Department of Complexity Science and Engineering
Graduate School of Frontier Sciences
The University of Tokyo
東京大学大学院新領域創成科学研究科
複雑理工学専攻

2020
令和元年度

Master's Thesis
修士論文

# Study on variance reduction for variational inference via multi-level Monte Carlo
# （マルチレベルモンテカルロ法を用いた変分推論における分散抑制法についての研究）

Supervisor: Lecturer Issei Sato
指導教員：佐藤　一誠　講師

2020年1月28日提出
Submitted January 28, 2020

**Masahiro Fujisawa**
**藤澤　将広**
**47-186121**

# Abstract

*Statistical machine learning* is a data processing technology that can make an intellectual inference, e.g., regression analysis, classification, or clustering, by automatically extracting important patterns from data. It has archived great success as an effective decision making tool and has been employed in various research/social application fields, including medical diagnosis and automatic driving systems. However, in the real world, we often obtain highly contaminated data or rarely get an enough amount of data for proper inference. Such a case, *uncertainty* arises on the output from statistical machine learning and can lead to wrong decision-making. This is a crucial issue; for example, in medical diagnosis or automatic driving, an decision-making error can be fatal or sometimes even life-threatening.

To avoid this risk, uncertainty estimation for the output is necessary. Although Bayesian inference is one of the most effective ways to evaluate uncertainty in this context, modern Bayesian statistic relies on a complex model whose posterior is hard to compute and can be applied to large-scale data, which can increase the computational load. Therefore, making an exact inference is typically intractable. Even though variational inference (VI) is typically used to approximate the posterior, the objective function of VI itself becomes intractable and can not be computed in a closed form for more complicated models such as Bayesian neural networks.

To bypass this, Monte Carlo variational inference (MCVI), which is a stochastic inference method based on the stochastic gradient, has proposed. However, the stochastic gradient often takes high variance, and this negatively affects the performance of MCVI. Therefore, the variance reduction of the stochastic gradient in variational inference is one of the important research topics, and various methods have been proposed in recent years. The drawbacks of these methods are that we need to use *heuristic* information and can not enjoy theoretical guarantee for convergence, variance reduction, or the variance-reduced gradient.

In this thesis, we propose a framework for variance reduction using the multi-level Monte Carlo (MLMC) method that makes it possible to "recycle" the parameter in the past obtained naturally in optimization. The framework is naturally compatible with reparameterized gradient estimators that are being used as a standard in MCVI. The proposed method has a novel optimization algorithm based on SGD and estimates the sample size for stochastic gradient estimation per level adaptively according to the ratio of the variance and computation cost in each iteration. Furthermore, we analyze the convergence of the gradient norm in stochastic gradient descent, the scale of the variance of the gradient estimator, and its quality in each optimization step on the basis of the *signal-to-noise* ratio. Finally, we experimentally evaluate the proposed method by comparing it with baseline methods on several benchmark data sets and confirm that the proposed method archives faster convergence, sometimes gets closer to the optimal value, and reduces the variance of the gradient estimator more than the other methods do.

# 概要

統計的機械学習は，データから重要なパターンを自動的に抽出することにより，回帰，分類，クラスタリングなどといった知的推論を行うことができるデータ処理技術である．この技術は，効果的な意思決定方法として非常に有益なものであると見做され，医療診断や車の自動運転システムなどを含む様々な研究・社会応用の分野に適用されている．しかし，実社会においては，得られたデータが激しいノイズに汚染されていたり，適正な推論結果を得るために十分な量のデータを入手することが困難である場合が頻繁に生じる．そのようなデータを，統計的機械学習を用いて解析した場合，その出力に，「不確実性」が生じてしまう．この「不確実性」の発生は，誤った意思決定に繋がり得るために，重大な問題となっている．例えば，医療診断や自動運転システムの場合，誤った意思決定が何らかの危険性，あるいは命を脅かす事態に繋がりかねない．

このようなリスクを回避するためには，統計的機械学習技術から得られた出力の不確実性を推定し，それを加味した決定を下すことが必要である．不確実性を推定・評価する上で最も効果的な方法の一つとされているものが，ベイズ推論である．しかし，近年のベイズ推論は，事後分布が簡単には計算できない，あるいは解析的な計算が不可能であるほどまでに複雑な確率モデルに適用されることが多い．加えて，大規模なデータに適用されることもあり，計算負荷が大きくなり得る．このような場合，事後分布自体を，扱いやすい別の分布で近似することを考えるが，その際によく使用されるのが，変分推論である．しかしながら，ベイジアンニューラルネットワークのような，さらに複雑なモデルを構築する場合，変分推論の目的関数自体が解析的に計算不可能になってしまい，閉じた形で推論を行うことができない．

この問題を回避するために，確率的勾配を基に確率的推論を行うモンテカルロ変分推論が開発されたが，確率的勾配はしばしば分散が大きくなってしまい，これがモンテカルロ変分推論の性能に悪い影響を与えている．よって，確率的勾配における分散抑制は，重要な研究課題の一つとなっており，様々な方法が提案されてきた．これらの手法の欠点は，分散を抑制するための新しい関数などを人為的に付与しなくてはならず，さらには「推論の収束性」,「分散抑制の程度」,「分散抑制された確率的勾配の質」についての理論保証がない，あるいは享受できないことである．

そこで，本論文では，マルチレベルモンテカルロ法を用い，最適化の行程で自然に入手できる過去のパラメータ情報を再利用する，新しい分散抑制法を提案する．提案法は，モンテカルロ変分推論において標準的に使用されつつある再パラメータ化勾配のもとで自然に導出される．また，最適化における各行程において，確率的勾配の分散と計算コストの比によって推定に用いる確率変数の数を動的に推定する，新しい確率的勾配推定法および最適化アルゴリズムを提案する．さらに，確率的勾配降下法における提案法の勾配の収束性，勾配の分散の大きさ，および SN 比に基づく勾配推定量の質について，理論解析を行う．最後に，いくつかの数値実験を行い，基準となる既存法と比較し，提案法の性能を評価する．この実験を通して，提案法が，既存法に比べて，推論の収束が高速で，モデルによっては最適な解を得，勾配の分散を抑制できていることを確認する．

