_k^{\mathrm{x}_1} \cdots \epsilon_k^{\mathrm{x}_d}\big)\big] \right|, \\
I_3 &= \left| \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}} \mathop{\mathbb{E}}_{P:\epsilon_k \geq a_N} \big[\log \big(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d}\big)\big] \right| + \left| \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}} \mathop{\mathbb{E}}_{\widetilde{P}:\epsilon_k \geq a_N} \big[\log \big(\epsilon_k^{\mathrm{x}_1} \cdots \epsilon_k^{\mathrm{x}_d}\big)\big] \right|,
\end{aligned} \tag{4.40}$$

where $\mathop{\mathbb{E}}_{P:\epsilon_k < a_N}$ means taking expectation under the probability measure P over $\epsilon_k^{\mathrm{x}_j} < a_N, j = 1, ..., d$.

Bound of I_1:

For any $\mathrm{x} \in \mathcal{Q}_2$,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{P:\epsilon_k < a_N} \big[\log \big(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d}\big)\big] \\
=& \int_0^{a_N} \cdots \int_0^{a_N} f_{N,k}(r_1, ..., r_d) \log \big(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d}\big) dr_1 \cdots dr_d.
\end{aligned} \tag{4.41}$$

where $f_{N,k}(r_1,...,r_d) = \begin{pmatrix} N-1 \\ k \end{pmatrix} \cdot \frac{\partial^d(q^k_{r_1,...,r_d})}{\partial r_1 \cdots \partial r_d} \cdot (1-p_{r_{\mathrm{m}}})^{N-k-1}$, and $r_m = \max\limits_{1\le j\le d} r_j$ [85]. Note that for sufficiently large N, we have,

$$\begin{aligned}
&\int_0^{a_N} \cdots \int_0^{a_N} \big|\log\big(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d}\big)\big| dr_1 \cdots dr_d \\
\le &\int_0^{a_N} \cdots \int_0^{a_N} \big|\log\big(\frac{r_1}{2} \cdots \frac{r_d}{2}\big)\big| dr_1 \cdots dr_d \\
\le &\int_0^{a_N} \cdots \int_0^{a_N} \big|\log\big(r_1 \cdots r_d\big)\big| dr_1 \cdots dr_d + \int_0^{a_N} \cdots \int_0^{a_N} d\log 2 dr_1 \cdots dr_d \\
= &- d(a_N)^{d-1} \int_0^{a_N} \log r dr + d\log 2 \bigg(\int_0^{a_N} dr\bigg)^d \\
\le &C_3 \frac{\big(\log N\big)^2}{C_1 N},
\end{aligned} \tag{4.42}$$

for some $C_3 > 0$. We now focus on bounding $f_{N,k}(r_1,...,r_d)$. We omit the subscripts of $q_{r_1,...,r_d}$ for simplicity from now. By the multivariate version of Faà di Bruno's formula [69], one obtains

$$\frac{\partial^d(q^k)}{\partial r_1 \cdots \partial r_d} = \sum_{\pi\in\Pi} \frac{d^{|\pi|}q^k}{(dq)^{|\pi|}} \cdot \prod_{B\in\pi} \frac{\partial^{|B|}q}{\prod_{j\in B}\partial r_j}, \tag{4.43}$$

where π runs through the set Π of all partitions of the set $1,...,d$. By Lemma 4.5, we have

$$\frac{\partial^{|B|}q}{\prod_{j\in B}\partial r_j} \le p(\mathrm{x}) r_{\mathrm{m}}^{d-|B|} + C_2 r_{\mathrm{m}}^{d-|B|+1}, \tag{4.44}$$

which implies that

$$\prod_{B\in\pi} \frac{\partial^{|B|}q}{\prod_{j\in B}\partial r_j} \le M r_{\mathrm{m}}^{(|\pi|-1)d}, \tag{4.45}$$

where $M = p^{*d} + 1$ and $p^* = \sup_{\mathrm{x}\in\mathcal{Q}} p(\mathrm{x})$. Therefore, for $|\pi| \le k$ and $r_{\mathrm{m}} \le a_N$ we can bound $f_{N,k}(r_1, ..., r_d)$ as

$$\begin{aligned} f_{N,k}(r_1, ..., r_d) =& \sum_{\pi\in\Pi} \binom{N-1}{k} \cdot \frac{d^{|\pi|}q^k}{(dq)^{|\pi|}} \cdot \prod_{B\in\pi} \frac{\partial^{|B|}q}{\prod_{j\in B}\partial r_j} \cdot (1-p_{r_{\mathrm{m}}})^{N-k-1} \\ \le& \sum_{\pi\in\Pi} \frac{(N-1)!}{(k-|\pi|)!(N-k-1)!} q^{k-|\pi|}(1-p_{r_{\mathrm{m}}})^{N-k-1} M r_{\mathrm{m}}^{(|\pi|-1)d} \\ \le& \sum_{\pi\in\Pi} M\cdot N^k p_{r_{\mathrm{m}}}^{k-|\pi|}(1-p_{r_{\mathrm{m}}})^{N-k-1} r_{\mathrm{m}}^{(|\pi|-1)d} \\ \le& \sum_{\pi\in\Pi} CM\cdot N^{|\pi|} r_{\mathrm{m}}^{(|\pi|-1)d} \\ \le& \sum_{\pi\in\Pi} CM\left(\frac{2k\log N}{C_1}\right)^{|\pi|-1} N \\ \le& |\Pi| CM\left(\frac{2k\log N}{C_1}\right)^{k-1} N, \end{aligned} \tag{4.46}$$

where the third inequality is due to the fact that $p^{k-|\pi|}(1-p)^{N-k-1} \le CN^{-k+|\pi|}$ for $p \in [0,1]$. Combining Eq (4.46) and Eq (4.42), we can bound the expectation in Eq (4.41) by

$$\left| \mathop{\mathbb{E}}_{P:\epsilon_k<a_N} \left[\log\left(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\right)\right] \right| \le C_4 \frac{\left(\log N\right)^{k+1}}{C_1^k} \tag{4.47}$$

for some $C_4 > 0$. It follows that the first term of I_1 is bounded by

$$\begin{aligned} \left| \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_2} \mathop{\mathbb{E}}_{P:\epsilon_k<a_N} \left[\log\left(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\right)\right] \right| &\le C_4 \frac{\left(\log N\right)^{k+1}}{C_1^k} \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_2}[1] \\ &\le C_4 \frac{\left(\log N\right)^{k+1}}{C_1^k} p^* \mu(x\in\mathcal{Q}_2) \\ &\le p^* C_4 \frac{\left(\log N\right)^{k+1}}{C_1^k} (d+1) a_N \\ &= (d+1)p^* C_4 \frac{\left(\log N\right)^{k+1}}{C_1^k} \left(\frac{2k\log N}{C_1 N}\right)^{\frac{1}{d}}. \end{aligned} \tag{4.48}$$

Since $\widetilde{P}$ is a sepcial case of P, the second term of I_1 can also be bounded by the same order. Thus, I_1 is bounded by

$$|I_1| \le C_5 \frac{\left(\log N\right)^{k+2}}{C_1^{k+1} N^{\frac{1}{d}}}, \tag{4.49}$$

for some $C_5 > 0$.

Bound of I_2:

For any $\mathrm{x} \in \mathcal{Q}_1$ and $\epsilon_k^{\mathrm{x}_j} < a_N, j = 1, ..., d$, it is easy to see that $\zeta_k^{\mathrm{x}_j} = \epsilon_k^{\mathrm{x}_j}$. Thus, I_2 can be bounded and rewritten as

$$\begin{aligned} I_2 &\leq \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_1} \left| \mathop{\mathbb{E}}_{P:\epsilon_k<a_N} \left[\log\left(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d}\right)\right] - \mathop{\mathbb{E}}_{\widetilde{P}:\epsilon_k<a_N} \left[\log\left(\epsilon_k^{\mathrm{x}_1} \cdots \epsilon_k^{\mathrm{x}_d}\right)\right] \right| \\ &= \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_1} \left| \int_0^{a_N} \cdots \int_0^{a_N} \left(f_{N,k}(r_1, ..., r_d) - \widetilde{f}_{N,k}(r_1, ..., r_d)\right) \log\left(r_1 \cdots r_d\right) dr_1 \cdots dr_d \right|, \end{aligned} \tag{4.50}$$

where $\widetilde{f}_{N,k}(r_1, ..., r_d) = \begin{pmatrix} N-1 \\ k \end{pmatrix} \frac{\partial^d(\widetilde{q}^k_{r_1,...,r_d})}{\partial r_1 \cdots \partial r_d} \cdot (1 - \widetilde{p}_{r_{\mathrm{m}}})^{N-k-1}$. Again, we omit the subscripts of $\widetilde{q}_{r_1,...,r_d}$ in the following analysis. Since we have

$$\begin{aligned} &\int_0^{a_N} \cdots \int_0^{a_N} \left| \log\left(r_1 \cdots r_d\right)\right| dr_1 \cdots dr_d \\ \leq & C_3 \frac{\left(\log N\right)^2}{C_1 N}, \end{aligned} \tag{4.51}$$

from (4.42), we now focus on bounding $f_{N,k}(r_1,...,r_d) - \widetilde{f}_{N,k}(r_1,...,r_d)$. Recall the Faà di Bruno's formula in Eq (4.43), and we have

$$
\begin{aligned}
&f_{N,k}(r_1,...,r_d)\\
=&\sum_{\pi\in\Pi}\binom{N-1}{k}\frac{\partial^{|\pi|}q^k}{(\partial q)^{|\pi|}}\prod_{B\in\pi}\frac{\partial^{|B|}q}{\prod_{j\in B}\partial r_j}(1-p_{r_{\mathrm{m}}})^{N-k-1}\\
=&\sum_{\pi\in\Pi}\binom{N-1}{k}\frac{k!}{(k-|\pi|)!}\big(p(\mathrm{x})r_1\cdots r_d+O(r_1\cdots r_d r_{\mathrm{m}})\big)^{k-|\pi|}\\
&\qquad\times\prod_{B\in\pi}\big(p(\mathrm{x})\prod_{j\in\widehat{B}}r_j+O(r_{\mathrm{m}}\prod_{j\in\widehat{B}}r_j)\big)\big(1-p(\mathrm{x})r_{\mathrm{m}}^d-O(r_{\mathrm{m}}^{d+1})\big)^{N-k-1}\\
=&\sum_{\pi\in\Pi}\binom{N-1}{k}\frac{k!}{(k-|\pi|)!}\big(p(\mathrm{x})r_1\cdots r_d\big)^{k-|\pi|}\big(1+O(r_{\mathrm{m}})\big)^{k-|\pi|}\prod_{B\in\pi}\big(p(\mathrm{x})\prod_{j\in\widehat{B}}r_j\big)\\
&\qquad\times\big(1+O(r_{\mathrm{m}})\big)\big(1-p(\mathrm{x})r_{\mathrm{m}}^d\big)^{N-k-1}\big(1-O(r_{\mathrm{m}}^{d+1})\big)^{N-k-1}\\
=&\sum_{\pi\in\Pi}\binom{N-1}{k}\frac{k!}{(k-|\pi|)!}\big(p(\mathrm{x})r_1\cdots r_d\big)^{k-|\pi|}\cdot\prod_{B\in\pi}\big(p(\mathrm{x})\prod_{j\in\widehat{B}}r_j\big)\\
&\qquad\times\big(1-p(\mathrm{x})r_{\mathrm{m}}^d\big)^{N-k-1}\cdot\big(1+O(r_{\mathrm{m}})\big)^{k}\big(1-O(r_{\mathrm{m}}^{d+1})\big)^{N-k-1}\\
=&\sum_{\pi\in\Pi}\binom{N-1}{k}\frac{\partial^{|\pi|}\widetilde{q}^k}{(\partial\widetilde{q})^{|\pi|}}\cdot\prod_{B\in\pi}\frac{\partial^{|B|}\widetilde{q}}{\prod_{j\in B}\partial r_j}\cdot(1-\widetilde{p}_{r_{\mathrm{m}}})^{N-k-1}\cdot\big(1+O(r_{\mathrm{m}})\big)^{k}\big(1-O(r_{\mathrm{m}}^{d+1})\big)^{N-k-1}\\
=&\widetilde{f}_{N,k}(r_1,...,r_d)\cdot\big(1+O(r_{\mathrm{m}})\big)^{k}\big(1-O(r_{\mathrm{m}}^{d+1})\big)^{N-k-1}
\end{aligned}
\tag{4.52}
$$

where the second equality is from Lemma 4.5 and Lemma 4.6 and the fifth equality is from the fact that $\widetilde{q}=p(\mathrm{x})r_1\cdots r_d$ and $\widetilde{p}_{r_{\mathrm{m}}}=p(\mathrm{x})r_{\mathrm{m}}^d$ for $\mathrm{x}\in\mathcal{Q}_1$ and $r_{\mathrm{m}}\le a_N$.

By Eq (4.52), we obtain the bound of the difference $f_{N,k}(r_1,...,r_d) - \widetilde{f}_{N,k}(r_1,...,r_d)$

$$\begin{aligned}
&|f_{N,k}(r_1,...,r_d) - \widetilde{f}_{N,k}(r_1,...,r_d)| \\
=&\left|\big(1+O(r_{\mathrm{m}})\big)^k\big(1-O(r_{\mathrm{m}}^{d+1})\big)^{N-k-1} - 1\right|\widetilde{f}_{N,k}(r_1,...,r_d) \\
\leq& C_6 r_{\mathrm{m}} \widetilde{f}_{N,k}(r_1,...,r_d) \\
\leq& C_6\left(\frac{2k\log N}{C_1 N}\right)^{\frac{1}{d}}|\Pi|CM\left(\frac{2k\log N}{C_1}\right)^{k-1}N,
\end{aligned} \tag{4.53}$$

for some $C_6 > 0$, where the last inequality is from Eq (4.46) and the fact that $\widetilde{P}$ is a special case of P. Combining Eq (4.53) and Eq (4.51), we obtain the bound of I_2

$$\begin{aligned}
I_2 &\leq C_3 C_6\left(\frac{2k\log N}{C_1 N}\right)^{\frac{1}{d}}|\Pi|CM\left(\frac{2k\log N}{C_1}\right)^{k-1}\frac{\big(\log N\big)^2}{C_1}\mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_1}[1] \\
&\leq C_7\frac{(\log N)^{k+2}}{C_1^{k+1}N^{\frac{1}{d}}},
\end{aligned} \tag{4.54}$$

for some $C_7 > 0$, as $\mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}_1}[1] \leq 1$.

Bound of I_3:

To bound the first term of I_3, we need to bound $\mathop{\mathbb{E}}_{P:\epsilon_k\geq a_N}\big[\big|\log\big(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\big)\big|\big]$ first. Note that the event $\{\epsilon_k \geq a_N\}$ is equivalent to that there is at least one $j \in \{1,...,d\}$ such that $\epsilon_k^{\mathrm{x}_j} \geq a_N$, and by the symmetry of the equation, the expectation over this set can be rewritten as

$$\mathop{\mathbb{E}}_{P:\epsilon_k\geq a_N}\big[\big|\log\big(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\big)\big|\big] = \sum_{i=1}^{d} C_d^i \mathop{\mathbb{E}}_{P:\begin{cases}\epsilon_{k,1:i}\geq a_N\\ \epsilon_{k,i:d} < a_N\end{cases}}\big[\big|\log\big(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\big)\big|\big]. \tag{4.55}$$

Consider each term in Eq (4.55)

$$\begin{aligned}
&\mathop{\mathbb{E}}_{P:\begin{cases}\epsilon_{k,1:i}\geq a_N\\ \epsilon_{k,i:d}<a_N\end{cases}}\left[\left|\log\left(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\right)\right|\right]\\
\leq&\mathop{\mathbb{E}}_{P:\begin{cases}\epsilon_{k,1:i}\geq a_N\\ \epsilon_{k,i:d}<a_N\end{cases}}\left[\left|\log\left(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_i}\right)\right|\right]+\mathop{\mathbb{E}}_{P:\begin{cases}\epsilon_{k,1:i}\geq a_N\\ \epsilon_{k,i:d}<a_N\end{cases}}\left[\left|\log\left(\zeta_k^{\mathrm{x}_{i+1}}\cdots\zeta_k^{\mathrm{x}_d}\right)\right|\right].
\end{aligned}\tag{4.56}$$

For $\epsilon_k^{\mathrm{x}_j}\geq a_N, j=1,...,i$ and sufficiently large N, we have

$$\begin{aligned}
\left|\log\left(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_i}\right)\right|&\leq\left|\log\left(\epsilon_k^{\mathrm{x}_1}/2\cdots\epsilon_k^{\mathrm{x}_i}/2\right)\right|\\
&\leq\left|\log\left(\frac{a_N}{2}\right)^i\right|\\
&\leq C_8\log N,
\end{aligned}\tag{4.57}$$

for some $C_8>0$. Using Lemma 4.3 and Eq (4.57), the first term of Eq (4.56) can be bounded by

$$\begin{aligned}
\mathop{\mathbb{E}}_{P:\begin{cases}\epsilon_{k,1:i}\geq a_N\\ \epsilon_{k,i:d}<a_N\end{cases}}\left[\left|\log\left(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_i}\right)\right|\right]&\leq C_8\log N\cdot\mathbb{P}\{\epsilon_{k,1:i}\geq a_N,\epsilon_{k,i:d}<a_N\}\\
&\leq C_8\log N\cdot P\{\epsilon_k\geq a_N\}\\
&\leq C_9\frac{(\log N)^{k+1}}{N^{2k}},
\end{aligned}\tag{4.58}$$

For some $C_9>0$.

Now consider the second term of Eq (4.56). Like Eq (4.42), the integration with respect to Lebesgue measure can be bounded as

$$\begin{aligned}
&\int_{a_N}^1\cdots\int_{a_N}^1\left(\int_0^{a_N}\cdots\int_0^{a_N}\left|\log\left(\zeta_k^{\mathrm{x}_{i+1}}\cdots\zeta_k^{\mathrm{x}_d}\right)\right|dr_{i+1}\cdots dr_d\right)dr_d\cdots dr_i\\
\leq&-(d-i)(a_N)^{d-i-1}\int_0^{a_N}\log r dr+(d-i)\log 2(\int_0^{a_N}dr)^{d-i}\\
\leq&C_{10}\log N,
\end{aligned}\tag{4.59}$$

for some $C_{10} > 0$. Again using the multivariate version of Faà di Bruno's formula, we can bound $f_{N,k}(r_1, ..., r_d)$ for $|\pi| \leq k$ and $r_{\mathrm{m}} \geq a_N$ as

$$\begin{aligned} f_{N,k}(r_1, ..., r_d) =& \sum_{\pi \in \Pi} \binom{N-1}{k} \cdot \frac{d^{|\pi|} q^k}{(dq)^{|\pi|}} \cdot \prod_{B \in \pi} \frac{\partial^{|B|} q}{\prod_{j \in B} \partial r_j} \cdot (1 - p_{r_{\mathrm{m}}})^{N-k-1} \\ \leq& \sum_{\pi \in \Pi} \frac{(N-1)!}{(k-|\pi|)!(N-k-1)!} q^{k-|\pi|} (1 - p_{r_{\mathrm{m}}})^{N-k-1} M r_{\mathrm{m}}^{(|\pi|-1)d} \\ \leq& \sum_{\pi \in \Pi} \frac{(N-1)!}{(k-|\pi|)!(N-k-1)!} (1 - C_1 a_N^d)^{N-k-1} M \\ \leq& C_{11} \frac{1}{N^k}, \end{aligned} \tag{4.60}$$

for some $C_{11} > 0$. Therefore, combining Eq (4.59) and Eq (4.60) leads to the bound of the second term of Eq (4.56)

$$\mathop{\mathbb{E}}_{P: \begin{cases} \epsilon_{k,1:i} \geq a_N \\ \epsilon_{k,i:d} < a_N \end{cases}} \left[\left| \log \left(\zeta_k^{\mathrm{x}_{i+1}} \cdots \zeta_k^{\mathrm{x}_d} \right) \right| \right] \leq C_{10} C_{11} \frac{\log N}{N^k}, \tag{4.61}$$

which is a larger bound then Eq (4.58). As a result we can bound Eq (4.56) by

$$\mathop{\mathbb{E}}_{P: \begin{cases} \epsilon_{k,1:i} \geq a_N \\ \epsilon_{k,i:d} < a_N \end{cases}} \left[\left| \log \left(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d} \right) \right| \right] \leq C_{10} C_{11} \frac{\log N}{N^k}. \tag{4.62}$$

Given Eq (4.62), we are now able to estimate Eq (4.55) and then the first term of I_3 by the same bound up to a constant. Similarly, we can also bound the second term of I_3 by $O\big(\frac{\log N}{N^k}\big)$. Thus, I_3 can be bounded by

$$I_3 \leq C_{12} \frac{\log N}{N^k}, \tag{4.63}$$

for some $C_{12} > 0$.

Finally, combining the upper bounds of I_1, I_2 and I_3, we obtain that the bias is bounded by

$$\left| \mathbb{E}[\widehat{H}_{tKSG}(X)] - H(X) \right| \leq C_{13} \frac{(\log N)^{k+2}}{C_1^{k+1} N^{\frac{1}{d}}}, \tag{4.64}$$

for some $C_{13} > 0$. □

4.1.6 Proof of variance bound for the truncated KSG estimator

In this proof, we determine that the variance bound of the truncated KSG estimator can be represented as $C\frac{(\log N)^{k+2}}{N}$, with C being a positive constant. This finding again indicates a linear decay rate for the variance, which is consistent with the behavior observed in other k-NN based estimators.

Proof. We let $\beta_i = \sum_{j=1}^{d} \log \zeta_{i,j}$, and define $\beta_i', i = 1, ..., N$ as the estimators after $\mathrm{x}^{(1)}$ is resampled and $\beta_i^*, i = 2, ..., N$ as the estimators after $\mathrm{x}^{(1)}$ is removed. It should be noted that this proof can be completed by following the roadmap in 4.1.4, and the only issue that needs to be validated here is that $\mathbb{E}[\beta_1^2] = O((\log N)^{k+2})$.

Again, we separate $\mathbb{E}\big[\beta_1^2\big]$ into two parts,

$$\mathbb{E}\big[\beta_1^2\big] = \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}P:\epsilon_k<a_N} \big[\beta_1^2\big] + \mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}P:\epsilon_k\geq a_N} \big[\beta_1^2\big], \tag{4.65}$$

where a_N is defined as in 4.1.5.

The bound of the first term in Eq (4.65)

First, we consider the bound of the first term in Eq (4.65). For any $\mathrm{x} \in \mathcal{Q}$,

$$\begin{aligned} &\mathop{\mathbb{E}}_{P:\epsilon_k<a_N} \big[\beta_1^2\big] \\ &= \int_0^{a_N} \cdots \int_0^{a_N} f_{N,k}(r_1, ..., r_d) \big[\log\big(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d}\big)\big]^2 dr_1 \cdots dr_d, \end{aligned} \tag{4.66}$$

where $f_{N,k}(r_1, ..., r_d) = \binom{N-1}{k} \cdot \frac{\partial^d (q^k_{r_1,...,r_d})}{\partial r_1 \cdots \partial r_d} \cdot (1 - p_{r_{\mathrm{m}}})^{N-k-1}$, and $r_{\mathrm{m}} = \max\limits_{1\leq j\leq d} r_j$ [85].

Note that for sufficiently large N, we have,

$$\begin{aligned}
&\int_0^{a_N}\cdots\int_0^{a_N}\left[\log\left(\zeta_k^{\mathrm{x}_1}\cdots\zeta_k^{\mathrm{x}_d}\right)\right]^2 dr_1\cdots dr_d\\
\leq&\int_0^{a_N}\cdots\int_0^{a_N}\left[\log\left(\frac{r_1}{2}\cdots\frac{r_d}{2}\right)\right]^2 dr_1\cdots dr_d\\
=&d\int_0^{a_N}\cdots\int_0^{a_N}\left[\log\left(\frac{r_1}{2}\right)\right]^2 dr_1\cdots dr_d+d(d-1)\int_0^{a_N}\cdots\int_0^{a_N}\log\left(\frac{r_1}{2}\right)\log\left(\frac{r_2}{2}\right)dr_1\cdots dr_d\\
\leq&C_3\frac{(\log N)^3}{N},
\end{aligned}\tag{4.67}$$

for some $C_3>0$. Recall Eq (4.46), and we can bound Eq (4.66) as:

$$\mathop{\mathbb{E}}_{P:\epsilon_k<a_N}\left[\beta_1^2\right]\leq C_4(\log N)^{k+2},\tag{4.68}$$

for some $C_4>0$. Thus, the first term in Eq (4.65) is bounded by

$$\mathop{\mathbb{E}}_{\mathrm{x}\in\mathcal{Q}}\mathop{\mathbb{E}}_{P:\epsilon_k<a_N}\left[\beta_1^2\right]\leq C_4(\log N)^{k+2}.\tag{4.69}$$

The second term in Eq (4.65)

Now we consider the second term in Eq (4.65).

Like the bound analysis of I_3 in 4.1.5, we can rewrite $\mathop{\mathbb{E}}_{P:\epsilon_k\geq a_N}\left[\beta_1^2\right]$ as

$$\mathop{\mathbb{E}}_{P:\epsilon_k\geq a_N}\left[\beta_1^2\right]=\sum_{i=1}^{d}C_d^i\mathop{\mathbb{E}}_{P:\begin{cases}\epsilon_{k,1:i}\geq a_N\\ \epsilon_{k,i:d}<a_N\end{cases}}\left[\beta_1^2\right].\tag{4.70}$$

Consider each term of Eq (4.55)

$$\begin{aligned}
&\mathop{\mathbb{E}}_{P:\begin{cases}\epsilon_{k,1:i} \geq a_N \\ \epsilon_{k,i:d} < a_N\end{cases}} \left[\beta_1^2\right] \\
\leq & 2\left(\mathop{\mathbb{E}}_{P:\begin{cases}\epsilon_{k,1:i} \geq a_N \\ \epsilon_{k,i:d} < a_N\end{cases}} \left[\left| \log\left(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_i}\right)\right|^2\right] + \mathop{\mathbb{E}}_{P:\begin{cases}\epsilon_{k,1:i} \geq a_N \\ \epsilon_{k,i:d} < a_N\end{cases}} \left[\left| \log\left(\zeta_k^{\mathrm{x}_{i+1}} \cdots \zeta_k^{\mathrm{x}_d}\right)\right|^2\right] \right)
\end{aligned} \tag{4.71}$$

For $\epsilon_k^{\mathrm{x}_j} \geq a_N, j = 1, ..., i$ and sufficiently large N, we have

$$\begin{aligned}
\left| \log\left(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_i}\right)\right|^2 &\leq \left| \log\left(\epsilon_k^{\mathrm{x}_1}/2 \cdots \epsilon_k^{\mathrm{x}_i}/2\right)\right|^2 \\
&\leq \left| \log\left(\frac{a_N}{2}\right)^i\right|^2 \\
&\leq C_5 (\log N)^2,
\end{aligned} \tag{4.72}$$

for some $C_5 > 0$. Using Lemma 4.3 and Eq (4.72), the first term of Eq (4.71) can be bounded by

$$\begin{aligned}
\mathop{\mathbb{E}}_{P:\begin{cases}\epsilon_{k,1:i} \geq a_N \\ \epsilon_{k,i:d} < a_N\end{cases}} \left[\left| \log\left(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_i}\right)\right|^2\right] &\leq C_5 (\log N)^2 \cdot \mathbb{P}\{\epsilon_{k,1:i} \geq a_N, \epsilon_{k,i:d} < a_N\} \\
&\leq C_5 (\log N)^2 \cdot P\{\epsilon_k \geq a_N\} \\
&\leq C_6,
\end{aligned} \tag{4.73}$$

for some $C_6 > 0$.

Now consider the second term of Eq (4.71). Like Eq (4.67), the integration with respect to Lebesgue measure is bounded as

$$\begin{aligned}
&\int_{a_N}^1 \cdots \int_{a_N}^1 \left(\int_0^{a_N} \cdots \int_0^{a_N} \left| \log\left(\zeta_k^{\mathrm{x}_{i+1}} \cdots \zeta_k^{\mathrm{x}_d}\right)\right|^2 dr_{i+1} \cdots dr_d \right) dr_d \cdots dr_i \\
\leq & C_7,
\end{aligned} \tag{4.74}$$

for some $C_7 > 0$. Therefore, combining Eq (4.74) and the PDF bound in Eq (4.60) leads to the bound of the second term of Eq (4.71)

$$\mathbb{E}_{P:\begin{cases}\epsilon_{k,1:i} \geq a_N \\ \epsilon_{k,i:d} < a_N\end{cases}} \left[\left| \log \left(\zeta_k^{\mathrm{x}_{i+1}} \cdots \zeta_k^{\mathrm{x}_d} \right) \right|^2 \right] \leq C_8, \tag{4.75}$$

for some $C_8 > 0$. As a result we can bound Eq (4.71) by

$$\mathbb{E}_{P:\begin{cases}\epsilon_{k,1:i} \geq a_N \\ \epsilon_{k,i:d} < a_N\end{cases}} \left[\left| \log \left(\zeta_k^{\mathrm{x}_1} \cdots \zeta_k^{\mathrm{x}_d} \right) \right| \right] \leq C_6 + C_8. \tag{4.76}$$

Given Eq (4.76), we are now able to estimate Eq (4.70) and then the second term of Eq (4.65) by the same bound up to a constant.

Finally, the expectation of β_1^2 is bounded as

$$\mathbb{E}[\beta_1^2] \leq C_9 (\log N)^{k+2}, \tag{4.77}$$

for some $C_9 > 0$. Following the same procedure in 4.1.4, we can obtain the bound of the variance of $\widehat{H}_{tKSG}(X)$

$$\mathrm{Var}[\widehat{H}_{tKSG}(X)] \leq C_{10} \frac{(\log N)^{k+2}}{N}, \tag{4.78}$$

for some $C_{10} > 0$. □

4.2 Convergence Analyses for the UM based estimators

Our theoretical analyses are based on the following assumptions.

Assumption 4.2. *Let $S = \{\mathrm{x}^{(i)}\}_{i=1}^N$ be the set of i.i.d samples used to construct the UM and p_{z}^S be the resulting density of Z in Eq. (3.26). Denote $C_2^N = \sup_{\mathrm{z} \in \mathcal{Q}^o} ||\nabla p_{\mathrm{z}}^S(\mathrm{z})||_1$, and assume*

that C_2^N *satisfies: (1)* $C_2^N \xrightarrow[N\to\infty]{\mathbb{P}} 0$*; (2) There exist a positive integer* M *and a positive real number* $\bar{C} < 1$ *such that:*

$$\forall N > M, \quad C_2^N \leq \bar{C}, \, a.s.$$

4.2.1 Proof of bias and MSE bounds for the UM-tKL estimator

In this proof, we establish a bias bound for the UM-tKL estimator as $C^N_{UM-tKL}\big(\frac{k}{N}\big)^{\frac{1}{d}}$ and an MSE (Mean Squared Error) bound of $C\frac{1}{N} + D^N_{UM-tKL}\big(\frac{k}{N}\big)^{\frac{2}{d}}$. Here, C denotes a positive constant, and both C^N_{UM-tKL} and D^N_{UM-tKL} converge to zero as N approaches infinity. The convergence behavior of C^N_{UM-tKL} provides insight into how the UM enhances the convergence rate of the KL estimator, offering a more efficient approach for entropy estimation.

Proof. Given a UM f, the density of the original distribution satisfies the change of variable formula,

$$p_{\mathrm{x}}(\mathrm{x}) = p_{\mathrm{z}}(f(\mathrm{x}))g(\mathrm{x}), \tag{4.79}$$

where $g(\mathrm{x}) = \left|\det \frac{\partial f(\mathrm{x})}{\partial \mathrm{x}}\right|$ is differentiable and positive for any $\mathrm{x} \in \mathbb{R}^d$ ([120, 40]). Recall that p_{x} is differentiable, and it follows that,

$$p_{\mathrm{z}}(\mathrm{z}) = \frac{p_{\mathrm{x}}(f^{-1}(\mathrm{z}))}{g(f^{-1}(\mathrm{z}))}, \tag{4.80}$$

is also differentiable for any $\mathrm{z} \in Q^o$. Thus, the supreme C_2^N is a well defined random variable.

Since p_{z}^S is a differentiable density function defined on $\mathcal{Q}$, there exists a $\mathrm{z}^* \in \mathcal{Q}$ such that $p_{\mathrm{z}}^S(\mathrm{z}^*) = 1$. By mean value theorem, we have

$$\begin{aligned} &|1 - p_{\mathrm{z}}^S(\mathrm{z})| \\ \leq &|\nabla p_{\mathrm{z}}^S(\xi) \cdot (\mathrm{z}^* - \mathrm{z})| \\ \leq &||\nabla p_{\mathrm{z}}^S(\xi)||_1 \cdot ||\mathrm{z}^* - \mathrm{z}||_\infty \\ \leq &C_2^N, \end{aligned} \tag{4.81}$$

where ξ is some vector in $\mathcal{Q}$. Thus, we have

$$1 - C_2^N \leq p_{\mathrm{x}}^N(\mathrm{x}) \leq 1 + C_2^N. \tag{4.82}$$

Now define $C_1^N = \inf_{\mathrm{z}\in\mathcal{Q}} p_{\mathrm{z}}^S(\mathrm{z})$. For $N > M$, the bias can then be bounded by

$$\begin{aligned}
&\big|\mathbb{E}[\widehat{H}_{\mathrm{UM-tKL}}(X)] - H(X)\big| \\
\leq&\mathbb{E}_{UM}\big|\mathbb{E}_X[\widehat{H}_{\mathrm{UM-tKL}}(X)] - H(X)\big| \\
\leq&\mathbb{E}\big[\frac{C_2^N}{(C_1^N)^{1+1/d}}\big]\big(\frac{k}{N}\big)^{\frac{1}{d}} \\
\leq&C_{UM-tKL}^N\big(\frac{k}{N}\big)^{\frac{1}{d}},
\end{aligned} \tag{4.83}$$

where $C_{UM-tKL}^N = \frac{1}{(1-\bar{C})^{1+1/d}}\mathbb{E}[C_2^N]$. Note that $C_2^N \xrightarrow[N\to\infty]{\mathbb{P}} 0$ and $C_2^N \leq \bar{C}$, *a.s.* for any $N > M$, we have $\lim_{N\to\infty} \mathbb{E}[C_2^N] = 0$ and therefore $\lim_{N\to\infty} C_{UM-tKL}^N = 0$. The MSE can be bounded by

$$\begin{aligned}
&\mathbb{E}[(\widehat{H}_{\mathrm{UM-tKL}}(X) - H(X))^2] \\
\leq&2\mathbb{E}[(\widehat{H}_{\mathrm{UM-tKL}}(X) - \mathbb{E}_X[\widehat{H}_{\mathrm{UM-tKL}}(X)])^2] + 2\mathbb{E}[(\mathbb{E}_X[\widehat{H}_{\mathrm{UM-tKL}}(X)] - H(X))^2] \\
=&2\mathbb{E}_{\mathrm{UM}}\mathbb{E}_X[(\widehat{H}_{\mathrm{UM-tKL}}(X) - \mathbb{E}_X[\widehat{H}_{\mathrm{UM-tKL}}(X)])^2] + 2\mathbb{E}_{\mathrm{UM}}[(\mathbb{E}_X[\widehat{H}_{\mathrm{UM-tKL}}(X)] - H(X))^2]
\end{aligned} \tag{4.84}$$

Note that when $N > M$, C_1^N and C_2^N satisfy Assumption 4.1. Then by Theorem 1, we can bound the first term of Eq. (4.84) by

$$2\mathbb{E}_{\mathrm{UM}}\mathbb{E}_X[(\widehat{H}_{\mathrm{UM-tKL}}(X) - \mathbb{E}_X[\widehat{H}_{\mathrm{UM-tKL}}(X)])^2] \leq C_1\frac{1}{N}, \tag{4.85}$$

for some $C_1 > 0$. The second term of Eq. (4.84) can be bounded by

$$\begin{aligned}
&2\mathbb{E}_{\mathrm{UM}}[(\mathbb{E}_X[\widehat{H}_{\mathrm{UM-tKL}}(X)] - H(X))^2] \\
\leq&2\mathbb{E}\big[\frac{(C_2^N)^2}{(C_1^N)^{2(1+1/d)}}\big]\big(\frac{k}{N}\big)^{\frac{2}{d}} \\
\leq&D_{UM-tKL}^N\big(\frac{k}{N}\big)^{\frac{2}{d}}
\end{aligned} \tag{4.86}$$

where $D_{UM-tKL}^N = \frac{2}{(1-\bar{C})^{2(1+1/d)}}\mathbb{E}[(C_2^N)^2]$. Again, we have, $\lim_{N\to\infty} D_{UM-tKL}^N = 0$ for any $N > M$. Thus, the MSE is bounded by

$$\mathbb{E}[(\widehat{H}_{\mathrm{UM-tKL}}(X) - H(X))^2] \leq C_1\frac{1}{N} + D_{UM-tKL}^N\big(\frac{k}{N}\big)^{\frac{2}{d}}. \tag{4.87}$$

□

4.2.2 Proof of bias and MSE bounds for the UM-tKSG estimator

In this proof, we establish a bias bound for the UM-tKL estimator as $C_{UM-tKSG}\frac{(\log N)^{k+2}}{N^{\frac{1}{d}}}$ and an MSE (Mean Squared Error) bound of $C\frac{(\log N)^{k+2}}{N} + D^N_{UM-tKSG}\frac{(\log N)^{2(k+2)}}{N^{\frac{2}{d}}}$. Here, C is a positive constant , and both $C^N_{UM-tKSG}$ and $D^N_{UM-tKSG}$ are positive constants dependent on $\bar{C}$. Unfortunately, the convergence behavior of $C^N_{UM-tKSG}$ remains undetermined, unlike the leading constant for the bias bound for the UM-tKL estimator. This aspect is earmarked for future research exploration.

Proof. For $N > M$, the bias can be bounded by

$$\begin{aligned}
&\left|\mathbb{E}[\widehat{H}_{\mathrm{UM-tKSG}}(X)] - H(X)\right| \\
\leq& C\mathbb{E}\big[\frac{\bar{p}^S_{\mathrm{z}}\big((\bar{p}^S_{\mathrm{z}})^d + 1\big)}{C_1^{k+1}}\big]\frac{(\log N)^{k+2}}{N^{\frac{1}{d}}} \\
\leq& C_{UM-tKSG}\frac{(\log N)^{k+2}}{N^{\frac{1}{d}}},
\end{aligned} \tag{4.88}$$

where C is a positive constant, $\bar{p}^S_{\mathrm{z}} = \sup_{\mathrm{z}\in\mathcal{Q}} p^S_{\mathrm{z}}(\mathrm{z})$ and $C_{UM-tKSG} = C\frac{(1+\bar{C})\left((1+\bar{C})^d+1\right)}{(1-\bar{C})^{k+1}}$. Similarly as the proof of Corollary 2 and by Theorem 2, we can bound the MSE by

$$\mathbb{E}[(\widehat{H}_{\mathrm{UM-tKSG}}(X) - H(X))^2] \leq C_2\frac{(\log N)^{k+2}}{N} + D^N_{UM-tKSG}\frac{(\log N)^{2(k+2)}}{N^{\frac{2}{d}}}, \tag{4.89}$$

where C_2 is a positive constant and $D^N_{UM-tKSG} = \left(C\frac{(1+\bar{C})\left((1+\bar{C})^d+1\right)}{(1-\bar{C})^{k+1}}\right)^2$. □

Chapter Five

On Estimating the Gradient of the Expected Information Gain in Bayesian Experimental Design

5.1 Introduction

The advancement of science and engineering heavily relies on the acquisition of data through experiments. However, conducting experiments can be resource-intensive and time-consuming. To maximize the information gained from collected data and optimize experimental outcomes, researchers turn to Bayesian experimental design (BED) which offers a systematic and powerful framework for making informed decisions about experimental setups and selecting optimal conditions for data collection. At its core, BED aims to strategically allocate resources to collect the most informative data, which leads to more accurate parameter estimation, model validation, and decision-making. It has been broadly applied in diverse scientific fields, including pharmacokinetic study [143], drug discovery [100], systems biology [86], compressed sensing [148] and physics simulations [102].

Mathematically, BED can be formulated as an optimization problem, where the objective is to maximize a specific function known as the utility function. The choice of the utility function is often driven by the purpose of the BED. In this chapter, our focus lies on BED's application to precise parameter estimation. Various utility functions have been employed to address parameter estimation challenges, and some notable examples include Bayesian A-posterior precision, Bayesian D-posterior precision, quadratic loss and expected information gain (EIG) (see [142] for a review). While the EIG stands out for its exceptional theoretical appeal among these functions, its use has been a long-standing challenge historically due to the computational complexity associated with estimating the evidence or marginal likelihood, which is analytically intractable or computationally expensive to evaluate directly. Consequently, BED based on EIG was once restricted to special cases where analytical solutions or simplifying assumptions allowed for a tractable computation of the evidence [25, 89]. Fortunately, recent advancements in Artificial Intelligence (AI) tools, particularly in neural EIG estimators and automatic differentiation frameworks, have significantly alleviated the computational challenge, enabling the efficient implementation of EIG-based BED in a wide range of non-linear and high-dimensional problems.

Accurate estimation of EIG has been widely recognized as one of the most significant barriers of EIG-based BED. However, our primary interest is not in the exact value of the EIG, but rather in the design variables that maximize the value. Motivated by this viewpoint, an alternative strategy is directly estimating the gradient of EIG w.r.t. the design variables and then using stochastic gradient descent to search for the optimal design. In this chapter, we propose two methods for estimating the EIG gradient. The first method, UEEG-MCMC, applies posterior samples generated by Markov Chain Monte Carlo (MCMC) to estimate the EIG gradient. It is shown effective across different scenarios, regardless of the ground-truth EIG values. The second method, BEEG-AP, is more simulation-efficient. However, its performance suffers when dealing with problems that have large ground-truth

EIG values. The chapter establishes a connection with nested Monte Carlo to analyze this behavior, shedding light on the limitations of BEEG-AP in such cases.

We validate the aforementioned attributes of the two proposed methods through a meticulous numerical experiment and diverse applications featuring varying expected EIG levels. Additionally, comprehensive comparisons are made with several bench-marking approaches, revealing the superior performance of our proposed methods.

The remainder of this chapter is organized as follows. We first finish this section with a review of the related work of EIG-based BED. Then in Section 5.2, we introduce the fundamental concepts and problem settings. Following that, in Section 5.3, we present a novel representation of the EIG gradient, and Section 5.4 details the corresponding estimation methods. To showcase the effectiveness of these methods, we present numerical experiments in Section 5.5. Finally, Section 5.8 summarizes our contributions.

5.1.1 Related Work

Rainforth et al.[134] provide a thorough review of modern methods for Bayesian experimental design. Early schemes for Bayesian experimental design used separate stages to estimate the Expected Information Gain (EIG) and optimize the design variables λ, both of which can be challenging tasks. For EIG estimation, the main computational challenge arises from the intractability of $p(y|\lambda)$ and several methods have been proposed to solve this problem. Notably, Nested Monte Carlo (NMC) [135, 144] emerged as a prominent method in this area. Additionally, Variational EIG estimators (also known as variational mutual information estimators) [49, 13, 41, 113] combined with deep learning techniques [19, 114, 4, 156] showed significant progress. Furthermore, alternative approaches for EIG estimation included using ratio estimation [165] as proposed in [82], and bounding EIG from below by two or more

entropies in the data space [6] which are then be estimated by entropy estimation methods [85]. A direct lower bound estimation for EIG was introduced in [169] for models with fixed normally distributed measurement noises, and the EIG Laplace approximation was proposed in [96]. Regarding the optimization of λ, conventional gradient-free approaches such as Bayesian optimization (BO) [155], Simultaneous perturbation stochastic approximation (SPSA) [158], simulation-based optimization (SBO)[108], and Nelder-Mead nonlinear simplex (NMNS)[112] were commonly employed. However, these methods encountered scalability issues when dealing with high-dimensional design variable spaces.