# Acknowledgement

First of all, I would like to express my deepest gratitude to my academic supervisor Lecturer Issei Sato for the continuous support of my research progression. His insightful guidance helped me in all the time of research and writing this thesis. I deeply respect his patience, motivation, and immense knowledge. Next, I would like to express my most sincere gratitude to Professor Masashi Sugiyama for his tremendous support and guidance throughout my master's program. Despite his overcrowding schedule, he always gave me the fruitful advice for my research and took care of me in many aspects everywhen. Without his assistance, I could not concentrate my research on peace, and this thesis would not have been possible. I would also like to thank Lecturer Junya Honda for the encouraging advice, discussions, and insightful comments. He gave me his opinion about my research at any time after the presentation.

Furthermore, my gratitude goes to all the members of Sugiyama-Sato-Honda Laboratory. Especially, I am grateful to Dr. Ikko Yamane, Futoshi Futami, and Soma Yokoi. I had a lot of fruitful discussions with them. Thanks to their immense knowledge in machine learning, I could find out the potential of my research and make my thesis better. Additionally, I would like to thank all the members of the NTT Communication Science Laboratory, especially Dr. Naonori Ueda, Dr. Tomoharu Iwata, Naoki Marumo, Dr. Masakazu Ishihata, and Dr. Takuma Otsuka. They pointed out the essential issues and gave helpful comments to blush up my research. I feel pleased to get a chance to present my research in front of them.

Finally, I must express my sincere gratitude to my parents for providing me with their lasting support and continuous encouragement throughout my life.

# Publications Related to This Thesis

1. Fujisawa, M. and Sato, I.
   Multi-level Monte Calro Variational Inference. *arXiv preprint* arXiv:1902.00468, 2019.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In this chapter, we state the introduction of this thesis. First of all, we overview the outline of statistical machine learning and uncertainty in Section 1.1. Next, we briefly introduce Bayesian inference that is important method to evaluate uncertainty in Section 1.2. Moreover, we explain an approximate inference including the relationship between VI and MCMC in Section 1.3. In addition, we introduce Monte Carlo variational inference that is the main method we focus on in Section 1.4. Finally, we summary the contribution and the organization of this thesis in Section 1.5 and Section 1.6.

## 1.1 Statistical Machine Learning and Uncertainty

*Statistical machine learning* is a data processing technology that can make an intellectual inference, e.g., regression analysis, classification, or clustering, by automatically extracting important patterns from data [33]. It has archived great success as an effective decision making tool and has been employed in various research/social application fields, e.g., recommendation [41, 72], image recognition [36], causal inference [42, 49], simulation [13], natural language process [10, 89], medical diagnosis [17, 45], and automatic driving systems [1]. Beyond the research community, it has been paid attention to by many people, societies, and industries. Therefore, it has been extensively studied over the past few decades.

On the other hand, we can not always get clean and/or enough data. For example, because of recent advances in sensor technology, we can get a vast amount of data with spiky noise or so many annotated data obtained by using crowd-sourcing technology, but it can be full of human errors [69, 57, 96, 4]. Besides, in astronomy or biology, it is hard to gather enough data because huge costs are needed to obtain lunar photos or conduct experiments in many settings. Such contaminated and/or limited data causes the *uncertainty* on the output from statistical machine learning technology (called *aleatoric* uncertainty or *epistemic* uncertainty [14]) and can yield wrong decision-making. It is sometimes a fatal problem, i.e., in the case of a medical diagnosis problem. Therefore, an additional inference method with uncertainty estimation is required to avoid making a potentially wrong decision.

**Aleatoric Uncertainty [14]** *Aleatoric* uncertainty captures noise inherent in the training data set. For example, in Figure 1.1, this could be sensor noise or motion noise. In addition, this uncertainty can not be reduced even if an enough amount of data is obtained. The column (d) on Figure 1.1 shows that aleatoric uncertainty increases on object boundaries and for objects far from the camera.

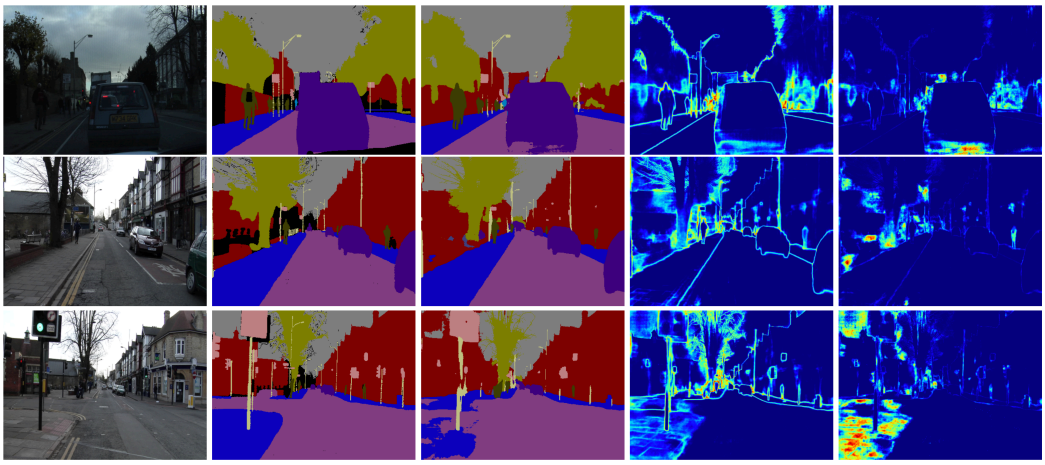


**Figure 1.1:** Illustrating the difference between aleatoric and epistemic uncertainty for semantic segmentation on the CamVid data set [6] [46].