Recent advancements have introduced efficient gradient-based approaches that leverage the reparameterization trick and automatic differentiation frameworks. These methods, such as those proposed in [48, 81, 182, 181], allow for simultaneous optimization of both the variational parameters and the design variables. Moreover, Goda et al. [57] presented a method that directly obtains an unbiased estimator of the EIG gradient using a randomized version of multilevel Monte Carlo (MLMC) method [139]. Furthermore, gradient estimators for implicit models [90, 176, 91, 151] with score matching techniques [75, 157] have emerged as another avenue for optimizing λ in a gradient-based way. Despite receiving limited attention in the Bayesian experimental design community, these methods hold promise and deserve further exploration.

5.2 Preliminary Knowledge

5.2.1 Problem Formulation

This section defines the variables and functions involved as well as the primary objective of Bayesian experimental design. Let $\lambda \in \mathcal{D}$ be the design variables that can be controlled

by users. The parameters to be inferred from the observed data y are denoted by θ and $\pi_\theta(\theta)$ denotes its prior that represents our knowledge or belief about the parameters before observing any data. ϵ represents the base model noises generated from a known distribution $\pi_\epsilon(\epsilon)$. The process of simulating the observed data can be modeled by a sampling path

$$y = g(\theta, \epsilon, \lambda). \tag{5.1}$$

In this context, we assume the existence of a tractable likelihood function $l(y|\theta, \lambda)$, which is derived from the sampling path. However, it is important to note that the marginal likelihood $p(y|\lambda)$ is not available in closed-form. Under certain design λ, the expected information gain (EIG) is defined as

$$U(\lambda) = \mathbb{E}_{\pi_{\theta(\theta)l(y|\theta,\lambda)}}[\log l(y|\theta, \lambda)] - \mathbb{E}_{p(y|\lambda)}[\log p(y|\lambda)]. \tag{5.2}$$

It represents the expected amount of information that observations y provide about θ and is well known as the mutual information [33] between θ and y in information theory community. The objective of Bayesian experimental design is then to maximize the EIG over the design variable space $\mathcal{D}$

$$\lambda^* = \arg\max_{\lambda \in \mathcal{D}} U(\lambda). \tag{5.3}$$

5.2.2 Simulation Cost

Estimating and optimizing the EIG entails generating observation samples and evaluating the corresponding likelihood values. This process can be computationally demanding, and its efficiency is of paramount importance in practical applications. In this section, we will illustrate how to quantify the simulation cost during the process when explicit models are considered. For explicit models, the sampling path can typically be written as a hierarchical structure

$$y = g(f(\theta, \lambda), \epsilon, \theta, \lambda), \tag{5.4}$$

where f is the forward model which dominates the main computational cost. Examples of the commonly used hierarchical structure include:

$$\text{Additive noise}: y = f(\theta, \lambda) + \sigma(\theta, \lambda)\epsilon. \tag{5.5}$$

$$\text{Multiplicative noise}: y = f(\theta, \lambda)(1 + \sigma(\theta, \lambda)\epsilon). \tag{5.6}$$

$$\text{Mixture of noises}: y = f(\theta, \lambda)(1 + \sigma_1(\theta, \lambda)\epsilon_1) + \sigma_2(\theta, \lambda)\epsilon_2. \tag{5.7}$$

For simplicity, we take the case of additive noise for example to analyze the simulation cost concerned in applications. Given a fixed parameter θ^*, simulating multiple observations (e.g. $y^{(k)} = f(\theta^*, \lambda) + \sigma(\theta^*, \lambda)\epsilon^{(k)}$, $k = 1, ..., K$) only involves a single simulation (i.e. $f(\theta^*, \lambda)$) of the forward model. Likewise, since the likelihood function can be analytically written as

$$l(y|\theta, \lambda) = \pi_\epsilon\Big(\frac{y - f(\theta, \lambda)}{\sigma(\theta, \lambda)}\Big), \tag{5.8}$$

evaluating the values of the likelihood w.r.t. multiple observations (e.g. $l(y^{(k)}|\theta^*, \lambda)$, $k = 1, ..., K$) also requires only a single forward pass of f. Thus, the simulation cost of generating observations from and evaluating the likelihood is directly related to the number of different parameters involved. This observation is important for the analysis of simulation costs of estimators in the later sections.

5.3 Posterior Expected Representations of the EIG Gradient

In this section, we introduce a novel representation of the EIG gradient that offers new insights into the development of efficient gradient estimators. To start with, we analyze the difficulty in directly computing the EIG gradient w.r.t. the design variables λ. Estimating the gradient of Eq. (5.2) directly with score-function estimators [106] could lead to high

variance. As a result, practitioners often turn to pathwise gradient estimators (also known as the reparameterization tricks) [106], as an alternative strategy for estimating this gradient:

$$\begin{aligned}\nabla_\lambda U(\lambda) =& \nabla_\lambda \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}[\log l(g(\theta,\epsilon,\lambda)|\theta,\lambda)] - \nabla_\lambda \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}[\log p(g(\theta,\epsilon,\lambda)|\lambda)] \\ =& \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}[\nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta,\lambda)] - \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}[\nabla_\lambda \log p(g(\theta,\epsilon,\lambda)|\lambda)].\end{aligned} \tag{5.9}$$

While obtaining $\nabla_\lambda g(\theta,\epsilon,\lambda)$ is typically straightforward using modern automatic differentiation frameworks (e.g. Tensorflow [1] and Pytorch [125]), the score functions $\nabla_y \log p(y|\lambda)$ and $\nabla_\lambda \log p(y|\lambda)$ usually do not have analytical forms, rendering the second term of the above estimator intractable. To address this challenge, we apply the key idea in [26], which involves using the tractable scores of likelihood $\nabla_y \log l(y|\theta,\lambda)$ and $\nabla_\lambda \log l(y|\theta,\lambda)$ to estimate the intractable score functions $\nabla_y \log p(y|\lambda)$ and $\nabla_\lambda \log p(y|\lambda)$. This can be summarized as the following Lemma 5.1.

Lemma 5.1. *The gradient of the logarithm of the marginal density w.r.t. the experimental condition λ admits the following representation:*

$$\nabla_\lambda \log p(g(\theta,\epsilon,\lambda))|\lambda) = -\mathbb{E}_{q(\theta'|g(\theta,\epsilon,\lambda),\lambda)}[\nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta',\lambda)], \tag{5.10}$$

where $q(\theta'|y,\lambda) \propto \pi_\theta(\theta')l(y|\theta',\lambda)$ is the posterior density of parameters given the observation sample y.

Using this lemma, we derive an entropy gradient estimator for the marginal distribution of y as stated in Theorem 5.1.

Theorem 5.1. *The gradient of the entropy $H(p(y|\lambda))$ w.r.t. the experimental condition λ satisfies*

$$\nabla_\lambda H(p(y|\lambda)) = -\mathbb{E}_{\pi_\theta(\theta)\pi_\epsilon(\epsilon)q(\theta'|g(\theta,\epsilon,\lambda),\lambda)}[\nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta',\lambda)], \tag{5.11}$$

where $q(\theta'|y,\lambda) \propto \pi_\theta(\theta')l(y|\theta',\lambda)$ is the posterior density of parameters given the observation sample y.

Using Theorem 5.1, we can get a posterior expected representation of the EIG gradient, as stated in the following Corollary 5.1.

Corollary 5.1. *The gradient of the EIG $U(\lambda)$ w.r.t. the experimental condition λ satisfies*

$$\nabla_\lambda U(\lambda) = \mathbb{E}_{\pi_\theta(\theta)\pi_\epsilon(\epsilon)q(\theta'|g(\theta,\epsilon,\lambda),\lambda)}[\nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta,\lambda) - \nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta',\lambda)], \tag{5.12}$$

where $q(\theta'|y,\lambda) \propto \pi_\theta(\theta')l(y|\theta',\lambda)$ is the posterior density of parameters given the observation sample y.

5.4 Estimating the EIG Gradient

Building upon the posterior expected representation of the EIG gradient in Eq. (5.12), we propose two estimators of EIG gradient. When integrated with stochastic gradient descent algorithms, these estimators seamlessly evolve into the respective algorithms for Bayesian experimental design. For simplicity, we denote the observation samples generated from the sampling path as $y^{(i)}(\lambda) = g(\theta^{(i)}, \epsilon^{(i)}, \lambda)$ throughout this section.

5.4.1 Unbiased Estimation of EIG Gradient with Markov Chain Monte Carlo

The most straightforward method to estimate the expectation in Eq. (5.12) is utilizing MCMC schemes. Specifically, giving samples $\{\theta^{(i)}\}_{i=1}^M$ and $\{\epsilon^{(i)}\}_{i=1}^M$ drawn from $\pi_\theta(\theta)\pi_\epsilon(\epsilon)$, the expectation in Eq. (5.12) can be estimated by Monte Carlo average as

$$\nabla_\lambda U(\lambda) \approx \frac{1}{M}\sum_{i=1}^M \nabla_\lambda \log l(y^{(i)}(\lambda)|\theta^{(i)},\lambda) - \frac{1}{M}\sum_{i=1}^M \mathbb{E}_{q(\theta'|y^{(i)}(\lambda),\lambda)}[\nabla_\lambda \log l(y^{(i)}(\lambda)|\theta',\lambda)]. \tag{5.13}$$

Ideally if we can draw samples $\{\theta^{(i,j)}\}_{j=1}^{N}$ exactly from the posterior $q(\theta'|y^{(i)}(\lambda),\lambda)$, we can obtain an unbiased estimator of $\mathbb{E}_{q(\theta'|y^{(i)}(\lambda),\lambda)}[\nabla_\lambda \log l(y^{(i)}(\lambda)|\theta',\lambda)]$:

$$\mathbb{E}_{q(\theta'|y^{(i)}(\lambda),\lambda)}[\nabla_\lambda \log l(y^{(i)}(\lambda)|\theta',\lambda)] \approx \frac{1}{N}\sum_{j=1}^{N} \nabla_\lambda \log l(y^{(i)}(\lambda)|\theta^{(i,j)},\lambda). \tag{5.14}$$

Combining Eq. (5.13) and Eq. (5.14) yields an unbiased estimator of EIG gradient in theory. In reality however, one often relies on Markov Chain Monte Carlo (MCMC) methods to sample the posterior distribution, which draws biased samples from the posterior distribution. We refer to the method as unbiased estimation of EIG gradient with MCMC (UEEG-MCMC). The simulation cost of a single gradient estimation for UEEG-MCMC is $O(M \times L)$, where L is the number of simulations used to perform MCMC. We reinstate that a finite-length MCMC can not produce unbiased samples from the posterior and as such it causes bias in the gradient estimator. Nevertheless, we emphasize that the bias lies in the samples and the estimator itself is unbiased provided that samples are generated perfectly from the posterior. As will be shown in the numerical examples, the bias due to MCMC is often much smaller than those in other methods especially for problems with large EIG values. Moreover, while we adopt MCMC for sampling the posterior here, the proposed method can be implemented with any sampling methods. To this end, if more effective sampling methods are available, they can be used instead of MCMC to reduce the estimation bias.

5.4.2 Biased Estimation of EIG Gradient with Atomic Priors

Generating an observation from or evaluating the likelihood for each new parameter sample requires to simulate the physical model considered once more. For expensive physical models, this constitutes the most significant computational cost.

In this section, we show how to obtain a simulation-efficient approach using *atomic* priors. Suppose a finite set of parameter-noise pairs $\Omega = \{(\theta^{(i)}, \epsilon^{(i)})\}_{i=1}^{M}$ are generated, where

$(\theta^{(i)}, \epsilon^{(i)}) \sim \pi_\theta(\theta)\pi_\epsilon(\epsilon)$, and we denote $\Theta = \{\theta^{(i)}\}_{i=1}^M$. Replacing the sampling distributions $\pi_\theta(\theta)\pi_\epsilon(\epsilon)$ by U_Ω and $\pi_\theta(\theta')$ by U_Θ, where U denotes the uniform distribution on the given set, we can approximate the sampling distribution over which the expected value is taken in Eq. (5.12) as

$$\begin{aligned} &\pi_\theta(\theta)\pi_\epsilon(\epsilon)q(\theta'|g(\theta,\epsilon,\lambda),\lambda) \\ =&\frac{\pi_\theta(\theta)\pi_\epsilon(\epsilon)\pi_\theta(\theta')l(g(\theta,\epsilon,\lambda)|\theta',\lambda)}{\int \pi_\theta(\theta')l(g(\theta,\epsilon,\lambda)|\theta',\lambda)d\theta'} \\ \approx&\sum_{i=1}^M \frac{\sum_{j=1}^M \delta_{\theta^{(i)}}(\theta)\delta_{\epsilon^{(i)}}(\epsilon)\delta_{\theta^{(j)}}(\theta')l(y^{(i)}(\lambda)|\theta^{(j)},\lambda)}{\sum_{j=1}^M l(y^{(i)}(\lambda)|\theta^{(j)},\lambda)}. \end{aligned} \tag{5.15}$$

Given this approximation of sampling distribution, we finally obtain a biased estimator of EIG gradient

$$\begin{aligned} &\nabla_\lambda U(\lambda) \\ \approx&\sum_{i=1}^M \frac{\sum_{j=1}^M l(y^{(i)}(\lambda)|\theta^{(j)},\lambda)\nabla_\lambda \log\left[\frac{l(y^{(i)}(\lambda)|\theta^{(i)},\lambda)}{l(y^{(i)}(\lambda)|\theta^{(j)},\lambda)}\right]}{\sum_{j=1}^M l(y^{(i)}(\lambda)|\theta^{(j)},\lambda)}. \end{aligned} \tag{5.16}$$

we refer to it as biased estimation of EIG gradient with atomic priors (BEEG-AP). As it requires only one batch of parameter samples for each gradient estimation, the simulation cost amounts to $O(M)$.

5.4.3 Unifying BEEG-AP and NMC

To provide a more comprehensive understanding of BEEG-AP, this section reveals its close connection to nested Monte Carlo (NMC). Indeed, BEEG-AP can be regarded as an approach that directly computes the gradient of the NMC estimator with sample reuse technique (srNMC) in [74]. We start this section by revisiting the concepts of NMC and srNMC.

The naïve NMC estimates the EIG as

$$U(\lambda) \approx \frac{1}{M}\sum_{i=1}^M \log \frac{l(y^{(i)}(\lambda)|\theta^{(i)},\lambda)}{\frac{1}{N}\sum_{j=1}^N l(y^{(i)}(\lambda)|\theta^{(i,j)},\lambda)}, \tag{5.17}$$

where $\theta^{(i)} \sim \pi_\theta(\theta)$, $\epsilon^{(i)} \sim \pi_\epsilon(\epsilon)$ and $\theta^{(i,j)} \sim \pi_\theta(\theta)$. This approximation requires a simulation cost of $O(M \times N)$. To reduce the cost to $O(M)$, Huan & Marzouk [74] propose reusing the batch of prior samples for the outer Monte Carlo sum in all inner Monte Carlo estimations (i.e., $\theta^{(i,j)} = \theta^{(j)}$ and $N = M$). This yields a more simulation-efficient estimator of EIG

$$\widehat{U}^M_{srNMC}(\lambda) = \frac{1}{M}\sum_{i=1}^{M} \log \frac{l(y^{(i)}(\lambda)|\theta^{(i)}, \lambda)}{\frac{1}{M}\sum_{j=1}^{M} l(y^{(i)}(\lambda)|\theta^{(j)}, \lambda)}. \tag{5.18}$$

This estimator is also related to the InfoNCE with a tractable conditional [114, 129], often utilized for the mutual information estimation in representation learning. In addition, similar sample reuse techniques have been applied to portfolio risk measurement problems [183, 46].

Now it is evident that the BEEG-AP can be directly derived from the gradient of Eq. (5.18) w.r.t. λ. This observation allows us to explore the theoretical behavior of the BED with BEEG-AP by investigating the convergence properties of the srNMC. The original paper of [74] only provides a simple numerical study of the bias of srNMC. Here, we give a more rigorous convergence analysis as the following theorems.

Theorem 5.2. *The expectation of $\widehat{U}^M_{srNMC}(\lambda)$ satisfies the following:*

1. $\mathbb{E}[\widehat{U}^M_{srNMC}(\lambda)]$ *is a lower bound on $U(\lambda)$ for any $M > 0$.*
2. $\mathbb{E}[\widehat{U}^M_{srNMC}(\lambda)]$ *is monotonically increasing in M, i.e., $\mathbb{E}[\widehat{U}^{M_1}_{srNMC}(\lambda)] \leq \mathbb{E}[\widehat{U}^{M_2}_{srNMC}(\lambda)]$ for $0 \leq M_1 \leq M_2$.*

The expectations above are taken over the distributions of the involved θ and y samples.

Theorem 5.3. *If $l(g(\theta, \epsilon, \lambda)|\theta', \lambda)$ is bounded away from 0 and uniformly bounded from above (i.e., $C_1 \leq l(g(\theta, \epsilon, \lambda)|\theta', \lambda) \leq C_2$ a.s. for some positive constants C_1 and C_2), then the mean squared error of $\widehat{U}^M_{srNMC}(\lambda)$ converges to 0 at rate $O(1/M)$.*

Theorem 5.2 indicates that, in expectation, the srNMC provides a lower bound estimation for the EIG and the gap can be increasingly narrowed as M increases. This

suggests that BED with BEEG-AP aims to use stochastic gradients to maximize a lower bound of the ground-truth EIG. Theorem 5.3 further establishes the consistency of srNMC and obtains a linear convergence rate of $O(1/M)$ under certain assumptions, indicating that the optimization objective of BED with BEEG-AP can be sufficiently close to true EIG as we increase the number of samples M. However, the following Theorem 5.4 suggests that achieving a negligible error is challenging in practice when the ground-truth EIG is large, even when these assumptions are admitted.

Theorem 5.4. *For any C satisfying $0 \leq C \leq U(\lambda)/2$, if $M \leq \exp(U(\lambda)/2)$, we have*

$$U(\lambda) - \widehat{U}^{M}_{srNMC}(\lambda) > C. \tag{5.19}$$

Indeed, this theorem tells us that the simulation cost required grows exponentially with the ground-truth EIG to achieve a reasonable error bound.

5.5 Experiments

A large variety of BED methods could exhibit poor performance due to the presence of large EIG values in the experiments to be designed. Before diving into numerical demonstrations (see https://github.com/ziq-ao/GradEIG for the research code and Appendix for further details of our experiments), we list a few bench-marking approaches and briefly discuss the above limitation they have in common.

PCE. Prior contrastive estimation (PCE) [48] estimates the EIG as

$$\widehat{U}^{M,N}_{PCE}(\lambda) = \frac{1}{M}\sum_{i=1}^{M} \log \frac{l(y^{(i)}(\lambda)|\theta^{(i)},\lambda)}{\frac{1}{N+1}\sum_{j=0}^{N} l(y^{(i)}(\lambda)|\theta^{(i,j)},\lambda)}, \tag{5.20}$$

where $\theta^{(i)}, \epsilon^{(i)} \sim \pi_\theta(\theta)\pi_\epsilon(\epsilon)$, $\theta^{(i,0)} = \theta^{(i)}$ and $\theta^{(i,j)} \sim \pi_\theta(\theta)$ for $j = 1, ..., N$. In contrast to srNMC, PCE only reuses one outer Monte Carlo sample in each inner Monte Carlo estimation,

resulting in a simulation cost of $O(M \times N)$. It is easy to check that $\widehat{U}^{M,N}_{PCE}(\lambda)$ does not exceed $\log N$, so PCE shares a similar result to the one stated in Theorem 5.4, implying its simulation inefficiency for estimating large EIG values.

ACE. To improve the inner Monte Carlo in the denominator, adaptive contrastive estimation (ACE) [48] introduces a posterior inference network q_ϕ parameterized by ϕ and use it as the proposal distribution for sampling, i.e.,

$$\widehat{U}^{M,N}_{ACE}(\lambda) = \frac{1}{M}\sum_{i=1}^{M} \log \frac{l(y^{(i)}(\lambda)|\theta^{(i)},\lambda)}{\frac{1}{N+1}\sum_{j=0}^{N} \frac{\pi_\theta(\theta^{i,j}) l(y^{(i)}(\lambda)|\theta^{(i,j)},\lambda)}{q_\phi(\theta^{(i,j)}|y^{(i)}(\lambda))}}, \tag{5.21}$$

where $\theta^{(i)}, \epsilon^{(i)} \sim \pi_\theta(\theta)\pi_\epsilon(\epsilon)$, $\theta^{(i,0)} = \theta^{(i)}$ and $\theta^{(i,j)} \sim q_\phi(\theta^{(i,j)}|y^{(i)}(\lambda))$ for $j = 1, ..., N$. Given this adaptive estimator, the network parameter ϕ and design variable λ are then optimized jointly. However, learning the posterior inference network can be challenging when there are strong dependencies between the conditional (the observations) and target variables (the parameters). This occurs when the ground-truth EIG values are large (i.e., there is a high mutual information between parameters and observations). In practice, we observe that general-purpose conditional density networks (such as Mixture Density Network [23] and Normalizing Flows [122]) usually fail or run indefinitely for invalid values during training in this case.

GradBED. Gradient-based Bayesian Experimental Design (GradBED) [83] designs experiments by optimizing a variational lower bound of mutual information between parameters and observations. A variety of candidates of lower bounds can be found in [83]. In this chapter, we only consider the following NWJ estimator [113]:

$$\widehat{U}^{M}_{NWJ}(\lambda) = \frac{1}{M}\sum_{i=1}^{M} \left[T_\psi(\theta^{(i)}, y^{(i)}(\lambda)) - \frac{1}{e}\exp\left(T_\psi(\theta^{(i)}, y'^{(i)}(\lambda)\right)\right], \tag{5.22}$$

where $\theta^{(i)}, \epsilon^{(i)} \sim \pi_\theta(\theta)\pi_\epsilon(\epsilon)$, $\theta'^{(i)}, \epsilon'^{(i)} \sim \pi_\theta(\theta)\pi_\epsilon(\epsilon)$, $y'^{(i)}(\lambda)) = g(\theta'^{(i)}, \epsilon'^{(i)}, \lambda)))$ and T_ψ is a neural network parametrised by ψ. The simulation cost is $O(M)$ for GradBED. During optimization, the network parameter ψ and design variable λ are updated simultaneously. As

studied in [156], the variance of certain variational mutual information estimators, including NWJ, could grow exponentially with the ground-truth mutual information (or EIG in Bayesian experimental design literature) and thereby lead to poor designs.