**Epistemic Uncertainty [14]** *Epistemic* uncertainty accounts for uncertainty in the model parameters, i.e., uncertainty, which captures our ignorance about which model generated our collected data [46]. This uncertainty can be eliminated away if an enough amount of data is obtained. In addition, it is often regarded as model uncertainty. The column (e) on Figure 1.1 shows that epistemic uncertainty increases for semantically and visually challenging pixels.

**Uncertainty Evaluation on the Bayesian Model** The bottom row on Figure 1.1 shows that the segmentation model fails to segment the footpath due to increased *epistemic* uncertainty but not *aleatoric* uncertainty. Therefore, the epistemic uncertainty evaluation would be important for making proper decision. One of the ways to evaluate epistemic uncertainty is constructing models with the Bayesian approach, which is called the Bayesian model. The Bayesian model consists of probability density functions and is inferred under Bayes's theorem. This inference process is called Bayesian inference. Thanks to the Bayesian inference, we can get the output with its distribution and confirm the uncertainty according to the shape of it.

## 1.2 Bayesian Inference

As noticed in the previous section, Bayesian inference is an essential method of statistical inference to make reasonable decisions with uncertainty. The object of Bayesian inference is to estimate the posterior distribution of latent variables $\mathbf{z}$ given observation $\mathbf{x}$: $p(\mathbf{z}|\mathbf{x})$, i.e.,

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{\int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z}}.$$

In the example of uncertainty introduced in the previous section, $\mathbf{x}$ is the input picture in the CamVid data set and $\mathbf{z}$ is the weight of the model constructed for semantic segmentation. The exact computation for $p(\mathbf{z}|\mathbf{x})$ amounts to sum or integration over all $\mathbf{z}$. We can flexibly

incorporate hypotheses and knowledge in the model as a likelihood and a prior. Because of this flexibility, Bayesian inference has found many applications in a wide range of research fields, e.g., medical science [51], biology [75, 34, 12], sociology [93], psychology [91], and economics [90]. By using this, we can evaluate epistemic uncertainty on the basis of the posterior: $p(\mathbf{z}|\mathbf{x})$, e.g., the scale parameter in the Gaussian distribution.

## 1.3 Approximate Inference

Modern Bayesian statistic relies on a complex (e.g., non-conjugate) model for which the posterior is not easy or able to compute. Furthermore, the entire inference procedure can be large-scale and therefore making an exact inference is typically intractable. To overcome these situations, we need efficient algorithms for approximating the posterior.

**Markov Chain Monte Carlo (MCMC)** In the context of posterior approximation, MCMC [37, 21] has been used for a long time. This method has evolved independently of modern Bayesian statistics. The famous algorithm of MCMC is the Metropolis-Hastings algorithm [59, 37], and the Gibbs sampler [23]. These methods have been applied to Bayesian statistics by Gelfand and Smith [22]. MCMC algorithms have been widely studied and extended by Blei et al. [3] (please see Robert and Casella [74] for a perspective). Furthermore, they also provide guarantees of producing (asymptotically) exact samples from the target distribution.

**Variational Inference (VI)** The object of VI is to seek a distribution, from a variational family of distributions, that best approximates an intractable posterior distribution [60]. Recently, extensive research has been conducted to reveal the behavior of VI theoretically and improve the performance of VI [7, 18, 56, 60, 64, 67, 68, 70, 76, 79, 80, 83, 84, 87, 88, 94].

**Relationship between VI and MCMC** However, when data sets are large, or the model we construct is complex, the problem occurs in the usage of MCMC algorithms. That is, they tend to be computationally intensive and need much computation for convergence. On the other hand, VI tends to be faster than MCMC for the large-scale data sets because it can easily enjoy the stochastic optimization scheme, which is computationally highly efficient [5]. It is important to note that, however, VI does not enjoy guarantees of getting samples from the target distribution because it can only find a density closest to the target distribution in a variational family.

In short, if we want to explore models quickly, VI would be suitable; on the other hand, if we want to get precise results and be able to pay a high computation cost for it , MCMC would be suitable.

Not only the data set size, but the geometry of the posterior distribution is also an important factor to use VI and MCMC properly. For example, Gibbs sampling [23] is a powerful tool to sample from target distributions. However, Gibbs sampling suffers restriction on the choice of models in. For the models where Gibbs sampling is not permitted, VI may perform better than a more general MCMC technique (e.g., Hamiltonian Monte Carlo [19]), even for small data sets [50].

In the big-data era, a scalable and high-speed inference method such as VI has become more and more required [95]. Therefore, understanding VI and improving its performance contribute to broadening the utility of modern Bayesian statistics in various situations.

## 1.4 Monte Carlo Variational Inference

Unfortunately, because of the very high complexity of recent models such as Bayesian neural networks, the objective function of VI itself is often intractable and can not be computed in a closed form. In this case, we often use stochastic gradient methods called Monte Carlo variational inference (MCVI) or black box variational inference [68]. In MCVI, sampling from a variational posterior distribution is the key to estimating the gradient stochastically. However, the stochastic gradient obtained with Monte Carlo approximation may cause slow convergence because of the high-variance. Therefore, the variance of the stochastic gradient estimator needs to be controlled carefully to make MCVI useful.