5.5.1 EIG Gradient Estimation Accuracy

We start by examining the empirical convergence properties of the proposed estimators. We consider a Bayesian linear regression model with tractable EIG gradients. Assume $n \times 1$ observations are generated by the following linear acquisition system

$$\mathrm{y} = \mathrm{D}\theta + \epsilon \tag{5.23}$$

where $\mathrm{D} = [1, \lambda', (\lambda')^2]$ is the design matrix obtained by the design vector $\lambda = (\lambda_1, ..., \lambda_n)$, $\theta = (\theta_1, \theta_2, \theta_3)'$ are the parameters of interest and ϵ are $n \times 1$ i.i.d. noises. In Bayesian framework, we assign a Gaussian prior $\theta \sim \mathcal{N}(0, I_3)$ on the unknown parameters and a Gaussian observation noise with variance σ^2, that is, $p(\mathrm{y}|\theta) = \mathcal{N}(\mathrm{D}\theta, \sigma^2 I_n)$. The above modeling admits a closed-form representation of EIG using the entropy expressions of multivariate normal distributions[3], that is,

$$U(\lambda) = \frac{1}{2}(\log \frac{|\mathrm{DD}' + \sigma^2 I_n|}{|\sigma^2 I_n|}). \tag{5.24}$$

The EIG gradient can then be analytically derived or directly computed by automatic differentiation frameworks.

In this study, we estimate and compare the biases of three approaches (BEEG-AP, UEEG-MCMC and PCE) with a large number of repeated trials. The scatter plots in Fig. 5.1 shows the comparisons of these estimated biases across 20 independent designs. From the top of Fig. 5.1, it is evident that the UEEG-MCMC has lower bias with the ground-truth EIG increases, outperforming the other two methods. On the other hand, the bottom of Fig. 5.1

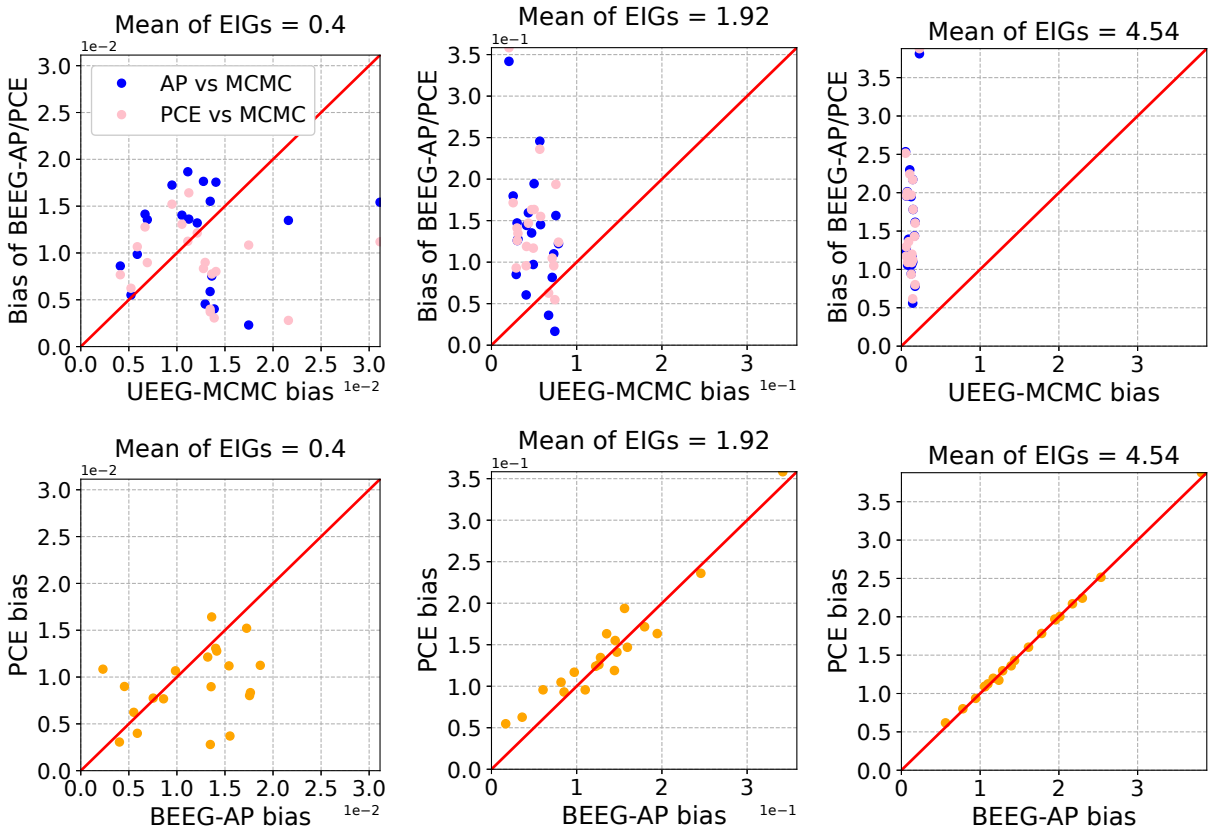


Figure 5.1: Top: the estimated biases of BEEG-AP and PCE versus those of UEEG-MCMC for 20 independent designs. Bottom: the estimated biases of PCE versus those of BEEG-AP for 20 independent designs.

exhibits that, while BEEG-AP requires fewer simulation costs, it yields a comparable level of bias to PCE.

5.5.2 A Toy Algebraic Model

In this experiment, we consider a toy problem with a single optimal design. The model is given by the following nonlinear map:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0.5\theta^3 d_1 + \theta \exp(-|0.2 - 0.5d_1|) + d_1^2 \\ 0.5\theta^3(d_2+1.6) + \theta \exp(-|0.6+0.5d_2|) + d_2^2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix}, \tag{5.25}$$

where θ is the model parameter, d_i are design variables and ϵ_i are independent observation noises. We assign a uniform prior $\theta \sim \text{Unif}(0, 1)$ on the parameter and restrict the design variables in the interval of $[0, 1]$. We conduct two experiments with different noise terms, i.e., a large noise scenario with $\epsilon_i \sim \mathcal{N}(0, (0.1)^2)$ and a small noise scenario with $\epsilon_i \sim \mathcal{N}(0, (0.0001)^2)$. By setting different noise terms, we can create scenarios to represent the small and large EIG cases and observe how the different methods perform under these conditions.

We applied five methods to this problem: BEEG-AP, UEEG-MCMC, ACE, PCE and GradBED. To mitigate the impact of randomness, we perform 20 independent runs for each method. The results for both the large and small noise settings are depicted in Fig. 5.2. From the figures we can see the final designs obtained by BEEG-AP and UEEG-MCMC are more concentrated compared to the designs generated by the other three methods. In particular, in the small noise case, UEEG-MCMC stands out as the only method where all designs eventually concentrate on a single point. Then we use different metrics to judge the quality of the designs obtained for the two settings. In the large noise case, we apply NMC with large samples to obtain high-quality estimations of the EIGs. Fig. 5.3 shows the estimated EIGs throughout the entire design space. Remarkably, we observe that the only optimal design identified by the estimations aligns with the the results obtained by BEEG-AP and UEEG-MCMC. In the small noise case, utilizing NMC for reliable EIG estimations becomes impractical due to the large simulation budget required. We therefore resort to using the posterior entropy as the metric to evaluate the quality of the designs, and the results are plotted in Fig. 5.4. From the figure, it is evident that BEEG-AP and UEEG-MCMC produce designs with smaller posterior entropy. Specifically, UEEG-MCMC yields even smaller posterior entropy than BEEG-AP, which demonstrates the superior performance of UEEG-MCMC in large EIG case.

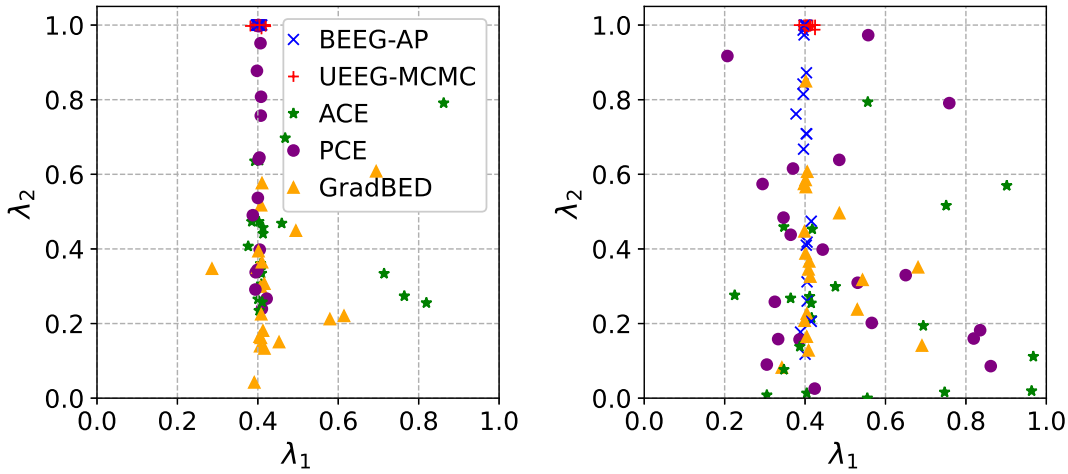


Figure 5.2: The final designs of 20 independent trials for large noise setting (left) and small noise setting (right).

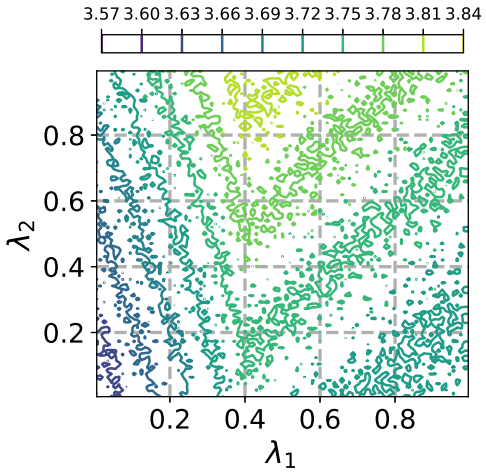


Figure 5.3: Estimates of EIG for large noise setting.

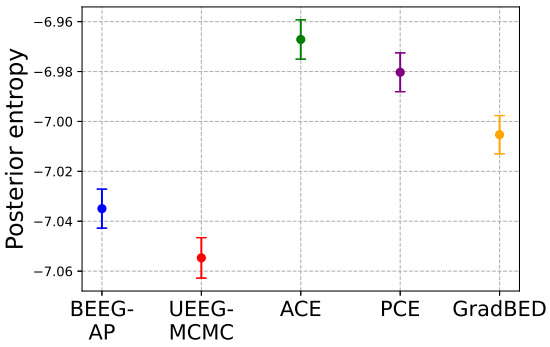


Figure 5.4: The posterior entropy for small noise setting. Shown are the means of entropy with their standard error bars.

5.5.3 Pharmacokinetic (PK) Model

Next, we consider an experimental design problem for PK studies. These studies aim to comprehend the underlying kinetics of a drug, shedding light on how it is absorbed, distributed, metabolized, and eliminated within the body over time. To achieve this understanding, it is a common practice to collect blood samples from the study subjects. However, the process of blood sampling involves various practical constraints, including financial limitations, participant burden, and ethical considerations. Therefore, researchers must strategically design the sampling strategy to gather meaningful information while minimizing the number of blood samples required. Here, we focus on the PK model introduced by [143]. The model under consideration consists of three parameters of interest $\theta = (k_a, k_e, V)$: the absorption rate constant k_a, the elimination rate constant k_e, and the volume of distribution V which represents the theoretical volume that the drug would need to occupy to achieve the current concentration in the blood plasma. The drug concentration of blood sample taken at time t, denoted as y_t, follows

$$y_t = \frac{D}{V} \cdot \frac{k_a}{k_a - k_e} \cdot (e^{-k_e t} - e^{-k_a t}) \cdot (1 + \epsilon_{1t}) + \epsilon_{2t}, \tag{5.26}$$

where $D = 400$ is the fixed dose administered at the beginning of the experiment, ϵ_{1t} and ϵ_{2t} are the multiplicative and additive Gaussian noises respectively. As in [143], we assign a log-normal prior on θ

$$\log \theta \sim N \left[\begin{pmatrix} \log(1) \\ \log(0.1) \\ \log(20) \end{pmatrix}, \begin{pmatrix} 0.05 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.05 \end{pmatrix} \right]. \tag{5.27}$$

The design variables are assumed to be the 10 blood sampling times $(d_1, ..., d_{10})$, $d_i \geq 0$ for $i = 1, ..., 10$, and the corresponding drug concentrations at these times $(y_{d_1}, ..., y_{d_{10}})$ form the observations.

Again, we create a small EIG setting with $\epsilon_{1t} \sim \mathcal{N}(0, 0.01)$ and $\epsilon_{2t} \sim \mathcal{N}(0, 0.1)$, and

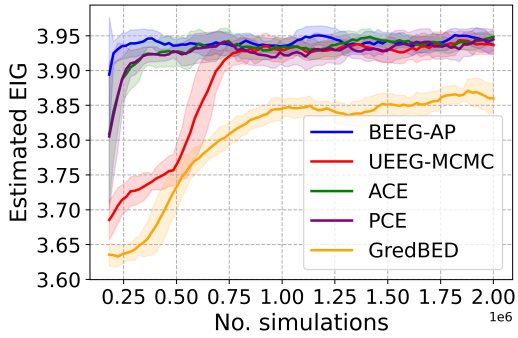


Figure 5.5: Optimization of EIG for PK model with multiplicative noise $\mathcal{N}(0, 0.01)$ and additive noise $\mathcal{N}(0, 0.1)$ as a function of number of simulations. Shown are the moving averages with the standard error bars.

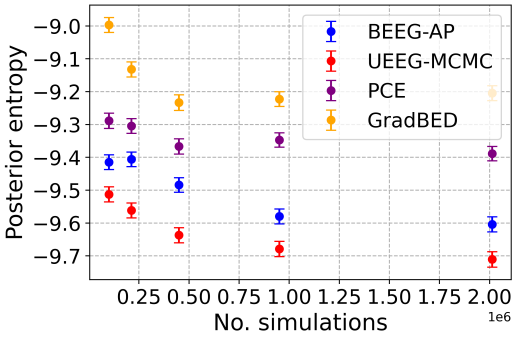


Figure 5.6: The posterior entropy for PK model with additive noise $\mathcal{N}(0, 0.001)$ as a function of number of simulations. Shown are the means of entropy with their standard error bars.

a large EIG setting with $\epsilon_{1t} = 0$ and $\epsilon_{2t} \sim \mathcal{N}(0, 0.001)$. In small EIG scenario we use large samples to compute high-quality NMC estimations to validate various methods, while in large EIG scenario we compare the posterior entropies obtained by them. Fig. 5.5 shows BEEG-AP outperforms other methods in terms of convergence rate in small EIG scenario. This can also be supported by examining the convergence histories of all methods, as shown in Fig. 5.9 in the Appendix. However, in large EIG scenario, UEEG-MCMC achieve the best performance as the validation experiments indicate in Fig. 5.6 and the convergence histories show in Fig. 5.10 in the Appendix. It should be mentioned that in this case ACE fails to learn a posterior inference network due to the strong dependencies between the observations and the parameters.

5.5.4 Signal Transducer and Activator of Transcription 5 (STAT5) model

Finally we aim to design the measurement times for a dynamic system modeled by ordinary differential equations (ODEs). We take the mathematical model of the core module of the Janus family of kinases (JAK)–signal transducer and activator of transcription (STAT) pathway in [164] as a case study. The core module of the JAK-STAT pathway is represented by the latent transcription factor STAT5, and the dynamics of STAT5 populations x_1, x_2, x_3 and x_4 can be described by four coupled ODEs (see Appendix for full details of the ODEs). The rate constants k_1, k_2 and the delay parameter τ are the three model parameters to be inferred from measured data.

It is experimentally challenging to directly measure distinct STAT5 populations separately. Instead, one can measure the amount of tyrosine phosphorylated STAT5 $y_1 = s_1(x_2 + x_3)$ and the total amount of STAT5 $y_2 = s_2(x_1 + x_2 + x_3)$. We assume the scaling parameters to be $s_1 = 0.33$ and $s_2 = 0.26$. We assign a uniform prior on $\theta = (k_1, k_2, \tau)$ with lower range $[0.5, 0.05, 4.0]$ and upper range $[3.0, 0.2, 10.0]$. The objective of the experimental design is to allocate 16 measurement times for STAT5 populations over a time span from 0 to 60 minutes, which yields 32 experimental measurements in total.

In this application, we set two levels of additive Gaussian observation noises, $\mathcal{N}(0, 10^{-4})$ and $\mathcal{N}(0, 10^{-6})$, representing the small and large EIG scenarios respectively. In both cases, we assess the quality of designs by the posterior entropies obtained. The results for the small EIG scenario depicted in Fig. 5.7 indicate that, both BEEG-AP and UEEG-MCMC outperform other methods and BEEG-AP appears the best. However, UEEG-MCMC demonstrates the best performance in the large EIG scenario, while BEEG-AP's performance falls below that of GradBED, as shown in Fig. 5.8. ACE fails in both scenario. The convergence histories of all approaches can be found in Fig. 5.11 and Fig. 5.12 in the Appendix.

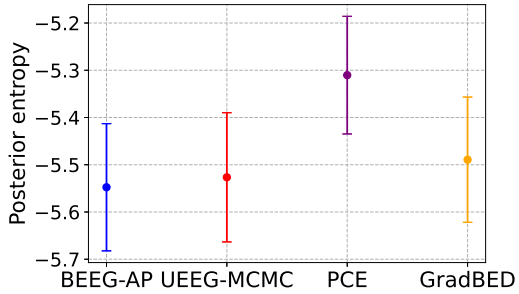


Figure 5.7: The posterior entropies for STAT5 model with additive noise $\mathcal{N}(0, 10^{-4})$ for designs obtained. Shown are the means of entropy with their standard error bars.

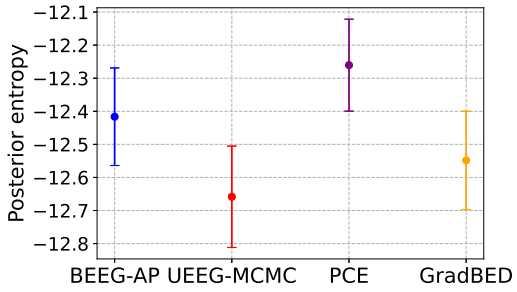


Figure 5.8: The posterior entropies for STAT5 model with additive noise $\mathcal{N}(0, 10^{-6})$ for designs obtained. Shown are the means of entropy with their standard error bars.

5.6 Proofs of Results

Lemma 5.1. *The gradient of the logarithm of the marginal density w.r.t. the experimental condition λ admits the following representation:*

$$\nabla_\lambda \log p(g(\theta, \epsilon, \lambda))|\lambda) = -\mathbb{E}_{q(\theta'|g(\theta,\epsilon,\lambda),\lambda)}[\nabla_\lambda \log l(g(\theta, \epsilon, \lambda)|\theta', \lambda)], \tag{5.28}$$

where $q(\theta'|y, \lambda) \propto \pi_\theta(\theta')l(y|\theta', \lambda)$ is the posterior density of parameters given the observation sample y.

Proof. Let $y = g(\theta, \epsilon, \lambda)$, we find

$$\begin{aligned}
&\nabla_\lambda \log p(g(\theta,\epsilon,\lambda)|\lambda) \\
=&\nabla_\lambda \log p(y|\lambda) \\
=&\frac{1}{p(y|\lambda)} \nabla_\lambda p(y|\lambda) \\
=&\frac{1}{p(y|\lambda)} \nabla_\lambda \int \pi_\theta(\theta') l(y|\theta',\lambda) d\theta' \\
=&\frac{1}{p(y|\lambda)} \int \pi_\theta(\theta') \nabla_\lambda l(y|\theta',\lambda) d\theta' \\
=&\int q(\theta'|y,\lambda) \frac{\nabla_\lambda l(y|\theta',\lambda)}{l(y|\theta',\lambda)} d\theta' \\
=&\int q(\theta'|y,\lambda) \nabla_\lambda \log l(y|\theta',\lambda) d\theta' \\
=&\mathbb{E}_{q(\theta'|y,\lambda)}[\nabla_\lambda \log l(y|\theta',\lambda)],
\end{aligned} \tag{5.29}$$

Finally, by plugging $y = g(\theta, \epsilon, \lambda)$ back into the equation, we obtain

$$\nabla_\lambda \log p(g(\theta,\epsilon,\lambda)|\lambda) = \mathbb{E}_{q(\theta'|g(\theta,\epsilon,\lambda),\lambda)}[\nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta',\lambda)]. \tag{5.30}$$

□

Theorem 5.1. *The gradient of the entropy $H(p(y|\lambda))$ w.r.t. the experimental condition λ satisfies*

$$\nabla_\lambda H(p(y|\lambda)) = -\mathbb{E}_{\pi_\theta(\theta)\pi_\epsilon(\epsilon)q(\theta'|g(\theta,\epsilon,\lambda),\lambda)}[\nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta',\lambda)], \tag{5.31}$$

where $q(\theta'|y,\lambda) \propto \pi_\theta(\theta') l(y|\theta',\lambda)$ is the posterior density of parameters given the observation sample y.

Proof. By the reparameterization trick, we have

$$\begin{aligned}
\nabla_\lambda H(p(y|\lambda)) &= -\nabla_\lambda \mathbb{E}_{\pi_\theta(\theta)\pi_\epsilon(\epsilon)}[\log p(g(\theta,\epsilon,\lambda)|\lambda)] \\
&= -\mathbb{E}_{\pi_\theta(\theta)\pi_\epsilon(\epsilon)}[\nabla_\lambda \log p(g(\theta,\epsilon,\lambda)|\lambda)].
\end{aligned} \tag{5.32}$$

Using Lemma 5.1, we can substitute the gradient of the logarithm of the marginal density in the above equation by the posterior expectation of the logarithm of the likelihood function. □

Corollary 5.1. *The gradient of the EIG $U(\lambda)$ w.r.t. the experimental condition λ satisfies*

$$\nabla_\lambda U(\lambda) = \mathbb{E}_{\pi_\theta(\theta)\pi_\epsilon(\epsilon)q(\theta'|g(\theta,\epsilon,\lambda),\lambda)}[\nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta,\lambda) - \nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta',\lambda)], \quad (5.33)$$

where $q(\theta'|y,\lambda) \propto \pi_\theta(\theta')l(y|\theta',\lambda)$ is the posterior density of parameters given the observation sample y.