There are two standard MCVI gradient estimators: the score function gradient estimator [64, 68] and the reparameterized gradient estimator [83, 70, 48]. The score function gradient estimator can be applied to both discrete and continuous random variables, but it often has high variance. In contrast, the reparameterized gradient estimator often has a lower variance for continuous random variables. Recently, Ruiz et al. [79] bridged these two gradient estimators. In addition, Tokui and Sato [85] and Jang et al. [40] proposed a reparameterization trick for discrete or categorical variables. Moreover, theoretical properties of the reparameterized gradient have been analyzed recently [94, 18]. As a result, the reparameterized gradient has become a more practical way to reduce the variance of a gradient estimator. Furthermore, Buchholz et al. [7] proposed using a randomized quasi-Monte Carlo (RQMC) method for MCVI, which can reduce the variance of gradient estimator lower than that of the MC method.

## 1.5 Contribution of This Thesis

In this section, we summary the technical contribution and explain how this thesis contributes to society.

### 1.5.1 Technical Contributions of This Thesis

In this thesis, we propose a novel framework for reducing the variance of gradient estimator in the MCVI framework on the basis of the multi-level Monte Carlo (MLMC) method. The proposed method is naturally derived from the reparameterized gradient estimator based on the reparameterization trick [48], achieves better predictive performance, and provides a faster convergence speed than the baseline methods do. Moreover, the proposed method can be easily implemented in modern inference libraries such as Stan [9], Edward [86], TensorFlow Probability [16], and Pytorch [65].

Through this thesis, we investigate the idea of using the MLMC method for MCVI on reparameterized gradient estimation. In addition, we develop an algorithm that gives a low-variance gradient estimator with decreasing the number of samples by "recycling" the gradient in the past.

We also guarantee the optimization convergence theoretically on the basis of stochastic gradient descent with a decreasing learning rate in both the proposed method and the existing methods. Thorough the convergence analysis, we found that the proposed method is able to accelerate the optimization compared to the baseline methods. Besides, we analyze the quality of the gradient estimator in terms of the *signal-to-noise* ratio (SNR). From this analysis, we show the proposed method could control it by making use of a learning-rate scheduler.

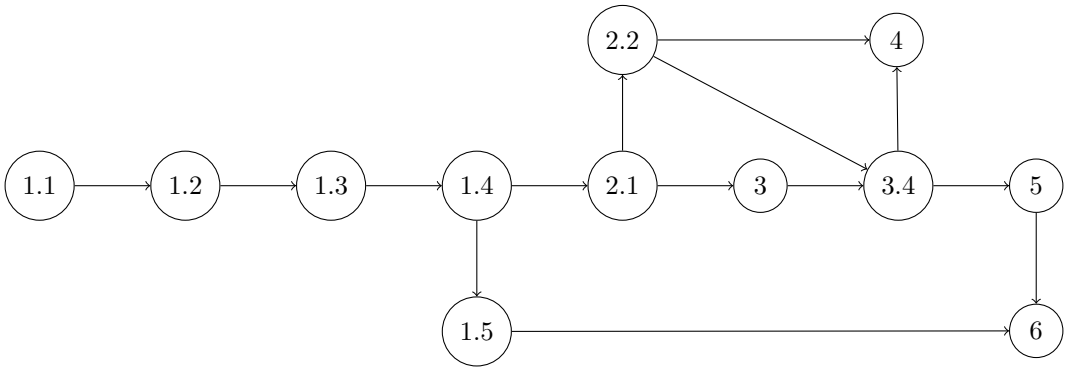


**Figure 1.2:** Relationship of chapters and sections

Finally, we confirm the performance of the proposed method through three experiments on toy and real-world data sets. According to the experimental results, we find that the proposed method could asymptotically reduce the variance of the gradient estimator as optimzation proceeds.

### 1.5.2 Contribution to Society

In recent years, the phenomena that we want to represent via models have become more complex, which has made it more difficult to derive and implement inference algorithms. To reduce the burden, automatic-inference tools such as Stan [9] or Edward [86] have developed, which infer the model we constructed automatically without implementing the learning algorithm. Among them, an approximate inference method which I introduced in Section 1.3 has attracted attention in recent years. This tool helps users focus on model construction by taking the hassle out of implementing inference algorithms each time.

Thanks to these tools, the flexibility of constructing the model is improved. However, there is a significant problem that it took much time for the convergence of the inference. Due to this, we have difficulty to perform trial-and-error analysis of the model in a short period of time. One of the reasons why this problem occurs is *dependency of the variance of the stochastic gradient on the inference accuracy and convergence* [68, 80, 70, 7].

My research contributes to this part. The proposed method can reduce the variance of gradient estimator more than the sampling-based *state-of-the-art* variance reduction method [7]. Therefore, my method makes the speed of convergence faster than the other methods and gives us more time to do trial and error to seek the proper model. Through this contribution, many physiological burden for modeling will be removed, and the statistical machine learning will more and more widely spread in society.

## 1.6 Organization of This Thesis

The rest of this thesis is organized as follows.

In Chapter 1, we introduce the concept of Bayesian inference, variational inference, and Monte Carlo variational inference to which we mainly contribute. Furthermore, we explain the contribution and the organization of this thesis briefly.

In Chapter 2, we concretely summarize the background method with its concept and variance reduction research, which is one of the crucial research fields on variational inference.

Through this chapter, we explain the reason why the research we worked on is necessary. In addition, we introduce the related work, and compare the proposed method with them.

We illustrate the proposed method, including the idea, central concept, algorithm, and its derivation in Chapter 3. Here, we explain how the MLMC method and Monte Carlo variational inference is combined. Besides, it is shown that how the number of samples is adapted for stochastic gradient estimation on the proposed method. Furthermore, in Section 3.4, we reveal theoretical properties in the proposed framework. Firstly, we analyze the convergence of the weighted averaging gradient on the basis of stochastic gradient descent with a learning-rate scheduler. Secondly, we evaluate the quality of the gradient estimator on the basis of the *signal-to-noise* ratio. These theoretical results on the proposed method are compared with the naive baseline method and the *state-of-the-art* method from the gradient variance reduction perspective. The proofs of these theoretical analyses are represented in Chapter 4.

Through several experiments on the real-world data sets, we evaluate the proposed method by comparing it with sampling-based baseline methods in Chapter 5. Here, we find that the proposed method achieves faster convergence than the existing methods, and gets closer to the optimum in several experiments. Moreover, we confirm the proposed method reduces the variance of gradient estimator more than the other methods do.

Finally, we describe the conclusion of this thesis and mention the future prospects in Chapter 6.

We show the relationship between each chapter and section in Figure 1.2.

# Chapter 2

# Background and Related Work

In this chapter, we explain the background of the proposed method and introduce the related work. First, we explain the preliminaries of variational inference, the reparameterized gradient, and Monte Carlo variational inference in Section 2.1. Moreover, we describe the problem which we focus on thorough this thesis in Section 2.1.4. Finally, we introduce the related work, including the basic variance reduction methods such as control variates and importance sampling in Section 2.2. For readability, we summarize the notation, which is mainly used in this thesis, in Section 2.3.

## 2.1 Background

In this section, we briefly summarize variational inference in section 2.1.1. Next, we introduce the gradient estimator based on the reparameterized gradient that is often used for complex model or reducing the variance of gradient estimator. In addition, we overview Monte Carlo variational inference (MCVI) based on the reparameterized gradient. Finally, we explain the variance problem on MCVI, which we mainly focus on.

### 2.1.1 Variational Inference

VI constructs an approximation by minimizing the Kullback-Leibler (KL) divergence between $p(\mathbf{z}|\mathbf{x})$ and a variational distribution $q(\mathbf{z}|\lambda)$,

$$\operatorname*{argmin}_{\lambda \in S} \mathrm{KL}(q(\mathbf{z}|\lambda) \| p(\mathbf{z}|\mathbf{x})),$$

where

$$\mathrm{KL}(q(\mathbf{z}|\lambda) \| p(\mathbf{z}|\mathbf{x})) = \int q(\mathbf{z}|\lambda) \log \frac{q(\mathbf{z}|\lambda)}{p(\mathbf{z}|\mathbf{x})} \mathrm{d}\mathbf{z},$$

and $\lambda \in \mathbb{R}^d$ is a single vector of all free parameters and $d$ is the dimension of the parameter space $S$ (see Figure 2.1). However, this objective cannot be computed because it can be expressed as

$$\begin{aligned} \mathrm{KL}(q(\mathbf{z}|\lambda) \| p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_{q(\mathbf{z}|\lambda)}[\log q(\mathbf{z}|\lambda) - \log p(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}_{q(\mathbf{z}|\lambda)}[\log q(\mathbf{z}|\lambda) - \log p(\mathbf{x}, \mathbf{z})] + \log p(\mathbf{x}), \end{aligned}$$

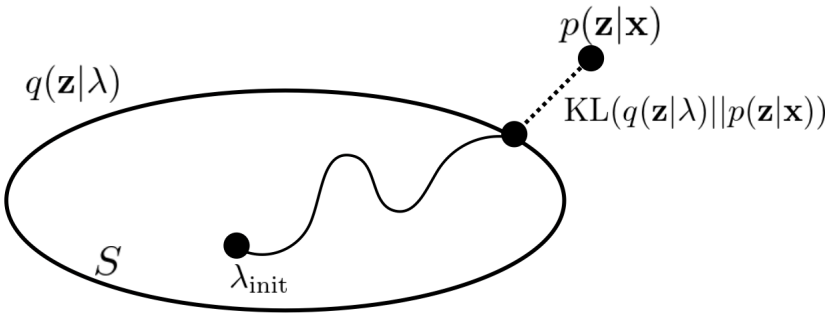


**Figure 2.1:** Schematic image of variational inference [3].

and the $\log p(\mathbf{x})$ term appears which is unknown. Therefore, we optimize an alternative objective function and that is equivalent to minimizing the KL divergence. Because of the non-negativity of KL, we can derive the lower bound as

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|\lambda)}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}|\lambda)],$$

which is called the evidence lower bound (ELBO) [44]. Thus, minimizing the KL divergence is the same as maximizing the ELBO:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\mathbf{z}|\lambda)}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}|\lambda)]. \tag{2.1}$$

When VI is applied for large-scale data or a complex model, however, it is hard, or even sometimes impossible, to compute the differentiation of the objective (2.1) with respect to $\lambda$ directly. One way to handle this problem is to use a stochastic gradient on the basis of two major gradient estimators: the score function gradient estimator [68] and the reparameterized gradient estimator [66, 48, 70]. These gradient estimators are obtained by approximating the expectation of the gradient of (2.1) with independent and identically distributed (i.i.d.) samples from $q(\mathbf{z}|\lambda)$. However, the score function gradient estimator tends to be noisy because of sample variance, which can negatively affect the accuracy of gradient estimation. Therefore, we focus on the reparameterized gradient estimator, which is a useful way to reduce the variance less than that of the score function gradient estimator.