Proof. Applying the reparameterization trick to the gradient of the negative conditional entropy term $\mathbb{E}_{\pi_{\theta(\theta)l(y|\theta,\lambda)}}[\log l(y|\theta,\lambda)]$ w.r.t. the experimental condition λ, we have

$$\begin{aligned}\nabla_\lambda \mathbb{E}_{\pi_{\theta(\theta)l(y|\theta,\lambda)}}[\log l(y|\theta,\lambda)] &= \mathbb{E}_{\pi_\theta(\theta)\pi_\epsilon(\epsilon)}[\nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta,\lambda)] \\ &= \mathbb{E}_{\pi_\theta(\theta)\pi_\epsilon(\epsilon)q(\theta'|g(\theta,\epsilon,\lambda),\lambda)}[\nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta,\lambda)].\end{aligned} \quad (5.34)$$

Combining the above equation with Theorem 5.1, we finally get Eq. (5.33). □

Theorem 5.2. *The expectation of $\widehat{U}^M_{srNMC}(\lambda)$ satisfies the following:*

1. *$\mathbb{E}[\widehat{U}^M_{srNMC}(\lambda)]$ is a lower bound on $U(\lambda)$ for any $M > 0$.*
2. *$\mathbb{E}[\widehat{U}^M_{srNMC}(\lambda)]$ is monotonically increasing in M, i.e., $\mathbb{E}[\widehat{U}^{M_1}_{srNMC}(\lambda)] \leq \mathbb{E}[\widehat{U}^{M_2}_{srNMC}(\lambda)]$ for $0 \leq M_1 \leq M_2$.*

Proof. To prove the first result in Theorem 5.2, we first note that the outer Monte Carlo terms are identically distributed. Thus,

$$\mathbb{E}[\widehat{U}^M_{srNMC}(\lambda)] = \mathbb{E}\left[\log \frac{l(y^{(1)}|\theta^{(1)},\lambda)}{\frac{1}{M}\sum_{j=1}^M l(y^{(1)}|\theta^{(j)},\lambda)}\right], \quad (5.35)$$

where $y^{(1)} = g(\theta^{(1)},\epsilon^{(1)},\lambda)$ and the expectation is taken over $p(y^{(1)}|\lambda)q(\theta^{(1)}|y^{(1)},\lambda)\prod_{j=2}^M \pi_\theta(\theta^{(j)})$. We proceed the rest as in [48]. We let $\delta = U(\lambda) - \mathbb{E}[\widehat{U}^M_{srNMC}(\lambda)]$, then

$$\begin{aligned}\delta &= \mathbb{E}\left[\log \frac{\frac{1}{M}\sum_{j=1}^M l(y^{(1)}|\theta^{(j)},\lambda)}{p(y^{(1)}|\lambda)}\right] \\ &= \mathbb{E}\left[\log \frac{\frac{1}{M}\sum_{j=1}^M q(\theta^{(j)}|y^{(1)},\lambda)\prod_{k\neq j}\pi_\theta(\theta^{(k)})}{\prod_{j=1}^M \pi_\theta(\theta^{(j)})}\right] \\ &= \mathbb{E}\left[\log \frac{P(\theta^{(1:M)}|y^{(1)})}{\prod_{j=1}^M \pi_\theta(\theta^{(j)})}\right],\end{aligned} \quad (5.36)$$

where $P(\theta^{(1:M)}|y^{(1)}) = \frac{1}{M}\sum_{j=1}^{M} q(\theta^{(j)}|y^{(1)},\lambda)\prod_{k\neq j}\pi_\theta(\theta^{(k)})$. Due to the permutation symmetry of the integrand over the labels $1,...,M$, the expectation keeps the same if it is instead taken over $p(y^{(1)}|\lambda)P(\theta^{(1:M)}|y^{(1)})$. Thus we have

$$\delta = \mathbb{E}\left[\mathrm{D}_{\mathrm{KL}}\left(P(\theta^{(1:M)}|y^{(1)})\|\prod_{j=1}^{M}\pi_\theta(\theta^{(j)})\right)\right] \geq 0, \tag{5.37}$$

where the expectation is taken over $p(y^{(1)}|\lambda)$.

To prove the second result, as in the former proof we let $\phi = \mathbb{E}[\widehat{U}^{M_2}_{srNMC}(\lambda)] - \mathbb{E}[\widehat{U}^{M_1}_{srNMC}(\lambda)]$. Then

$$\begin{aligned}\phi &= \mathbb{E}\left[\log\frac{\frac{1}{M_1}\sum_{j=1}^{M_1} l(y^{(1)}|\theta^{(j)},\lambda)}{\frac{1}{M_2}\sum_{j=1}^{M_2} l(y^{(1)}|\theta^{(j)},\lambda)}\right] \\ &= \mathbb{E}\left[\log\frac{Q(\theta^{(1:M_2)}|y^{(1)})}{P(\theta^{(1:M_2)}|y^{(1)})}\right],\end{aligned} \tag{5.38}$$

where the expectation is taken over $p(y^{(1)}|\lambda)q(\theta^{(1)}|y^{(1)},\lambda)\prod_{j=2}^{M_2}\pi_\theta(\theta^{(j)})$ and $Q(\theta^{(1:M_2)}|y^{(1)}) = \frac{1}{M_1}\sum_{j=1}^{M_1} q(\theta^{(j)}|y^{(1)},\lambda)\prod_{k\neq j}^{M_2}\pi_\theta(\theta^{(k)})$. Again, using the permutation symmetry, we can get the same expectation if the sampling distribution is taken as $p(y^{(1)}|\lambda)Q(\theta^{(1:M_2)}|y^{(1)})$ instead. Thus,

$$\phi = \mathbb{E}\left[\mathrm{D}_{\mathrm{KL}}\left(Q(\theta^{(1:M_2)}|y^{(1)})\|P(\theta^{(1:M_2)}|y^{(1)})\right)\right] \geq 0, \tag{5.39}$$

where the expectation is taken over $p(y^{(1)}|\lambda)$. □

Theorem 5.3. *If $l(g(\theta,\epsilon,\lambda)|\theta',\lambda)$ is bounded away from 0 and uniformly bounded from above (i.e., $C_1 \leq l(g(\theta,\epsilon,\lambda)|\theta',\lambda) \leq C_2$ a.s. for some positive constants C_1 and C_2), then the mean squared error of $\widehat{U}^{M}_{srNMC}(\lambda)$ converges to 0 at rate $O(1/M)$.*

Proof. For simplicity, we denote $f(\theta,\epsilon) = \log\frac{l(g(\theta,\epsilon,\lambda)|\theta,\lambda)}{p(g(\theta,\epsilon,\lambda)|\lambda)}$. Then the ground-truth EIG can be represented as $U(\lambda) = \mathbb{E}[f(\theta,\epsilon)]$. Using Minkowski's inequality, the mean squared error of $\widehat{U}^{M}_{srNMC}(\lambda)$ can be bounded by

$$\mathbb{E}[(U(\lambda) - \widehat{U}^{M}_{srNMC}(\lambda))^2] = \|U(\lambda) - \widehat{U}^{M}_{srNMC}(\lambda)\|_2^2 \leq U^2 + V^2 + 2UV \leq 2(U^2+V^2), \tag{5.40}$$

where the expectation is taken over $\prod_{i=1}^{M}\pi_\theta(\theta^{(i)})\pi_\epsilon(\epsilon^{(i)})$, $U = \left\|U(\lambda) - \frac{1}{M}\sum_{i=1}^{M} f(\theta^{(i)},\epsilon^{(i)})\right\|_2$ and $V = \left\|\frac{1}{M}\sum_{i=1}^{M} f(\theta^{(i)},\epsilon^{(i)}) - \widehat{U}^{M}_{srNMC}(\lambda)\right\|_2$. Since $l(g(\theta,\epsilon,\lambda)|\theta',\lambda)$ is uniformly bounded from below and above, we have $f \in L^2$. Also noting that U is the square of mean error of a Monte Carlo estimation, it is easy to get $U = O(1/\sqrt{M})$. Now we turn to bound V. Using the assumption that $l(g(\theta,\epsilon,\lambda)|\theta',\lambda) \geq C_1$ a.s., we have

$$\begin{aligned} V &= \Big\|\frac{1}{M}\sum_{i=1}^{M}\log\frac{\frac{1}{M}\sum_{j=1}^{M} l(y^{(i)}|\theta^{(j)},\lambda)}{p(y^{(i)}|\lambda)}\Big\|_2 \\ &\leq \frac{1}{M}\sum_{i=1}^{M}\Big\|\log\frac{\frac{1}{M}\sum_{j=1}^{M} l(y^{(i)}|\theta^{(j)},\lambda)}{p(y^{(i)}|\lambda)}\Big\|_2 \\ &\leq \frac{1}{C_1 M}\sum_{i=1}^{M}\Big\|\frac{1}{M}\sum_{j=1}^{M} l(y^{(i)}|\theta^{(j)},\lambda) - p(y^{(i)}|\lambda)\Big\|_2, \end{aligned} \tag{5.41}$$

where $y^{(i)} = g(\theta^{(i)},\epsilon^{(i)},\lambda)$. For each term of the above equation, by Minkowski's inequality we have

$$\begin{aligned} &\Big\|\frac{1}{M}\sum_{j=1}^{M} l(y^{(i)}|\theta^{(j)},\lambda) - p(y^{(i)}|\lambda)\Big\|_2 \\ \leq &\frac{1}{M}\Big\|l(y^{(i)}|\theta^{(i)},\lambda) - l(y^{(i)}|\theta'^{(i)},\lambda)\Big\|_2 + \Big\|\frac{1}{M}\sum_{j\neq i}^{M} l(y^{(i)}|\theta^{(j)},\lambda) + \frac{1}{M}l(y^{(i)}|\theta'^{(i)},\lambda) - p(y^{(i)}|\lambda)\Big\|_2, \end{aligned} \tag{5.42}$$

where $\theta'^{(i)} \sim \pi_\theta(\theta)$. Using the assumption that $l(g(\theta,\epsilon,\lambda)|\theta',\lambda) \leq C_2$, the first term of above equation can be bounded by $2C_2/M$. The square of the second equation can be bounded as

$$\begin{aligned} &\Big\|\frac{1}{M}\sum_{j\neq i}^{M} l(y^{(i)}|\theta^{(j)},\lambda) + \frac{1}{M}l(y^{(i)}|\theta'^{(i)},\lambda) - p(y^{(i)}|\lambda)\Big\|_2^2 \\ =&\mathbb{E}\Big[\mathbb{E}\Big[\Big(\frac{1}{M}\sum_{j\neq i}^{M} l(y^{(i)}|\theta^{(j)},\lambda) + \frac{1}{M}l(y^{(i)}|\theta'^{(i)},\lambda) - p(y^{(i)}|\lambda)\Big)^2\Big|y^{(i)}\Big]\Big] \\ =&\mathbb{E}\Big[\mathrm{Var}\Big[\frac{1}{M}\sum_{j\neq i}^{M} l(y^{(i)}|\theta^{(j)},\lambda) + \frac{1}{M}l(y^{(i)}|\theta'^{(i)},\lambda)\Big|y^{(i)}\Big]\Big] \\ =&\frac{1}{M}\mathbb{E}\Big[\mathrm{Var}\Big[l(y^{(i)}|\theta'^{(i)},\lambda)\Big|y^{(i)}\Big]\Big] \end{aligned} \tag{5.43}$$

where the first equality is obtained by the tower property of conditional expectation. It can be further bounded $O(1/M)$ noting that $l(y^{(i)}|\theta'^{(i)},\lambda)$ is uniformly bounded from below and

above. Therefore, we have $V = O(1/\sqrt{M})$ as well. Finally, using the obtained bounds of U and V we get the mean squared error of $\widehat{U}^M_{srNMC}(\lambda)$

$$\mathbb{E}[(U(\lambda) - \widehat{U}^M_{srNMC}(\lambda))^2] \leq 2(U^2 + V^2) = O(1/M). \tag{5.44}$$

□

Theorem 5.4. *For any C satisfying $0 \leq C \leq U(\lambda)/2$, if $M \leq \exp(U(\lambda)/2)$, we have*

$$U(\lambda) - \widehat{U}^M_{srNMC}(\lambda) > C. \tag{5.45}$$

Proof. We first show that $\widehat{U}^M_{srNMC}(\lambda)$ does not exceed $\log M$. By the definition of srNMC, we have

$$\begin{aligned} \widehat{U}^M_{srNMC}(\lambda) &= \frac{1}{M}\sum_{i=1}^{M} \log \frac{l(y^{(i)}|\theta^{(i)}, \lambda)}{\frac{1}{M}\sum_{j=1}^{M} l(y^{(i)}|\theta^{(j)}, \lambda)} \\ &\leq \frac{1}{M}\sum_{i=1}^{M} \log \frac{l(y^{(i)}|\theta^{(i)}, \lambda)}{\frac{1}{M} l(y^{(i)}|\theta^{(i)}, \lambda)} \\ &\leq \frac{1}{M}\sum_{i=1}^{M} \log M = \log M, \end{aligned} \tag{5.46}$$

where $y^{(i)} = g(\theta^{(i)}, \epsilon^{(i)}, \lambda)$. Using this inequality and $M \leq \exp(U(\lambda)/2)$, it is easy to get

$$\begin{aligned} U(\lambda) - \widehat{U}^M_{srNMC}(\lambda) &\geq U(\lambda) - \log M \\ &\geq U(\lambda)/2 \geq C. \end{aligned} \tag{5.47}$$

□

5.7 Further details of experiments

5.7.1 EIG Gradient Estimation Accuracy

We assume 3 design variables (i.e. $\lambda = (\lambda_1, \lambda_2, \lambda_3)$) for this test. The 20 independent designs are uniformly drawn from $[-1, 1]^3$. To estimate the biases associated with the methods

under investigation, we perform 100 independent trials. Table 5.1 summarizes the number of samples used to estimate the gradient for a single design.

Method	Number of samples
BEEG-AP	$100 \times 100(M)$
UEEG-MCMC	$100 \times 100(L)$
PCE	$100 \times 100(M) \times (100(N) + 1)$

Table 5.1: Method and number of samples for the test of EIG gradient estimation accuracy.

5.7.2 A Toy Algebraic Model

In this test, we allocate a simulation budget of 2×10^4 for optimizing the design variables for each method. For UEEG-MCMC, we use slice sampling [111] with a thinning factor of 2 to draw 10 samples for the posterior sampling in Eq. (5.14). For ACE, we use a conditional Gaussian posterior inference network $q_\phi(\cdot|y) = \text{normpdf}(\cdot|\mu_\phi(y), e^{2\sigma_\phi(y)})$, where μ_ϕ and σ_ϕ are the two outputs of a two-layer fully connected network with 50 hidden units and ReLU activation functions. For GradBED, we use a two-layer fully connected network T_ψ with 50 hidden units and ReLU activation functions. The number of samples used to estimate a single gradient for each method is given by Table 5.2.

Method	Number of samples
BEEG-AP	$100(M)$
UEEG-MCMC	$10(M) \times$ No. samples from slice sampling
ACE	$10(M) \times (10(N) + 1)$
PCE	$10(M) \times (10(N) + 1)$
GradBED	$100(M)$

Table 5.2: Method and number of samples for the toy model.

To validate the quality of designs, we use NMC with 10,000 samples for each estimate of EIG in Fig. 5.3. The estimated posterior entropy in Fig. 5.4 is obtained as follows. In our experimental setup, each method yields 20 final designs. For each of these designs, we conduct simulations resulting in 500 observed data points derived from the marginal likelihood, thus forming 500 *guess* posteriors. Subsequently, kernel density estimation (KDE) is applied to these posteriors, utilizing 100 posterior samples for each, to approximate the entropies of these posteriors. The posterior entropy for each method is then computed by averaging these approximated entropies.

5.7.3 PK Model

In this application, we allocate a simulation budget of 2×10^6 for optimizing the design variables for each method. For UEEG-MCMC, we use an adaptive Metropolis-Hastings (MH) method with a thinning factor of 95 to draw only one sample for the posterior sampling in Eq. (5.14). For ACE, we utilize the Mixture Density Network [23] as our chosen posterior inference network, with 3 hidden layers, 50 hidden units in each layer and ReLU activation function. For GradBED, we follow the network settings in [81]. Specifically, we use a one-layer connected network T_ψ consisting of 300 hidden units and ReLU activation functions. The number of samples used to estimate a single gradient for each method is given by Table 5.3.

Method	Number of samples
BEEG-AP	$100(M)$
UEEG-MCMC	$1(M) \times 100$(approximated cost of the adaptive MH)
ACE	$10(M) \times (10(N) + 1)$
PCE	$10(M) \times (10(N) + 1)$
GradBED	$100(M)$

Table 5.3: Method and number of samples for PK model and STAT5 model.

In the validation experiments, we use NMC with 10,000 samples for each estimate of EIG in Fig. 5.5. In Fig. 5.6, the posterior entropy for each method is estimated via 1000 independent trials (see Appendix 5.7.2 for the procedure). For each trial, we use KDE to estimate the *guess* posterior entropy with 1000 samples.

5.7.4 STAT5 Model

The dynamics of STAT5 populations can be described by four coupled ODEs

$$\begin{aligned}
\dot{x}_1 &= -k_1 x_1 \,\mathrm{EpoR}_A(t) + k_2 x_3(t-\tau) \\
\dot{x}_2 &= -x_2^2 + k_1 x_1 \,\mathrm{EpoR}_A(t) \\
\dot{x}_3 &= -k_2 x_3 + x_2^2 \\
\dot{x}_4 &= -k_2 x_3(t-\tau) + k_2 x_3.
\end{aligned} \tag{5.48}$$

The variables in the model are defined as follows. x_1 represents unphosphorylated STAT5. x_2 and x_3 represent tyrosine phosphorylated monomeric STAT5 and tyrosine phosphorylated dimeric STAT5 respectively. x_4 is the nuclear STAT5. $\mathrm{EpoR}_A(t)$ describes the erythropoietin receptor activity that determines the STAT5 response.

We assume that $x_1(0) = 3.71$ is the only non-zero initial state of the ODES. To facilitate the solvability of the ordinary differential equations (ODEs), as in [126, 12], we apply linear interpolation to synthesize the function $\mathrm{EpoR}_A(t)$ with the original data in [164], and use a delay chain of length N to approximate the delayed term $x_3(t-\tau)$,

$$\begin{aligned}
\dot{q}_1 &= \frac{N}{\tau}\left(\mathrm{in}(t) - q_1\right) \\
\dot{q}_2 &= \frac{N}{\tau}\left(q_1 - q_2\right) \\
&\cdots \\
\dot{q}_{N-1} &= \frac{N}{\tau}\left(q_{N-2} - q_{N-1}\right) \\
\mathrm{out} &= \frac{N}{\tau}\left(q_{N-1} - \mathrm{out}(t)\right),
\end{aligned} \tag{5.49}$$

where $N = 8$, $\text{in}(t) = x_3(t)$ and $\text{out}(t) = x_3(t - \tau)$.

To build the sampling path that supports backpropagation through ODE solutions, we utilize the package `torchdiffeq` [30] to solve the ODEs with 3/8-Runge-Kutta method [27]. Linear interpolation is then applied to get the observations at any measurement times.

In this application, we allocate a simulation budget of 5×10^5 for optimizing the design variables for each method. The settings for the methods involved are consistent with those employed for the PK model, and the sample size for each method is also provided in Table 5.3. In the validation experiments, the posterior entropy for each method shown in Fig. 5.7 and Fig. 5.8 is estimated via 100 independent trials (see Appendix 5.7.2 for the procedure). For each trial, KDE is employed with 1000 posterior samples.

5.8 Conclusion

In this work we have proposed two approaches, UEEG-MCMC and BEEG-AP, to Bayesian experimental design based on EIG gradient estimation. We use MCMC sampling techniques to build the gradient estimation for UEEG-MCMC, while BEEG-AP approximates the EIG gradients with atomic priors. Both of them are straightforward to implement and have demonstrated improved performance compared to bench-marking methods.

Our theoretical analysis aligns well with the numerical results. Specifically, BEEG-AP exhibits superior simulation efficiency when dealing with problems that have small ground-truth EIG values, making it a favorable option in such cases. On the other hand, UEEG-MCMC shows robustness across various EIG levels, making it suitable for a broader range of experimental scenarios. We believe the work provides researchers and practitioners with promising tools and guidance to optimize their experiments and make informed decisions

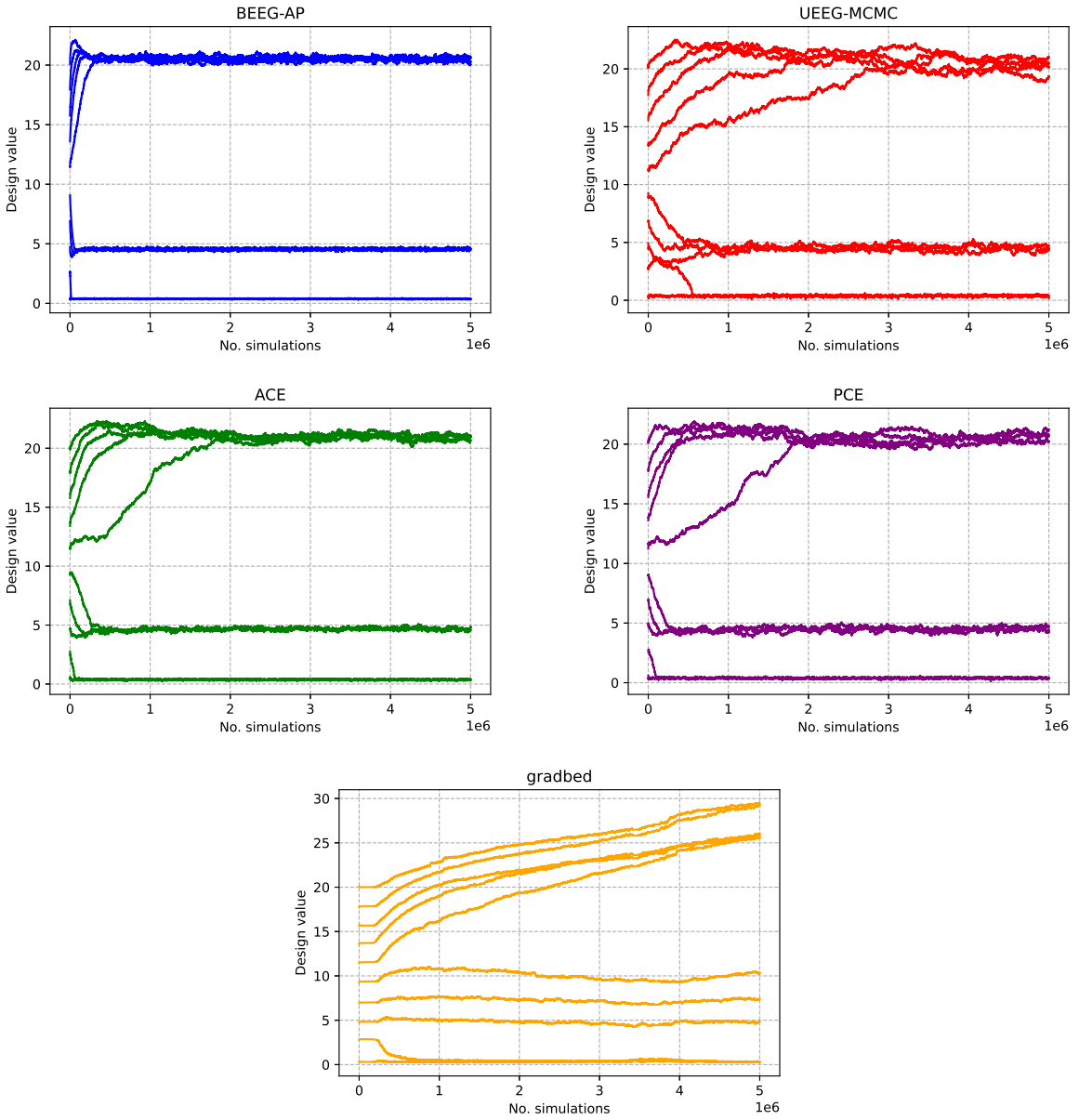


Figure 5.9: Convergence of the individual design dimensions for PK model with multiplicative noise $\mathcal{N}(0, 0.01)$ and additive noise $\mathcal{N}(0, 0.1)$.

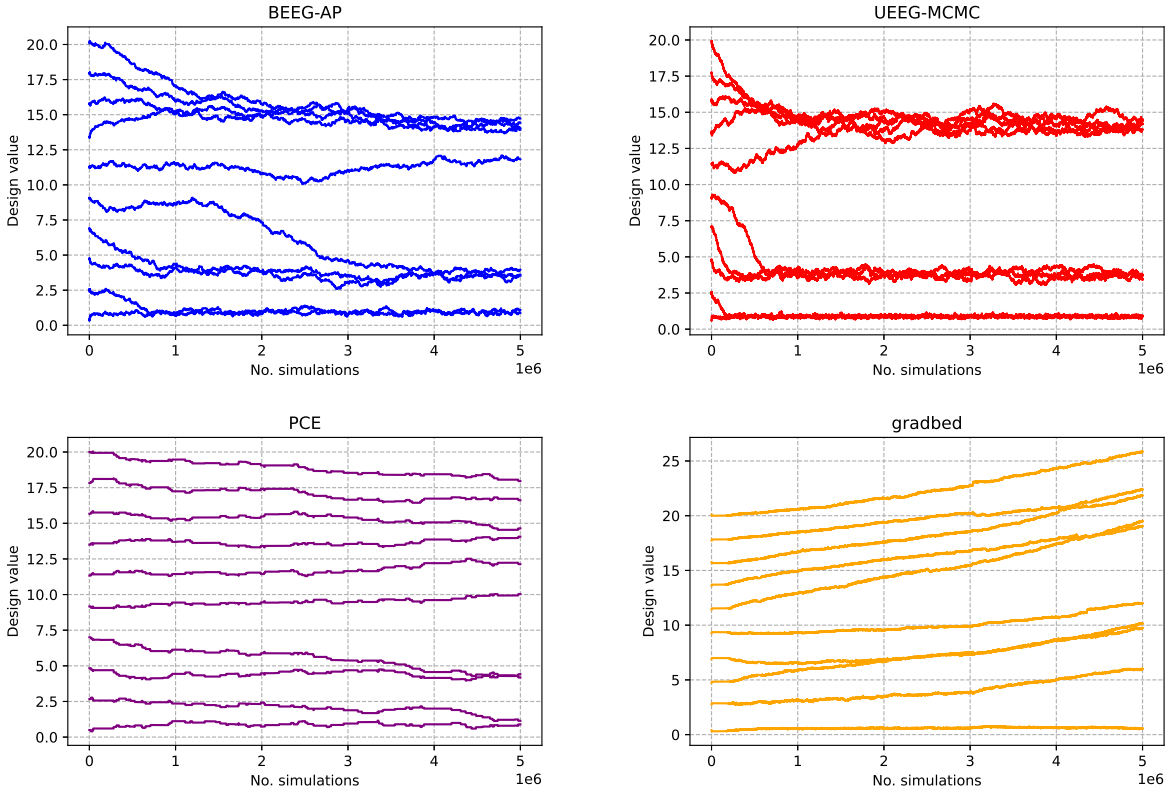


Figure 5.10: Convergence of the individual design dimensions for PK model with additive noise $\mathcal{N}(0, 0.001)$.

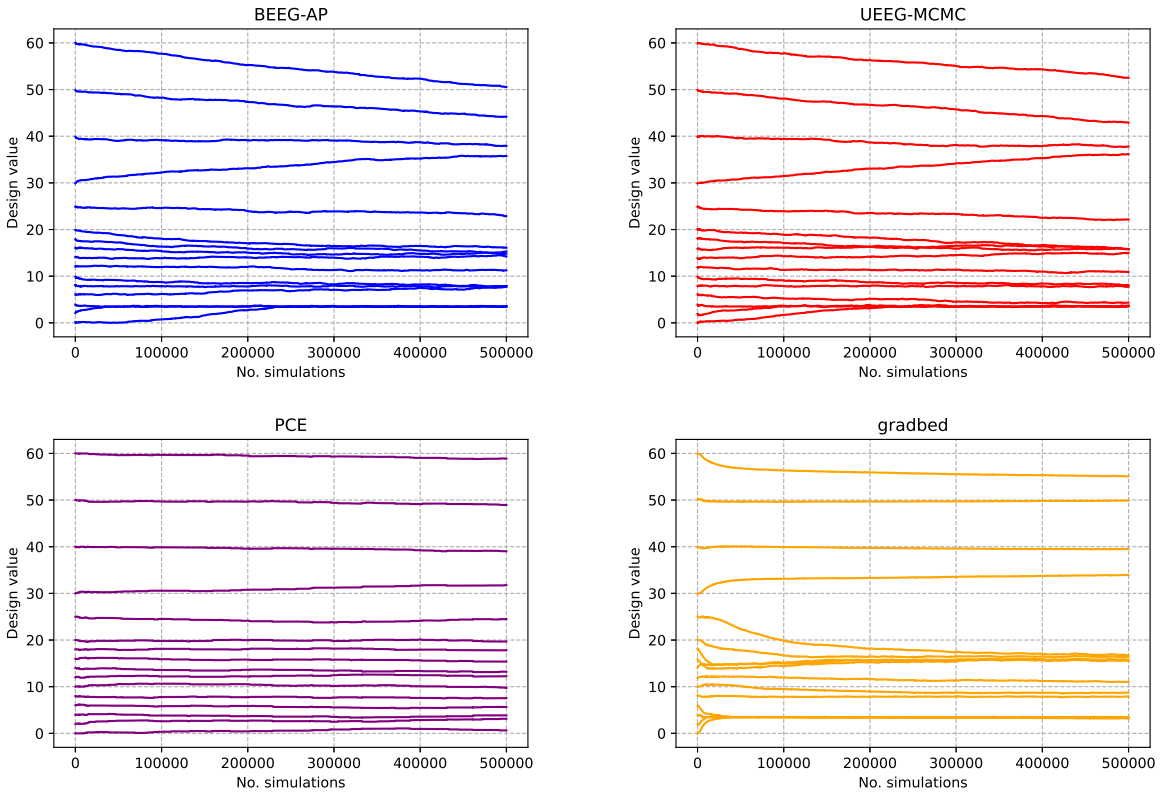


Figure 5.11: Convergence of the individual design dimensions for STAT5 model with additive noise $\mathcal{N}(0, 0.01)$.

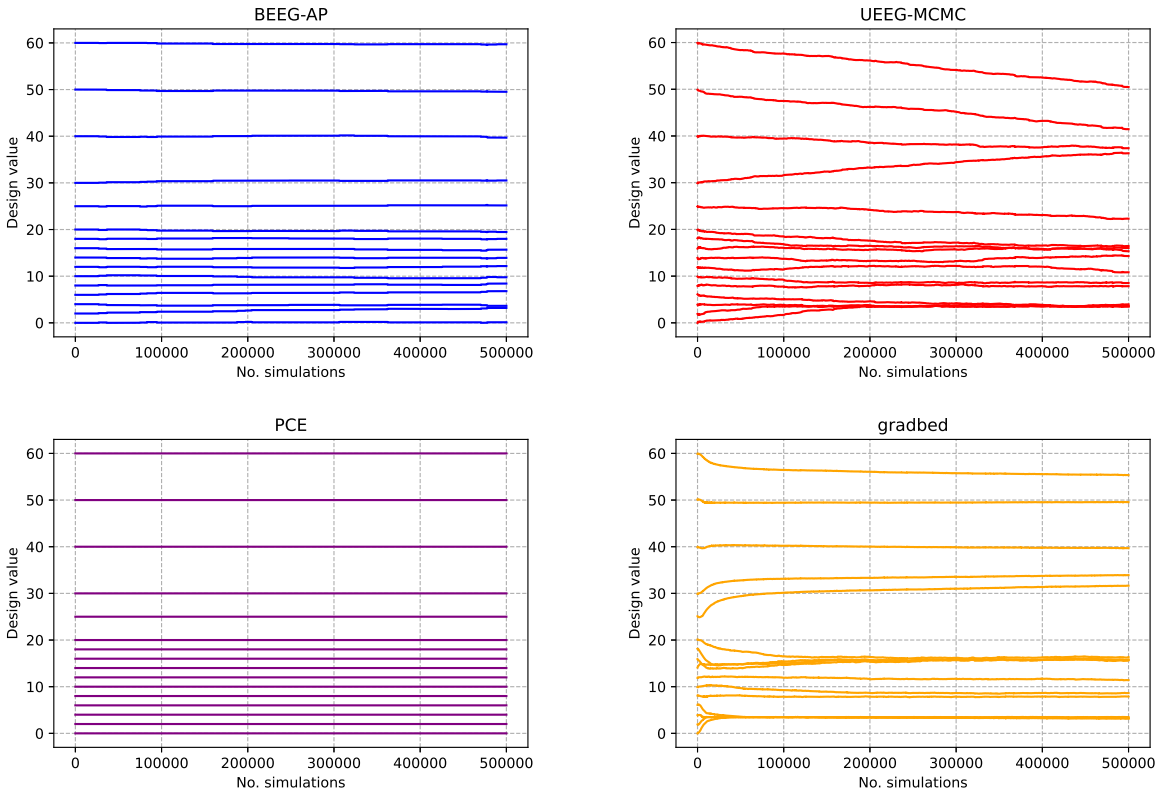


Figure 5.12: Convergence of the individual design dimensions for STAT5 model with additive noise $\mathcal{N}(0, 0.001)$.

across different domains.

Chapter Six

Discussion

6.1 Contributions

In conclusion, this thesis represents a comprehensive exploration of entropy estimation and optimization, particularly within the context of Bayesian Experimental Design (BED). It has successfully navigated through a series of complex challenges and theoretical intricacies, contributing significantly to the field of information theory and experimental design.

The first major contribution of this thesis is the extensive survey of existing and emerging techniques in entropy estimation and optimization, with a special focus on their application in BED. This survey not only provided a deep understanding of traditional methods like plug-in and k-NN estimators but also shed light on advanced techniques such as variational estimators and entropy gradient estimation methods. This exploration has opened up new perspectives in BED, particularly in understanding how its objectives are related to entropy optimization.

The second significant contribution is the development of a novel method for high-dimensional entropy estimation. By combining a k-NN based estimator with a normalizing

flow-based mapping technique, the thesis has addressed the challenge of reducing bias for entropy estimation in high-dimensional spaces. This transform-based method marks a substantial advancement in entropy estimation, providing a more accurate and efficient tool for dealing with entropy estimation problems for complex and high-dimensional distributions.

The third key contribution is the advancement of BED through the introduction of entropy gradient estimation techniques. The development of UEEG-MCMC and BEEG-AP methods for estimating the expected information gain (EIG) gradient is a testament to the innovative approach of this research. These methods have been thoroughly analyzed both theoretically and empirically, proving their effectiveness and limitations in various scenarios.

Looking ahead, the thesis proposes two promising directions for future research. These proposed paths not only build upon the findings and methodologies developed in the thesis but also aim to address emerging challenges and opportunities in the field. The next section of this chapter will delve into these research trajectories.

6.2 Future directions

6.2.1 Global optimization for Bayesian experimental design via Bayesian optimization with local search

In Chapter 5, we introduced the UEEG-MCMC and BEEG-AP methodologies for the design of Bayesian-optimal experiments, which have shown promise in efficiently navigating the design space. Despite their effectiveness, these gradient-based methods are susceptible to convergence on local optima, a limitation that is well-documented within the domain. In contrast, Bayesian Optimization (BO) has been a preferred strategy for global optimization in various BED frameworks. However, BO's scalability is challenged by high-dimensional

design spaces, where its performance is significantly impeded by the increase in search space, as demonstrated by [48].

Addressing this gap, the development of scalable global optimization strategies for BED that can operate effectively within high-dimensional contexts stands as an imperative research trajectory. Our prospective solution draws inspiration from the Bayesian Optimization with Local Search (BOwLS) approach outlined by Gao et al. (2020), which cleverly amalgamates the strengths of local and global search strategies. Leveraging this concept, we aim to architect a hybrid method that synergizes the robust local search capabilities of our proposed gradient-based methods with the global search prowess of BO.

As we embark on this venture, it is pertinent to acknowledge the innovative groundwork laid by BOwLS. This approach provides a compelling framework for our proposed methodology, which we believe can inherit and expand upon the efficacy of BOwLS in tackling the challenges presented by high-dimensional design problems. Our preliminary findings are encouraging, suggesting that such an integrated approach could indeed yield a more versatile and powerful tool for the BED community.

Bayesian Optimization with Local Search

Bayesian Optimization with Local Search (BOwLS) is designed to identify the global minimum of a given objective function, denoted as $f(x)$, by ingeniously combining a Multi-search (MS) algorithm [101] with traditional BO. The core philosophy of BOwLS is straightforward: it applies BO to a modified function which preserves the same global minima as the original function $f(x)$, but is redefined to incorporate local search outcomes. Let $\mathcal{L}$ denote the local search solver, which is described by the equation:

$$x^* = \mathcal{L}(f(\cdot), x), \tag{6.1}$$

where x is the initial search point, and x^* represents the resulting local minimum point. The solver $\mathcal{L}$ can encompass any local optimization technique, with the stipulation that it delivers a unique local minimum for any given starting point x. Building on the solver $\mathcal{L}$ and the original function f, a novel function $F_{\mathcal{L}}$ is defined as:

$$y = F_{\mathcal{L}}(x) = f(x^*), \tag{6.2}$$

where x^* arises from applying the local solver $\mathcal{L}$ to the function f with the initial point x. In essence, the function $F_{\mathcal{L}}$ accepts an initial point x and outputs the value of f at the local minimum found by $\mathcal{L}$. This ensures that $F_{\mathcal{L}}$ is a properly defined function with identical global minima to the function f. Fig. 6.1 provides a graphic depiction of the LS-defined function and its Gaussian Process (GP) approximation. Subsequent to defining $F_{\mathcal{L}}$, the established BO algorithm is employed to optimize this new function, with the global minimum of $F_{\mathcal{L}}$ identified by BO accepted as the global minimum of the original function f. The above procedure of this method is summarized in Algorithm 3. It is worth noting that BOwLS functions essentially as a MS algorithm that utilizes the BO experimental design criterion to determine subsequent starting points for the local search.

BED via BOwLS

Recall that in Bayesian Experimental Design (BED), our objective is to maximize the expected information gain (EIG) over the design space. This can be mathematically represented as:

$$U(\lambda) = \mathbb{E}_{\pi_{\theta(\theta)l(y|\theta,\lambda)}}[\log l(y|\theta,\lambda)] - \mathbb{E}_{p(y|\lambda)}[\log p(y|\lambda)]. \tag{6.3}$$

where $U(\lambda)$ is the utility function in terms of the design λ. Incorporating the BOwLS framework, we can redefine our objective function as the negation of the utility, thus aiming to find the design that minimizes this new function:

$$f(\lambda) = -U(\lambda), \tag{6.4}$$

Algorithm 3 The BOwLS algorithm

Require: N (total number of calls to the local solver), the initial dataset of input parameters $\{\mathbf{x}_n\}_{1\leq n\leq N_0}$ (randomly chosen from the domain where the optimization is conducted).

let $D_0 = \emptyset$;
for $n = 1{:}N_0$ **do**
    solve $[y^*, \mathbf{x}^*] = \mathcal{L}(f(\mathbf{x}), \mathbf{x}_n)$;
    let $y_n = y^*$;
    augment data $D_n = D_{n-1} \cup \{(\mathbf{x}_n, y_n)\}$
end for
construct a GP model from D_{N_0}, denoted as $\hat{f}_{N_0}$;
$n = N_0$;
while $n \leq N$ **do**
    $\mathbf{x}_{n+1} = \arg\max \alpha(\mathbf{x}; \hat{f}_n)$
    solve $[y^*, \mathbf{x}^*] = \mathcal{L}(f(\mathbf{x}), \mathbf{x}_{n+1})$;
    let $y_{n+1} = y^*$;
    augment data $D_{n+1} = D_n \cup \{(\mathbf{x}_{n+1}, y_{n+1})\}$;
    update GP model obtaining $\hat{f}_{n+1}$;
    $n = n + 1$;
end while
return $y_{\min} = \min\{y_i\}_{i=1}^{N}$

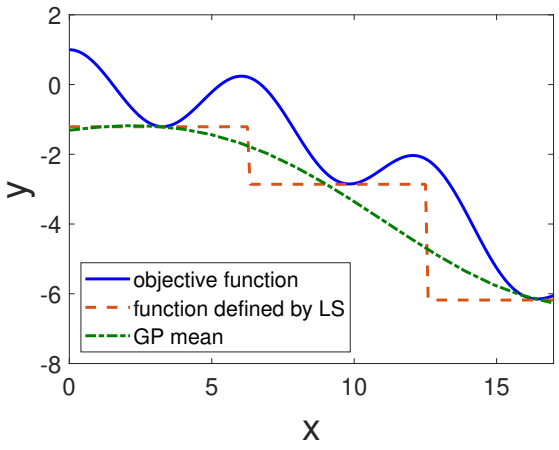


Figure 6.1: A schematic illustration of the BOwLS algorithm (reprinted from [55]): the solid line is the original objective function, the dashed line is the function defined by LS, and the dashed-dotted line is the GP regression of the LS defined function.

which is amenable to gradient-based optimization methods. The local solver $\mathcal{L}$ in the context of BOwLS is constituted by the UEEG-MCMC or BEEG-AP algorithms combined with a gradient ascent approach. Upon each iteration, starting from an initial design λ, the local solver seeks to find a local optimum λ^*. The value of the LS-defined function is then computed as:

$$y = F_{\mathcal{L}}(\lambda) = -U(\lambda^*), \tag{6.5}$$

where the function $F_{\mathcal{L}}$ encapsulates the process of finding the local minimum of our objective function. Nested Monte Carlo (NMC) methods or the entropy estimators detailed in Chapter 3 can be used to evaluate this function.

We apply the proposed method to the illustrative algebraic model as described in [74], which is designed to simulate two dependent observations derived from a nonlinear mapping and noise components:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \theta^3\lambda_1^2 + \theta\exp(-|0.2-\lambda_1|) \\ \theta^3\lambda_2^2 + \theta\exp(-|0.2-\lambda_2|) \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix}, \tag{6.6}$$

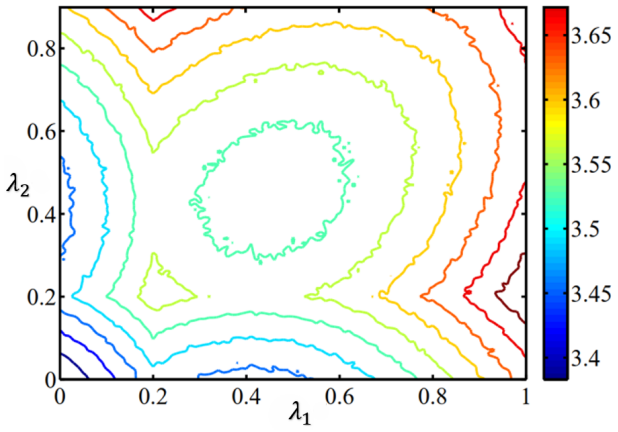
where θ is the parameter of interest, λ_i represents the design variables, and $\epsilon_i \sim \mathcal{N}(0, (0.01)^2)$ accounts for independent observation errors. The parameter θ follows a uniform prior distribution, $\theta \sim \text{Unif}(0, 1)$, and the design space for λ_1 and λ_2 is constrained to $[0, 1]$ and $[0, 0.85]$, respectively. In this numerical example, we contrast the performance of BOwLS optimization against its constituent methods—BO and gradient-based optimization (implemented via BEEG-AP)—across 20 independent trials for each.The comparative results are illustrated in Fig. 6.2. From the figures, we can see that designs obtained by BED via BOwLS are more concentrated on the global optimum compared to those obtained by BO and BEEG-AP. Our preliminary findings thus suggest that the integrated approach exhibits an increased precision in finding a glocal optimal design.