### 2.1.2 Gradient Estimator based on Reparameterization Trick

A reparameterized gradient is a notable approach for learning complex models or reducing the estimation variance based on the reparameterization trick [48]. In this gradient, the variable $\mathbf{z}$ generated from the distribution $q(\mathbf{z}|\lambda)$ is expressed as a deterministic transformation $\mathcal{T}(\cdot)$ of another simple distribution $p(\epsilon)$ over a noise variable $\epsilon$. Therefore, $\mathbf{z}$ can be expressed as $\mathbf{z} = \mathcal{T}(\epsilon; \lambda)$ where $\epsilon \overset{\text{i.i.d.}}{\sim} p(\epsilon)$. We often use an affine transformation $\mathcal{T}(\epsilon; \lambda) = \mathbf{m} + \mathbf{v}\epsilon$ [83], when $\lambda = (\mathbf{m}, \mathbf{v})$. If we set $p(\epsilon)$ as the standard normal Gaussian $\mathcal{N}(0, I_d)$, for example, $\mathcal{T}(\epsilon; \lambda)$ is equal to samples from $\mathcal{N}(\mathbf{m}, \mathbf{v}^\top \mathbf{v})$. By using the reparameterization trick, the gradient of the ELBO can be expressed as the expectation with respect to $p(\epsilon)$ instead of $q(\mathbf{z}|\lambda)$:

$$\nabla_\lambda \mathcal{L}(\lambda) = \mathbb{E}_{p(\epsilon)}[\nabla_\lambda \log p(\mathbf{x}, \mathcal{T}(\epsilon; \lambda)) - \nabla_\lambda \log q(\mathcal{T}(\epsilon; \lambda)|\lambda))]. \tag{2.2}$$

Thus, the distribution needed for the expectation is fixed, and the gradient estimator is obtained by approximating the expectation with i.i.d. random variable $\epsilon$ from $p(\epsilon)$.

### 2.1.3 Monte Carlo Variational Inference (MCVI)

In the general MCVI framework, the gradient of the ELBO is represented as an expectation $\nabla_\lambda \mathcal{L}(\lambda) = \mathbb{E}[g_\lambda(\tilde{\mathbf{z}})]$ over a random variable $\tilde{\mathbf{z}}$, where $g_\lambda(\cdot)$ is a function of the gradient of Eq.(2.1). For the reparameterization estimator, Eq.(2.2) with $\tilde{\mathbf{z}} = \epsilon$ leads to the expression, $\nabla_\lambda \mathcal{L}(\lambda) = \mathbb{E}_{p(\epsilon)}[g_\lambda(\epsilon)]$, where

$$g_\lambda(\epsilon) = \nabla_\lambda \log p(\mathbf{x}, \mathcal{T}(\epsilon; \lambda)) - \nabla_\lambda \log q(\mathcal{T}(\epsilon; \lambda)|\lambda)). \tag{2.3}$$

To estimate this gradient stochastically, we use an *unbiased* estimator calculated by averaging over i.i.d. samples $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_N\}$:

$$\widehat{\nabla}_{\lambda_t} \mathcal{L}(\lambda_t) = \hat{g}_{\lambda_t}(\epsilon_{1:N}) = \frac{1}{N} \sum_{n=1}^{N} g_{\lambda_t}(\epsilon_n),$$

where $t$ represents the optimization step. The ELBO can then be optimized on the basis of $\hat{g}_{\lambda_t}(\epsilon_{1:N})$ by using some form of stochastic optimization (e.g., stochastic gradient descent (SGD) [73], AdaGrad [20], and Adam [47]). For example, optimization can be performed by iterating SGD updates with a decreasing learning rate $\alpha_t = \alpha_0 \eta_t$:

$$\lambda_{t+1} = \lambda_t - \alpha_t \hat{g}_{\lambda_t}(\tilde{\mathbf{z}}).$$

Here, $\alpha_0$ is the initial value of the learning rate, and $\eta_t > 0$ is the value of the learning-rate scheduler which is defined in Section 3.6.

### 2.1.4 Variance Problem on MCVI

Since MCVI was introduced, VI has become a useful way of handling various model architecture and coping with scalable inference on big data [95]. However, it has a crucial problem that the convergence of the stochastic optimization scheme tends to be slow when the magnitude of the variance of the gradient estimator becomes high because of the Monte Carlo estimation. Buchholz et al. [7] showed the convergence of optimization on the basis of the Randomized Quasi-Monte Carlo (RQMC) method [35, 82, 63, 55, 15] and SGD with fixed learning rate depends on the variance of the stochastic gradient estimator.

**Theorem 2.1** (Buchholz et al. [7])**.** *Let $F$ be a function with Lipschitz continuous derivatives, i.e., there exists $L > 0$ s.t. $\forall \lambda, \bar{\lambda} \|\nabla F(\lambda) - F(\bar{\lambda})\|_2^2 \leq L \|\lambda - \bar{\lambda}\|_2^2$. Let $U_N = \{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$ be an sobol sequence [82] and assume $\forall \lambda$, $G : \mathbf{u} \mapsto g_{\Gamma(\mathbf{u})(\lambda)}$ has cross partial derivatives up to order $d$. Further assume the constant learning rate satisfies $\alpha < 2/L$ and let $\mu = 1 - \alpha L/2$. Then $\forall \lambda$, $\mathrm{trVar}_{U_N}[\hat{g}_N(\lambda)] \leq M_V \cdot r(N)$, where $M_V < \infty$ and $r(N) = \mathcal{O}(N^{-2})$,*

$$\frac{\sum_{t=1}^{T} \mathbb{E}[\|\nabla F(\lambda_t)\|_2^2]}{T} \leq \frac{1}{2\mu} \alpha L M_V r(N) + \frac{F(\lambda_1) - F(\lambda^*)}{\alpha \mu T},$$

*where $\lambda_t$ is iteratively defined in the SGD update-rule as $\lambda_{t+1} = \lambda_{t+1} - \alpha \hat{g}_N(\lambda_t)$. Consequently,*

$$\lim_{T \to \infty} \frac{\sum_{t=1}^{T} \mathbb{E}[\|\nabla F(\lambda_t)\|_2^2]}{T} = \frac{1}{2\mu} \alpha L M_V r(N).$$

In the above, $\Gamma$ is a transforming function from the RQMC sequence to the original random variables from the target distribution. These results underline that the sum of the

**Table 2.1:** Relationship between previous work and this work. "CV" stands for control variates, "RB" for Rao-Blackwellization, and "IS" for importance sampling. Denote by $N$ the number of random variables from $p(\epsilon)$, and $t$ is an iteration step. "SF" means a score function, and "RG" stands for a reparameterized gradient. Denote by $N_t$ and $\eta_t$ the estimated sample size adaptively and the learning-rate scheduler function in iteration step $t$. "SNR" stands for *signal-to-noise* ratio.

| | Method | Order of variance | Gradient estimator | Sample size | Convergence analysis | SNR analysis |
|---|---|---|---|---|---|---|
| Ranganath et al. [68] | CV & RB | $\mathcal{O}(N^{-1})$ | SF | Fixed | - | - |
| Ruiz et al. [78] | IS | $\mathcal{O}(N^{-1})$ | SF | Fixed | - | - |
| Roeder et al. [76] | Stop Gradient | $\mathcal{O}(N^{-1})$ | RG | Fixed | - | - |
| Miller et al. [60] | CV | $\mathcal{O}(N^{-1})$ | RG | Fixed | - | - |
| Sakaya and Klami [80] | IS | $\mathcal{O}(N^{-1})$ | SF & RG | Fixed | - | - |
| Li et al. [56] | Adaptive IS | $\mathcal{O}(N^{-1})$ | SF | Fixed | - | - |
| Buchholz et al. [7] | RQMC | $\mathcal{O}(N^{-2})$ | SF & RG | Fixed | ✓(fixed learning rate) | - |
| This work | MLMC | $\mathcal{O}(\eta_{t-1}^2 N_t^{-1})$ | RG | Adaptive | ✓(learning-rate scheduling) | ✓ |

gradient norms depends on the variance of gradient estimators [7]. Therefore, as the scale of the variance of gradient estimators is reduced, we can get the results that are closer to the optimum when the optimization converges. Thus, the variance reduction of the stochastic gradient is a crucial problem on the performance of MCVI.

## 2.2 Related Work

In the MCVI context, many techniques have been proposed for variance reduction, such as control variates [32], Rao-Blackwellization [68], importance sampling [78, 8, 80, 56], and any others [84, 76]. Furthermore, since Kingma and Welling [48] and Rezende et al. [70] introduced the reparameterization trick, the reparameterized gradient has also been extended or applied, e.g., the generalized reparameterized gradients [79], control variates on reparameterized gradients [60], and the doubly reparameterized gradient [88]. These methods have been combined with the classical variance reduction methods in the above.

Here, we introduce the preliminaries and recent works of three major approaches of variance reduction, i.e., the control variate, importance sampling, and low-variance sampling approaches. In addition, we define the signal-to-noise ratio and review the related work which used it. Finally, we summarize the relationship between previous work and our work in Table 2.1.

### 2.2.1 Control Variates

One of the classic methods to reduce the variance of Monte Carlo sampling is using the control variates method [32]. When we want to estimate $\mathbb{E}[f]$ and there is a function $h$ which is correlated to $f$ with a known expectation $\mathbb{E}[h]$, we can use an unbiased estimator for $\mathbb{E}[f]$ based on $N$ i.i.d samples $\omega^{(n)}$ as follows:

$$\hat{f}(\omega^{(n)}) = N^{-1} \sum_{n=1}^{N} \{f(\omega^{(n)}) - a(h(\omega^{(n)}) - \mathbb{E}[h])\}. \tag{2.4}$$

Then, the variance of $\hat{f}(\omega^{(n)})$ is expressed as

$$\mathbb{V}[\hat{f}(\omega^{(n)})] = \mathbb{V}[f(\omega^{(n)})] + a^2 \mathbb{V}[h(\omega^{(n)})] - 2a\mathrm{Cov}(f(\omega^{(n)}), h(\omega^{(n)})),$$

and the optimal value for $a$ is

$$a^* = \rho \cdot \sqrt{\frac{\mathbb{V}[f]}{\mathbb{V}[h]}},$$

where $\rho$ is the correlation between $f$ and $h$. Therefore, the variance of this estimator is reduced by factor $1 - \rho^2$ (see Giles [26]).

In MCVI context, the function $f(\omega)$ is equal to $g_\lambda(\epsilon)$ defined in Eq. (2.3). Ranganath et al. [68] has firstly proposed to use CS methods to MCVI, and it has been extended by Miller et al. [60] on the basis of the reparameterized gradient estimator.

However, it is difficult to chose a function $h$ which is highly correlated to $g_\lambda(\epsilon)$. Moreover, as Buchholz et al. [7] mentioned, CV methods do not archieve the improvement of the $\mathcal{O}(N^{-1})$ rate of the variance of gradient estimator.

### 2.2.2 Importance Sampling (IS)

Importance sampling (IS) [74, 77] is one of bias-correction methods to improve the accuracy of estimators by sampling from a different proposal distribution. When an unbiased estimator such as Monte Carlo estimator is used, a bias is banished. Therefore, it is only necessary to consider how to reduce the estimation variance. The accuracy improvement of an unbiased estimator gets more significant as the number of Monte Carlo samples increases. However, a huge number of samples are needed to reduce the estimation variance; therefore, it is not an effective way in practice [7, 30]. Thus, instead, we consider setting a proposal distribution $r(x)$ as reducing the variance of Monte Carlo estimator. That is, if we want to calcurate the expectation of some function $f(x)$ over $p(x)$, we can introduce $r(x)$ and rewritten as,

$$\begin{aligned}
\mathbb{E}_{p(x)}[f(x)] &= \int p(x) f(x) \mathrm{d}x \\
&= \int r(x) \frac{p(x)}{r(x)} f(x) \mathrm{d}x \\
&= \mathbb{E}_{r(x)}\left[\frac{p(x)}{r(x)} f(x)\right].
\end{aligned}$$