6.2.2 Bayesian experimental design for implicit models using entropy gradient estimation

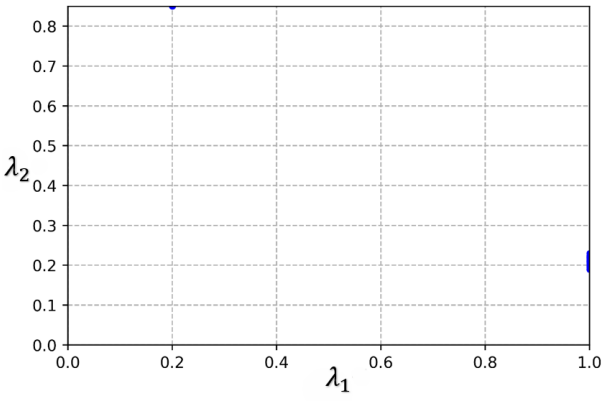
In the context of implicit models, unique challenges arise in Bayesian experimental design (BED) due to the absence of closed-form expressions for both the marginal likelihood function $p(y|\lambda)$ and the likelihood function $l(y|\theta, \lambda)$. Diverging from the predominant focus in existing literature on estimating the expected information gain (EIG), this research, as initiated in Chapter 5, aims to develop techniques for estimating the gradient of EIG when implicit models are considered. Our approach begins with the introduction of a simplified expression for the EIG gradient, based on the score function identity outlined below.

Lemma 6.1 (Score Function Identity)**.** *Consider a parametric statistical model defined by a probability density function $f(x; \xi)$, where ξ denotes the model parameter. The score function $S(\xi)$ is the gradient of the log-likelihood function with respect to ξ (or its derivative in one-dimensional cases), expressed as:*

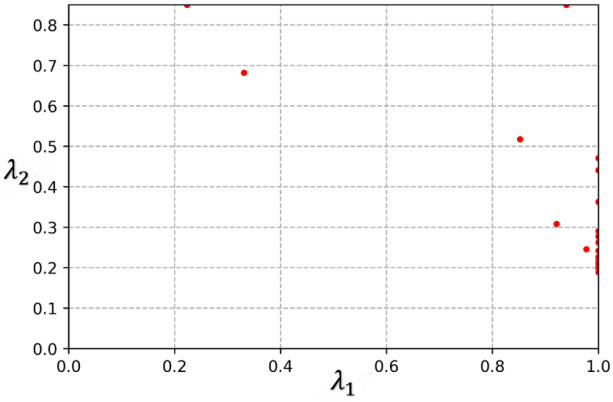
$$S(x; \xi) = \nabla_\xi \log f(x; \xi). \tag{6.7}$$



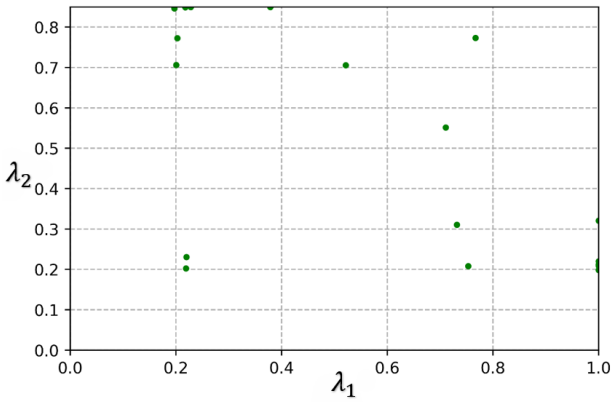
(a) The estimated EIGs throughout the entire design space, reprinted from [74].



(b) The final designs for BED via BOwLS.



(c) The final designs for BED via BO.



(d) The final designs for BED via BEEG-AP

Figure 6.2: The design results for the illustrative algebraic model.

The score function identity asserts that the expected value of $S(x;\xi)$ *under the model is zero:*

$$\mathbb{E}_{f(x;\xi)}[S(x;\xi)] = \mathbb{E}_{f(x;\xi)}\left[\nabla_\xi \log f(x;\xi)\right] = 0. \tag{6.8}$$

Proof. By simple calculation, we have

$$\begin{aligned}
\mathbb{E}_{f(x;\xi)}\left[\nabla_\xi \log f(x;\xi)\right] &= \mathbb{E}_{f(x;\xi)}\left[\frac{\nabla_\xi f(x;\xi)}{f(x;\xi)}\right] \\
&= \int f(x;\xi)\frac{\nabla_\xi f(x;\xi)}{f(x;\xi)}dx \\
&= \int \nabla_\xi f(x;\xi)dx \\
&= \nabla_\xi \int f(x;\xi)dx \\
&= \nabla_\xi 1 = 0
\end{aligned} \tag{6.9}$$

□

Building on the foundation laid by Lemma 6.1, we can rewrite the gradient of the EIG as detailed in Theorem 6.1. This theorem, which evolves from the notations established in Section 5.2, explicates the relationship between the gradient of EIG and the score functions $\nabla_y \log l(y|\theta,\lambda)$ and $\nabla_y \log p(y|\lambda)$.

Theorem 6.1. *Leveraging the established notations, the gradient of the EIG, denoted as $U(\lambda)$, with respect to parameter λ is given by:*

$$\nabla_\lambda U(\lambda) = \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}\left[\left(\nabla_y \log l(y|\theta,\lambda) - \nabla_y \log p(y|\lambda)\right)|_{y=g(\theta,\epsilon,\lambda)} \cdot \nabla_\lambda g(\theta,\epsilon,\lambda)\right]. \tag{6.10}$$

Proof. By the reparameterization trick, we have

$$\begin{aligned}
\nabla_\lambda U(\lambda) =& \nabla_\lambda \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}[\log l(g(\theta,\epsilon,\lambda)|\theta,\lambda)] - \nabla_\lambda \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}\left[\log p(g(\theta,\epsilon,\lambda)|\lambda)\right] \\
=& \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}\left[\nabla_\lambda \log l(g(\theta,\epsilon,\lambda)|\theta,\lambda)\right] - \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}\left[\nabla_\lambda \log p(g(\theta,\epsilon,\lambda)|\lambda)\right] \\
=& \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}\left[\left(\nabla_y \log l(y|\theta,\lambda) - \nabla_y \log p(y|\lambda)\right)|_{y=g(\theta,\epsilon,\lambda)} \cdot \nabla_\lambda g(\theta,\epsilon,\lambda)\right] \\
&+ \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}\left[\left(\nabla_\lambda \log l(y|\theta,\lambda) - \nabla_\lambda \log p(y|\lambda)\right)|_{y=g(\theta,\epsilon,\lambda)}\right] \\
=& \mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}\left[\left(\nabla_y \log l(y|\theta,\lambda) - \nabla_y \log p(y|\lambda)\right)|_{y=g(\theta,\epsilon,\lambda)} \cdot \nabla_\lambda g(\theta,\epsilon,\lambda)\right].
\end{aligned} \tag{6.11}$$

The last equality is due to the following identity:

$$\mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}\left[\left(\nabla_\lambda \log l(y|\theta,\lambda) - \nabla_\lambda \log p(y|\lambda)\right)|_{y=g(\theta,\epsilon,\lambda)}\right] = 0, \tag{6.12}$$

which is derived as follows:

$$\begin{aligned}
&\mathbb{E}_{\pi_{\theta(\theta)\pi_\epsilon(\epsilon)}}\left[\left(\nabla_\lambda \log l(y|\theta,\lambda) - \nabla_\lambda \log p(y|\lambda)\right)|_{y=g(\theta,\epsilon,\lambda)}\right] \\
=&\mathbb{E}_{\pi_{\theta(\theta)l(y|\theta,\lambda)}}\left[\nabla_\lambda \log l(y|\theta,\lambda) - \nabla_\lambda \log p(y|\lambda)\right] \\
=&\mathbb{E}_{\pi_{\theta(\theta)l(y|\theta,\lambda)}}\left[\nabla_\lambda \log l(y|\theta,\lambda) - \nabla_\lambda \log p(y|\lambda)\right] \\
=&\mathbb{E}_{\pi_{\theta(\theta)}\mathbb{E}_{l(y|\theta,\lambda)}}\left[\nabla_\lambda \log l(y|\theta,\lambda)\right] - \mathbb{E}_{p(y|\lambda)}\left[\nabla_\lambda \log p(y|\lambda)\right] \\
=&0,
\end{aligned} \tag{6.13}$$

where the last equality is obtained by applying the score function identity to $l(y|\theta,\lambda)$ and $p(y|\lambda)$ respectively. □

Estimating the EIG Gradient for implicit models

In estimating the EIG gradient for implicit models, Monte Carlo averaging can be employed to approximate the expectation in Eq. (6.10). Given samples $\theta^{(i)}i = 1^M$ and $\epsilon^{(i)}i = 1^M$ drawn from $\pi_\theta(\theta)\pi_\epsilon(\epsilon)$, the estimation proceeds as follows:

$$\nabla_\lambda U(\lambda) \approx \frac{1}{M}\sum_{i=1}^{M}\left(\nabla_y \log l(y^{(i)}|\theta^{(i)},\lambda) - \nabla_y \log p(y^{(i)}|\lambda)\right)\cdot \nabla_\lambda g(\theta^{(i)},\epsilon^{(i)},\lambda), \tag{6.14}$$

where $y^{(i)} = g(\theta^{(i)},\epsilon^{(i)},\lambda)$.

While the term $\nabla_\lambda g(\theta,\epsilon,\lambda)$ is readily accessible through modern automatic differentiation frameworks, directly calculating the score functions $\nabla_y \log l(y^{(i)}|\theta^{(i)},\lambda)$ and $\nabla_y \log p(y^{(i)}|\lambda)$ is more challenging. This issue can be addressed using an effective score function estimator, such as the Stein gradient estimator (refer to Section 2.3.2), utilizing the realizations of y from $l(y|\theta^{(i)},\lambda)$ and $p(y|\lambda)$ respectively. By obtaining approximations $\widehat{s_{\mathcal{L}}}^{(i)}$ for $\nabla_y \log l(y^{(i)}|\theta^{(i)},\lambda)$ and $\widehat{s_{\mathcal{M}}}^{(i)}$ for $\nabla_y \log p(y^{(i)}|\lambda)$, we can estimate the EIG gradient as follows:

$$\widehat{\nabla_\lambda U(\lambda)} \approx \frac{1}{M}\sum_{i=1}^{M}\left(\widehat{s_{\mathcal{L}}}^{(i)} - \widehat{s_{\mathcal{M}}}^{(i)}\right)\cdot \nabla_\lambda g(\theta^{(i)},\epsilon^{(i)},\lambda), \tag{6.15}$$

Here, $\widehat{s_{\mathcal{L}}}^{(i)}$ is computed using samples $y_{\mathcal{L}}^{(i,j)} = g(\theta^{(i)}, \epsilon^{(i,j)}, \lambda)$ for $j = 0, ..., N$, with $\epsilon^{(i,0)} = \epsilon^{(i)}$. Similarly, $\widehat{s_{\mathcal{M}}}^{(i)}$ employs samples $y_{\mathcal{M}}^{(i,j)} = g(\theta^{(i,j)}, \epsilon^{(i,j)}, \lambda)$ for $j = 0, ..., N$, with $\theta^{(i,0)} = \theta^{(i)}$ and $\epsilon^{(i,0)} = \epsilon^{(i)}$.

Joint design for summary statistics and design variables

In revisiting the primary aim of Bayesian experimental design (BED) in this thesis, which is to improve parameter inference, we encounter a unique challenge with implicit models: the unavailability of a closed-form likelihood function. This necessitates reliance on likelihood-free inference techniques. Prominent among these are Approximate Bayesian Computation (ABC) methods [15, 154] and more recent developments in conditional density estimation using Artificial Neural Networks (ANN), including Sequential Neural Posterior Estimation (SNPE) [119, 99, 60], Sequential Neural Likelihood (SNL) [121], and Sequential Ratio Estimation (SRE) [72, 43].

To enhance the performance of likelihood-free inference methods, a key practice involves the selection of summary statistics [45, 154]. However, determining low-dimensional yet informative summary statistics without domain-specific expertise remains a challenge in practice. Addressing this challenge, we propose an innovative solution: the integration of a summary statistics network within the BED framework.This integration allows for the network parameters to be considered as additional design variables, enabling their joint optimization with the original model design variables. Specifically, suppose the summary statistics network is denoted as S_ϕ with ϕ being the network parameters, then original model design variables λ are replaced by the joint design variables (λ, ϕ) and the original sampling path g is replaced by the composition

$$y = S_\phi(g(\theta, \epsilon, \lambda)). \tag{6.16}$$

The approximate EIG gradient, incorporating the summary statistics network, is then

formulated as:

$$\widehat{\nabla_{(\lambda,\phi)} U(\lambda,\phi)} \approx \frac{1}{M} \sum_{i=1}^{M} \left(\widehat{s_{\mathcal{L}}}^{(i)} - \widehat{s_{\mathcal{M}}}^{(i)} \right) \cdot \nabla_{(\lambda,\phi)} S_\phi(g(\theta^{(i)}, \epsilon^{(i)}, \lambda)), \tag{6.17}$$

where $\widehat{s_{\mathcal{L}}}^{(i)}$ is computed using samples $y_{\mathcal{L}}^{(i,j)} = S_\phi(g(\theta^{(i)}, \epsilon^{(i,j)}, \lambda))$, $\epsilon^{(i,j)} \sim \pi_\epsilon(\epsilon)$ for $j = 0, ..., N$, with $\epsilon^{(i,0)} = \epsilon^{(i)}$, and $\widehat{s_{\mathcal{M}}}^{(i)}$ is computed using samples $y_{\mathcal{M}}^{(i,j)} = S_\phi(g(\theta^{(i,j)}, \epsilon^{(i,j)}, \lambda))$, $\theta^{(i,j)}\epsilon^{(i,j)} \sim \pi_\theta(\theta)\pi_\epsilon(\epsilon)$ for $j = 0, ..., N$, with $\theta^{(i,0)} = \theta^{(i)}$ and $\epsilon^{(i,0)} = \epsilon^{(i)}$.

This integration of a summary statistics network in the BED framework potentially offers two advantages: 1) Enhanced score estimation through low-dimensional data representations; and 2) Targeted optimization towards designs that maximize the EIG of summary statistics, rather than raw data. Looking ahead, our future research endeavors will delve into these advantages in greater depth. We aim to rigorously evaluate and refine this integration, exploring its potential to revolutionize the BED process, especially in applications dealing with implicit models and high-dimensional data.

6.2.3 Conclusion

In conclusion, this section of the thesis presents two innovative future research directions in Bayesian Experimental Design (BED), each addressing key challenges in the field.

The first proposed direction is the enhancement of global optimization in BED through the integration of Bayesian Optimization with Local Search (BOwLS). This approach aims to integrate the robust local search capabilities of gradient-based methods with the global search potential of Bayesian Optimization. Targeting the high-dimensional design spaces where traditional methods fall short, this strategy promises to more effectively identify global optima, potentially leading to improvements in the precision and efficiency of experimental designs.

The second direction focuses on the advancement of entropy gradient estimation for implicit models. This research path is particularly crucial in addressing the complexities inherent in Bayesian models that lack closed-form likelihood expressions. By developing new techniques for entropy gradient estimation, this approach aims to provide a deeper and more nuanced understanding of such models, enhancing the overall toolkit available for statistical analysis and application in various complex scenarios.

Together, these future research avenues underscore a commitment to addressing some of the most challenging aspects of BED, promising to drive significant advancements in the field. Their exploration and development are expected to yield valuable contributions to the realm of Bayesian statistics and experimental design methodologies.

References

[1] Martín Abadi. "TensorFlow: learning functions at scale". In: *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming*. 2016, pp. 1–1.

[2] Ibrahim Ahmad and Pi-Erh Lin. "A nonparametric estimation of the entropy for absolutely continuous distributions (corresp.)" In: *IEEE Transactions on Information Theory* 22.3 (1976), pp. 372–375.

[3] Nabil Ali Ahmed and DV Gokhale. "Entropy expressions and their estimators for multivariate distributions". In: *IEEE Transactions on Information Theory* 35.3 (1989), pp. 688–692.

[4] Alexander A Alemi et al. "Deep variational information bottleneck". In: *arXiv preprint arXiv:1612.00410* (2016).

[5] Christophe Andrieu et al. "An introduction to MCMC for machine learning". In: *Machine learning* 50 (2003), pp. 5–43.

[6] Ziqiao Ao and Jinglai Li. "An approximate KLD based experimental design for models with intractable likelihoods". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 3241–3251.

[7] Ziqiao Ao and Jinglai Li. "Entropy estimation via normalizing flow". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 9. 2022, pp. 9990–9998.

[8] Ziqiao Ao and Jinglai Li. "Entropy estimation via uniformization". In: *Artificial Intelligence* (2023), p. 103954.

[9] Ziqiao Ao and Jinglai Li. "On Estimating the Gradient of the Expected Information Gain in Bayesian Experimental Design". In: *arXiv preprint arXiv:2308.09888* (2023).

[10] Gil Ariel and Yoram Louzoun. "Estimating differential entropy using recursive copula splitting". In: *Entropy* 22.2 (2020), p. 236.

[11] Soumaya Azzi, Bruno Sudret, and Joe Wiart. "Sensitivity analysis for stochastic simulators using differential entropy". In: *International Journal for Uncertainty Quantification* 10.1 (2020).

[12] Julio R Banga and Eva Balsa-Canto. "Parameter estimation and optimal experimental design". In: *Essays in biochemistry* 45 (2008), pp. 195–210.

[13] David Barber and Felix Agakov. "The im algorithm: a variational approach to information maximization". In: *Advances in neural information processing systems* 16.320 (2004), p. 201.

[14] Russell R Barton and John S Ivey Jr. "Nelder-Mead simplex modifications for simulation optimization". In: *Management Science* 42.7 (1996), pp. 954–973.

[15] Mark A Beaumont, Wenyang Zhang, and David J Balding. "Approximate Bayesian computation in population genetics". In: *Genetics* 162.4 (2002), pp. 2025–2035.

[16] Christian Beck. "Generalised information and entropy measures in physics". In: *Contemporary Physics* 50.4 (2009), pp. 495–510.

[17] Joakim Beck et al. "Fast Bayesian experimental design: Laplace-based importance sampling for the expected information gain". In: *Computer Methods in Applied Mechanics and Engineering* 334 (2018), pp. 523–553.

[18] Jan Beirlant et al. "Nonparametric entropy estimation: An overview". In: *International Journal of Mathematical and Statistical Sciences* 6.1 (1997), pp. 17–39.

[19] Mohamed Ishmael Belghazi et al. "Mutual information neural estimation". In: *International conference on machine learning*. PMLR. 2018, pp. 531–540.

[20] Thomas B Berrett, Richard J Samworth, Ming Yuan, et al. "Efficient multivariate entropy estimation via k-nearest neighbour distances". In: *Annals of Statistics* 47.1 (2019), pp. 288–318.

[21] Gérard Biau and Luc Devroye. *Lectures on the nearest neighbor method*. Vol. 246. Springer, 2015.

[22] Lucien Birgé and Pascal Massart. "Estimation of integral functionals of a density". In: *The Annals of Statistics* (1995), pp. 11–29.

[23] Christopher M Bishop. "Mixture density networks". In: (1994).

[24] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. "Variational inference: A review for statisticians". In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877.

[25] David M Borth. "A total entropy criterion for the dual problem of model discrimination and parameter estimation". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 37.1 (1975), pp. 77–87.

[26] Johann Brehmer et al. "Mining gold from implicit models to improve likelihood-free inference". In: *Proceedings of the National Academy of Sciences* 117.10 (2020), pp. 5242–5249.

[27] John Charles Butcher. "A history of Runge-Kutta methods". In: *Applied numerical mathematics* 20.3 (1996), pp. 247–260.

[28] Daniel R Cavagnaro et al. "Adaptive design optimization: A mutual information-based approach to model discrimination in cognitive science". In: *Neural computation* 22.4 (2010), pp. 887–905.

[29] C-I Chang et al. "Survey and comparative analysis of entropy and relative entropy thresholding techniques". In: *IEE Proceedings-Vision, Image and Signal Processing* 153.6 (2006), pp. 837–850.

[30] Ricky T. Q. Chen et al. "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems* (2018).

[31] Wei-Chia Chen, Ammar Tareen, and Justin B Kinney. "Density estimation on small data sets". In: *Physical review letters* 121.16 (2018), p. 160605.

[32] Alex R Cook, Gavin J Gibson, and Christopher A Gilligan. "Optimal observation times in experimental epidemic processes". In: *Biometrics* 64.3 (2008), pp. 860–868.

[33] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[34] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[35] David Darmon. "Specific differential entropy rate estimation for continuous-valued time series". In: *Entropy* 18.5 (2016), p. 190.

[36] Mahasen B Dehideniya, Christopher C Drovandi, and James M McGree. "Optimal Bayesian design for discriminating between models with intractable likelihoods in epidemiology". In: *Computational Statistics & Data Analysis* 124 (2018), pp. 277–297.

[37] Matthias Dehmer and Abbe Mowshowitz. "A history of graph entropy measures". In: *Information Sciences* 181.1 (2011), pp. 57–78.

[38] Antonio Di Crescenzo and Maria Longobardi. "Entropy-based measure of uncertainty in past lifetime distributions". In: *Journal of Applied probability* 39.2 (2002), pp. 434–440.

[39] Adji B Dieng et al. "Prescribed generative adversarial networks". In: *arXiv preprint arXiv:1910.04302* (2019).

[40] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density estimation using Real NVP". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017.

[41] Monroe D Donsker and SR Srinivasa Varadhan. "Asymptotic evaluation of certain Markov process expectations for large time, I". In: *Communications on Pure and Applied Mathematics* 28.1 (1975), pp. 1–47.

[42] Christopher C Drovandi and Anthony N Pettitt. "Bayesian experimental design for models with intractable likelihoods". In: *Biometrics* 69.4 (2013), pp. 937–948.

[43] Conor Durkan, Iain Murray, and George Papamakarios. "On contrastive learning for likelihood-free inference". In: *International conference on machine learning*. PMLR. 2020, pp. 2771–2781.

[44] Bradley Efron and Charles Stein. "The jackknife estimate of variance". In: *The Annals of Statistics* (1981), pp. 586–596.

[45] Paul Fearnhead and Dennis Prangle. "Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74.3 (2012), pp. 419–474.

[46] Runhuan Feng and Peng Li. "Sample recycling method–a new approach to efficient nested Monte Carlo simulations". In: *Insurance: Mathematics and Economics* 105 (2022), pp. 336–359.

[47] Luisa Turrin Fernholz. *Von Mises calculus for statistical functionals*. Vol. 19. Springer Science & Business Media, 2012.

[48] Adam Foster et al. "A unified stochastic gradient approach to designing bayesian-optimal experiments". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2959–2969.

[49] Adam Foster et al. "Variational Bayesian optimal experimental design". In: *Advances in Neural Information Processing Systems* 32 (2019).

[50] Charles W Fox and Stephen J Roberts. "A tutorial on variational Bayesian inference". In: *Artificial intelligence review* 38 (2012), pp. 85–95.

[51] Cristopher GS Freitas et al. "A detailed characterization of complex networks using Information Theory". In: *Scientific reports* 9.1 (2019), p. 16689.

[52] Jianbo Gao, Jing Hu, and Wen-wen Tung. "Entropy measures for biological signal analyses". In: *Nonlinear Dynamics* 68 (2012), pp. 431–444.

[53] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. "Efficient estimation of mutual information for strongly dependent variables". In: *Artificial intelligence and statistics*. 2015, pp. 277–286.

[54] Weihao Gao, Sewoong Oh, and Pramod Viswanath. "Demystifying Fixed k-Nearest Neighbor Information Estimators". In: *IEEE Transactions on Information Theory* 64.8 (2018), pp. 5629–5661.

[55] Yuzhou Gao, Tengchao Yu, and Jinglai Li. "Bayesian optimization with local search". In: *Machine Learning, Optimization, and Data Science: 6th International Conference, LOD 2020, Siena, Italy, July 19–23, 2020, Revised Selected Papers, Part II 6*. Springer. 2020, pp. 350–361.

[56] Mathieu Germain et al. "Made: Masked autoencoder for distribution estimation". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 881–889.

[57] Takashi Goda et al. "Unbiased MLMC stochastic gradient-based optimization of Bayesian experimental designs". In: *SIAM Journal on Scientific Computing* 44.1 (2022), A286–A311.

[58] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[59] Mohammed Nawaz Goria et al. "A new class of random vector entropy estimators and its applications in testing statistical hypotheses". In: *Journal of Nonparametric Statistics* 17.3 (2005), pp. 277–297.

[60] David Greenberg, Marcel Nonnenmacher, and Jakob Macke. "Automatic posterior transformation for likelihood-free inference". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2404–2414.

[61] L Gyorfi and EC Van der Meulen. "On nonparametric estimation of entropy functionals". In: *Nonparametric Functional Estimation and Related Topics,(G. Roussas ed.), Kluwer Academic Publisher, Amsterdam* (1990), pp. 81–95.

[62] L Györfi and Edward C van der Meulen. "An entropy estimate based on a kernel density estimation". In: *Limit Theorems in Probability and Statistics* (1989), pp. 229–240.

[63] László Györfi and Edward C Van der Meulen. "Density-free convergence properties of various estimators of entropy". In: *Computational Statistics & Data Analysis* 5.4 (1987), pp. 425–436.

[64] Tuomas Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International conference on machine learning*. PMLR. 2018, pp. 1861–1870.

[65] Markus Hainy, Werner G Müller, and Helga Wagner. "Likelihood-free simulation-based optimal design with an application to spatial extremes". In: *Stochastic Environmental Research and Risk Assessment* 30.2 (2016), pp. 481–492.

[66] Peter Hall and Sally C Morton. "On the estimation of entropy". In: *Annals of the Institute of Statistical Mathematics* 45 (1993), pp. 69–88.

[67] M Hamada et al. "Finding near-optimal Bayesian experimental designs via genetic algorithms". In: *The American Statistician* 55.3 (2001), pp. 175–181.

[68] Yanjun Han et al. "Optimal rates of entropy estimation over Lipschitz balls". In: *The Annals of Statistics* 48.6 (2020), pp. 3228–3250.

[69] Michael Hardy. "Combinatorics of partial derivatives". In: *arXiv preprint math/0601149* (2006).

[70] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[71] Frank Heinrich et al. "Information gain from isotopic contrast variation in neutron reflectometry on protein–membrane complex structures". In: *Journal of applied crystallography* 53.3 (2020), pp. 800–810.

[72] Joeri Hermans, Volodimir Begy, and Gilles Louppe. "Likelihood-free mcmc with amortized approximate ratio estimators". In: *International conference on machine learning*. PMLR. 2020, pp. 4239–4248.

[73] R Devon Hjelm et al. "Learning deep representations by mutual information estimation and maximization". In: *International Conference on Learning Representations*. 2018.

[74] Xun Huan and Youssef M Marzouk. "Simulation-based optimal Bayesian experimental design for nonlinear systems". In: *Journal of Computational Physics* 232.1 (2013), pp. 288–317.

[75] Aapo Hyvärinen and Peter Dayan. "Estimation of non-normalized statistical models by score matching." In: *Journal of Machine Learning Research* 6.4 (2005).

[76] Shunsuke Ihara. *Information theory for continuous systems*. Vol. 2. World Scientific, 1993.

[77] AV Ivanov and MN Rozhkova. "Properties of the Statistical Estimate of the Entropy of a Random Vector With a Probability Density." In: *PROB. INFO. TRANS.* 17.3 (1982), pp. 171–177.

[78] Edwin T Jaynes. "Information theory and statistical mechanics". In: *Physical review* 106.4 (1957), p. 620.

[79] Harry Joe. "Estimation of entropy and other functionals of a multivariate density". In: *Annals of the Institute of Statistical Mathematics* 41 (1989), pp. 683–697.

[80] Kirthevasan Kandasamy et al. "Nonparametric von Mises Estimators for Entropies, Divergences and Mutual Informations." In: *NIPS*. Vol. 15. 2015, pp. 397–405.

[81] Steven Kleinegesse and Michael U Gutmann. "Bayesian experimental design for implicit models by mutual information neural estimation". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5316–5326.

[82] Steven Kleinegesse and Michael U Gutmann. "Efficient Bayesian experimental design for implicit models". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 476–485.

[83] Steven Kleinegesse and Michael U Gutmann. "Gradient-based Bayesian experimental design for implicit models using mutual information lower bounds". In: *arXiv preprint arXiv:2105.04379* (2021).

[84] LF Kozachenko and Nikolai N Leonenko. "Sample estimate of the entropy of a random vector". In: *Problemy Peredachi Informatsii* 23.2 (1987), pp. 9–16.

[85] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. "Estimating mutual information". In: *Physical review E* 69.6 (2004), p. 066138.

[86] Clemens Kreutz and Jens Timmer. "Systems biology: experimental design". In: *The FEBS journal* 276.4 (2009), pp. 923–942.

[87] Akshay Krishnamurthy et al. "Nonparametric estimation of renyi divergence and friends". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 919–927.

[88] Isthrinayagy Krishnarajah et al. "Novel moment closure approximations in stochastic epidemics". In: *Bulletin of mathematical biology* 67 (2005), pp. 855–873.

[89] Jeremy Lewi, Robert Butera, and Liam Paninski. "Sequential optimal design of neurophysiology experiments". In: *Neural computation* 21.3 (2009), pp. 619–687.

[90] Yingzhen Li and Richard E Turner. "Gradient estimators for implicit models". In: *arXiv preprint arXiv:1705.07107* (2017).

[91] Jae Hyun Lim et al. "AR-DAE: towards unbiased neural entropy gradient estimation". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6061–6071.

[92] Qiang Liu and Dilin Wang. "Stein variational gradient descent: A general purpose bayesian inference algorithm". In: *Advances in neural information processing systems* 29 (2016).

[93] Gabriel Loaiza-Ganem, Yuanjun Gao, and John P Cunningham. "Maximum entropy flow networks". In: *arXiv preprint arXiv:1701.03504* (2017).

[94] Gabriel Loaiza-Ganem, Yuanjun Gao, and John P. Cunningham. "Maximum Entropy Flow Networks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[95] Quan Long. "Multimodal information gain in Bayesian design of experiments". In: *Computational Statistics* 37.2 (2022), pp. 865–885.

[96] Quan Long et al. "Fast estimation of expected information gains for Bayesian experimental designs based on Laplace approximations". In: *Computer Methods in Applied Mechanics and Engineering* 259 (2013), pp. 24–39.

[97] Warren M Lord, Jie Sun, and Erik M Bollt. "Geometric k-nearest neighbor estimation of entropy and mutual information". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.3 (2018), p. 033114.

[98] Alfred James Lotka. *Elements of physical biology*. Williams & Wilkins, 1925.

[99] Jan-Matthis Lueckmann et al. "Flexible statistical inference for mechanistic models of neural dynamics". In: *Advances in neural information processing systems* 30 (2017).

[100] Jiankun Lyu et al. "Ultra-large library docking for discovering new chemotypes". In: *Nature* 566.7743 (2019), pp. 224–229.

[101] Rafael Martí. "Multi-start methods". In: *Handbook of metaheuristics* (2003), pp. 355–368.

[102] JA Melendez et al. "Designing optimal experiments: an application to proton Compton scattering". In: *The European Physical Journal A* 57 (2021), pp. 1–24.

[103] Ruth K Meyer and Christopher J Nachtsheim. "The coordinate-exchange algorithm for constructing exact optimal experimental designs". In: *Technometrics* 37.1 (1995), pp. 60–69.

[104] Erik G Miller. "A new class of entropy estimators for multi-dimensional densities". In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).* Vol. 3. IEEE. 2003, pp. III–297.

[105] Jonas Mockus. "The application of Bayesian methods for seeking the extremum". In: *Towards global optimization* 2 (1998), p. 117.

[106] Shakir Mohamed et al. "Monte carlo gradient estimation in machine learning". In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 5183–5244.

[107] Kevin R Moon et al. "Ensemble estimation of information divergence". In: *Entropy* 20.8 (2018), p. 560.

[108] Peter Müller. "Simulation based optimal design". In: *Handbook of Statistics* 25 (2005), pp. 509–518.

[109] Jay I Myung, Daniel R Cavagnaro, and Mark A Pitt. "A tutorial on adaptive design optimization". In: *Journal of mathematical psychology* 57.3-4 (2013), pp. 53–67.

[110] Alireza Namdari and Zhaojun Li. "A review of entropy measures for uncertainty quantification of stochastic processes". In: *Advances in Mechanical Engineering* 11.6 (2019), p. 1687814019857350.

[111] Radford M Neal. "Slice sampling". In: *The annals of statistics* 31.3 (2003), pp. 705–767.

[112] John A Nelder and Roger Mead. "A simplex method for function minimization". In: *The computer journal* 7.4 (1965), pp. 308–313.

[113] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. "Estimating divergence functionals and the likelihood ratio by convex risk minimization". In: *IEEE Transactions on Information Theory* 56.11 (2010), pp. 5847–5861.

[114] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748* (2018).

[115] Antony Overstall and James McGree. "Bayesian Design of Experiments for Intractable Likelihood Models Using Coupled Auxiliary Models and Multivariate Emulation". In: *Bayesian Analysis* 15.1 (2020), pp. 103–131.

[116] Antony M Overstall and David C Woods. "Bayesian design of experiments using approximate coordinate exchange". In: *Technometrics* 59.4 (2017), pp. 458–470.

[117] Filippo Pagani, Martin Wiegand, and Saralees Nadarajah. "An n-dimensional Rosenbrock Distribution for MCMC Testing". In: *arXiv preprint arXiv:1903.09556* (2019).

[118] Liam Paninski. "Asymptotic theory of information-theoretic experimental design". In: *Neural Computation* 17.7 (2005), pp. 1480–1507.

[119] George Papamakarios and Iain Murray. "Fast ε-free inference of simulation models with bayesian conditional density estimation". In: *Advances in neural information processing systems* 29 (2016).

[120] George Papamakarios, Theo Pavlakou, and Iain Murray. "Masked autoregressive flow for density estimation". In: *Advances in Neural Information Processing Systems*. 2017, pp. 2338–2347.

[121] George Papamakarios, David Sterratt, and Iain Murray. "Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 837–848.

[122] George Papamakarios et al. "Normalizing flows for probabilistic modeling and inference". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 2617–2680.

[123] No-Wook Park. "Using maximum entropy modeling for landslide susceptibility mapping with multiple geoenvironmental data sets". In: *Environmental Earth Sciences* 73 (2015), pp. 937–949.

[124] Emanuel Parzen. "On estimation of a probability density function and mode". In: *The annals of mathematical statistics* 33.3 (1962), pp. 1065–1076.

[125] Adam Paszke et al. "Automatic differentiation in pytorch". In: (2017).

[126] Martin Peifer and Jens Timmer. "Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting". In: *IET systems biology* 1.2 (2007), pp. 78–88.

[127] Steven J Phillips, Robert P Anderson, and Robert E Schapire. "Maximum entropy modeling of species geographic distributions". In: *Ecological modelling* 190.3-4 (2006), pp. 231–259.

[128] Georg Pichler et al. "A differential entropy estimator for training neural networks". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 17691–17715.

[129] Ben Poole et al. "On variational bounds of mutual information". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5171–5180.

[130] Michael JD Powell. "On search directions for minimization algorithms". In: *Mathematical programming* 4 (1973), pp. 193–201.

[131] David J Price et al. "An induced natural selection heuristic for finding optimal Bayesian experimental designs". In: *Computational Statistics & Data Analysis* 126 (2018), pp. 112–124.

[132] David J Price et al. "On the efficient determination of optimal Bayesian experimental designs using ABC: A case study in optimal observation of epidemics". In: *Journal of Statistical Planning and Inference* 172 (2016), pp. 1–15.

[133] Thomas Rainforth. "Automating inference, learning, and design using probabilistic programming". PhD thesis. University of Oxford, 2017.

[134] Tom Rainforth et al. "Modern bayesian experimental design". In: *arXiv preprint arXiv:2302.14545* (2023).

[135] Tom Rainforth et al. "On nesting monte carlo estimators". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4267–4276.

[136] Bo Ranneby. "The maximum spacing method. An estimation method related to the maximum likelihood method". In: *Scandinavian Journal of Statistics* (1984), pp. 93–112.

[137] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. "On stochastic optimal control and reinforcement learning by approximate inference". In: *Proceedings of Robotics: Science and Systems VIII* (2012).

[138] Danilo Rezende and Shakir Mohamed. "Variational inference with normalizing flows". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1530–1538.

[139] Chang-han Rhee and Peter W Glynn. "Unbiased estimation with square root convergence for SDE models". In: *Operations Research* 63.5 (2015), pp. 1026–1043.

[140] Murray Rosenblatt. "Remarks on some nonparametric estimates of a density function". In: *The annals of mathematical statistics* (1956), pp. 832–837.

[141] Elizabeth Ryan, Christopher Drovandi, and Anthony Pettitt. "Fully Bayesian experimental design for pharmacokinetic studies". In: *Entropy* 17.3 (2015), pp. 1063–1089.

[142] Elizabeth G Ryan et al. "A review of modern computational algorithms for Bayesian optimal design". In: *International Statistical Review* 84.1 (2016), pp. 128–154.

[143] Elizabeth G Ryan et al. "Towards Bayesian experimental design for nonlinear models that require a large number of sampling times". In: *Computational Statistics & Data Analysis* 70 (2014), pp. 45–60.

[144] Kenneth J Ryan. "Estimating expected information gains for experimental designs with application to the random fatigue-limit model". In: *Journal of Computational and Graphical Statistics* 12.3 (2003), pp. 585–603.

[145] Nicol N Schraudolph. "Gradient-based manipulation of nonparametric entropy estimates". In: *IEEE Transactions on Neural Networks* 15.4 (2004), pp. 828–837.

[146] Paola Sebastiani and Henry P Wynn. "Maximum entropy sampling and optimal Bayesian experimental design". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 62.1 (2000), pp. 145–157.

[147] Matthias Seeger. "Gaussian processes for machine learning". In: *International journal of neural systems* 14.02 (2004), pp. 69–106.

[148] Matthias W Seeger and Hannes Nickisch. "Compressed sensing and Bayesian experimental design". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 912–919.

[149] Claude Elwood Shannon. "A mathematical theory of communication". In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[150] Michael C Shewry and Henry P Wynn. "Maximum entropy sampling". In: *Journal of applied statistics* 14.2 (1987), pp. 165–170.

[151] Jiaxin Shi, Shengyang Sun, and Jun Zhu. "A spectral approach to gradient estimation for implicit distributions". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4644–4653.

[152] Harshinder Singh et al. "Nearest neighbor estimates of entropy". In: *American journal of mathematical and management sciences* 23.3-4 (2003), pp. 301–321.

[153] Shashank Singh and Barnabás Póczos. "Finite-sample analysis of fixed-k nearest neighbor density functional estimators". In: *Advances in neural information processing systems*. 2016, pp. 1217–1225.

[154] Scott A Sisson, Yanan Fan, and Mark Beaumont. *Handbook of approximate Bayesian computation*. CRC Press, 2018.

[155] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical bayesian optimization of machine learning algorithms". In: *Advances in neural information processing systems* 25 (2012).

[156] Jiaming Song and Stefano Ermon. "Understanding the limitations of variational mutual information estimators". In: *arXiv preprint arXiv:1910.06222* (2019).

[157] Yang Song et al. "Sliced score matching: A scalable approach to density and score estimation". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 574–584.

[158] James C Spall. "An overview of the simultaneous perturbation method for efficient optimization". In: *Johns Hopkins apl technical digest* 19.4 (1998), pp. 482–492.

[159] James C Spall. "Implementation of the simultaneous perturbation algorithm for stochastic optimization". In: *IEEE Transactions on aerospace and electronic systems* 34.3 (1998), pp. 817–823.

[160] Kumar Sricharan, Dennis Wei, and Alfred O Hero. "Ensemble estimators for multivariate entropy estimation". In: *IEEE transactions on information theory* 59.7 (2013), pp. 4374–4388.

[161] Niranjan Srinivas et al. "Gaussian process optimization in the bandit setting: No regret and experimental design". In: *arXiv preprint arXiv:0912.3995* (2009).

[162] Charles M Stein. "Estimation of the mean of a multivariate normal distribution". In: *The annals of Statistics* (1981), pp. 1135–1151.

[163] Taiji Suzuki et al. "Approximating mutual information by maximum likelihood density ratio estimation". In: *New challenges for feature selection in data mining and knowledge discovery*. PMLR. 2008, pp. 5–20.

[164] Ira Swameye et al. "Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by databased modeling". In: *Proceedings of the National Academy of Sciences* 100.3 (2003), pp. 1028–1033.

[165] Owen Thomas et al. "Likelihood-free inference by ratio estimation". In: *Bayesian Analysis* 17.1 (2022), pp. 1–31.

[166] Michalis K Titsias and Francisco Ruiz. "Unbiased implicit variational inference". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 167–176.

[167] Surya T Tokdar and Robert E Kass. "Importance sampling: a review". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2.1 (2010), pp. 54–60.

[168] Marc Toussaint. "Robot trajectory optimization using approximate inference". In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 1049–1056.

[169] Panagiotis Tsilifis, Roger G Ghanem, and Paris Hajali. "Efficient Bayesian experimentation using an expected information gain lower bound". In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1 (2017), pp. 30–62.

[170] Alexandre B Tsybakov and EC Van der Meulen. "Root-n consistent estimators of entropy for densities with unbounded support". In: *Scandinavian Journal of Statistics* (1996), pp. 75–83.

[171] Oldrich Vasicek. "A test for normality based on sample entropy". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 38.1 (1976), pp. 54–59.

[172] Apostol Vassilev and Timothy A Hall. "The importance of entropy to information security". In: *Computer* 47.2 (2014), pp. 78–81.

[173] Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. "A primer on kernel methods". In: *Kernel methods in computational biology* 47 (2004), pp. 35–70.

[174] Pascal Vincent. "A connection between score matching and denoising autoencoders". In: *Neural computation* 23.7 (2011), pp. 1661–1674.

[175] Vito Volterra. *Variazioni e fluttuazioni del numero d'individui in specie animali conviventi*. C. Ferrari, 1927.

[176] Liangjian Wen et al. "Gradient estimation of information measures in deep learning". In: *Knowledge-Based Systems* 224 (2021), p. 107046.

[177] Eric Wolsztynski, Eric Thierry, and Luc Pronzato. "Minimum-entropy estimation in semi-parametric models". In: *Signal Processing* 85.5 (2005), pp. 937–949.

[178] Mingzhang Yin and Mingyuan Zhou. "Semi-implicit variational inference". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5660–5669.

[179] Longlin Yu and Cheng Zhang. "Semi-Implicit Variational Inference via Score Matching". In: *arXiv preprint arXiv:2308.10014* (2023).

[180] Xinjie Yu and Mitsuo Gen. *Introduction to evolutionary algorithms*. Springer Science & Business Media, 2010.

[181] Vincent D Zaballa and Elliot E Hui. "Stochastic Gradient Bayesian Optimal Experimental Designs for Simulation-based Inference". In: *arXiv preprint arXiv:2306.15731* (2023).

[182] Jiaxin Zhang, Sirui Bi, and Guannan Zhang. "A scalable gradient free method for bayesian experimental design with implicit models". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 3745–3753.

[183] Kun Zhang et al. "Sample Recycling for Nested Simulation with Application in Portfolio Risk Measurement". In: *arXiv preprint arXiv:2203.15929* (2022).

[184] Sue Zheng, Jason Pacheco, and John Fisher. "A robust approach to sequential information theoretic planning". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5941–5949.

[185] Brian D Ziebart et al. "Maximum entropy inverse reinforcement learning." In: *Aaai*. Vol. 8. Chicago, IL, USA. 2008, pp. 1433–1438.