Therefore, we can estimate the expectation by i.i.d. Monte Carlo samples from $r(x)$ as a substitute for $p(x)$, i.e.,

$$\mathbb{E}_{p(x)}[f(x)] \approx \frac{1}{N} \sum_{i=1}^{N} \frac{p(x_i)}{r(x_i)} f(x_i),$$

where $x_{1:N} \overset{\text{i.i.d.}}{\sim} r(x)$. By doing so, when we set

$$\hat{f}^{\mathrm{IS}}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{p(x_i)}{r(x_i)} f(x_i),$$

the variance of the importance-sampled Monte Carlo estimator can be expressed as

$$\begin{aligned}
\mathbb{V}[\hat{f}^{\mathrm{IS}}(x)] &= \frac{1}{N} \mathbb{E}_{r(x)}\left[\frac{p^2(x)}{r^2(x)} f^2(x)\right] - \frac{1}{N}\left(\mathbb{E}_{r(x)}\left[\frac{p(x)}{r(x)} f(x)\right]\right)^2 \\
&= \frac{1}{N} \mathbb{E}_{r(x)}\left[\frac{p^2(x)}{r^2(x)} f^2(x)\right] - \frac{1}{N}\left(\mathbb{E}_{p(x)}[f(x)]\right)^2.
\end{aligned}$$

According to the fact that the variance of Monte Carlo estimator is expressed as

$$\mathbb{V}[\hat{f}(x)] = \frac{1}{N}\mathbb{E}_{p(x)}[f^2(x)] - \frac{1}{N}\Big(\mathbb{E}_{p(x)}[f(x)]\Big)^2,$$

where

$$\hat{f}(x) = \frac{1}{N}\sum_{i=1}^{N} f(x_i),$$

the importance-sampled Monte Carlo estimator can reduce the variance if we set $r(x)$ larger than $p(x)$. In MCVI context, the function $f(x)$ and $p(x)$ is equal to $g_\lambda(\epsilon)$ defined in Eq. (2.3) and $p(\epsilon)$.

Ruiz et al. [78] has firstly introduced the IS method to the MCVI context, and it has been extended by Sakaya and Klami [80] for the reparameterized gradient-based MCVI. Moreover, Li et al. [56] has investigated the adaptive version of IS by using moment matching.

However, IS methods still have the difficulty of adequately constructing a proposal distribution $r(x)$ for MCVI. Additionally, as well as CV methods, IS methods can not improve the $\mathcal{O}(N^{-1})$ rate of variance of gradient estimator, which is equal to that of ordinary Monte Carlo estimator [7].

### 2.2.3 Low-Variance Sampling Approach

Although many low-variance-sampling methods have been proposed in the MCMC context such as [58, 24], the idea of using more sophisticated Monte Carlo sampling to reduce the variance of the estimator in the MCVI framework has only been investigated recently. The object of this framework is to improve the $\mathcal{O}(N^{-1})$ rate of the variance of the gradient estimator. Ranganath et al. [68] and Ruiz et al. [78] suggested using quasi-Monte Carlo (QMC), and Tran et al. [87] applied it to a specific model. Recently, Buchholz et al. [7] proposed a variance reduction method by using randomized QMC (RQMC), which can achieve, in the best case, the $\mathcal{O}(N^{-2})$ rate of the variance in the MCVI framework (called QMCVI). However, it is known that the estimations made with the QMC-based method are sometimes worse than those made with MC methods because of a potentially bad interaction between the underlying deterministic points and the function to be estimated [54].

### 2.2.4 Signal-to-Noise Ratio (SNR)

Signal-to-Noise ratio (SNR) is difined as the absolute value of the expected estimator scaled by its standard derivation. In the MCVI context, it is defined as follows:

**Definition 2.1** (Signal-to-Noise Ratio for Gradient Estimator)**.**

$$\mathrm{SNR}(\lambda) = \frac{\|\mathbb{E}_{p(\epsilon_{1:N})}[\hat{g}_\lambda(\epsilon_{1:N})]\|_2^2}{\sqrt{\mathbb{V}[\hat{g}_\lambda(\epsilon_{1:N})]}}.$$

SNR provides a measure of the quality of a gradient estimator. Though a high SNR does not always indicate a good stochastic optimization scheme (as the target objective itself might be poorly chosen), a low SNR is always problematic as it indicates that the gradient estimator is dominated by noise: if $\mathrm{SNR} \to 0$, then the estimator become completely random [67]. Recently, Rainforth et al. [67] analyzed the behavior of an importance-weighted

stochastic gradient in terms of SNR and revealed the differences in the effect of increasing the number of importance weights between inference and generative networks in the variational auto-encorder [48]. In addition, there have been several theoretical and empirical analyses of stochastic gradient estimators by using SNR [39, 53, 81, 88].

## 2.3 Notation

Finally, we show the notation and symbols used in the rest of this thesis in Table 2.2.

**Table 2.2:** List of symbols used in the main text.

| Symbol | Meaning |
|---|---|
| $\mathbf{x}$ | The observed data-point |
| $\mathbf{z}$ | The vectors of latent variables or weight variables |
| $\lambda$ | The parameter of variational distribution |
| $\epsilon$ | The random variables from $p(\epsilon)$ |
| $\epsilon_{1:N}$ | The $N$ set of $\epsilon$: $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_N\}$ |
| $\mathcal{T}(\cdot)$ | The deterministic transformation function |
| $\alpha_0$ | The initial value of learning-rate |
| $\eta_t$ | The learning-rate schedular function at the optimization step $t$ |
| $\mathbb{V}_t$ | The *one-sample* variance of gradient estimator at the optimization step $t$ |
| $C_t$ | The *one-sample* cost of gradient estimator at the optimization step $t$ |
| $p(\mathbf{x}, \mathbf{z})$ | The joint distribution of $\mathbf{x}$ and $\mathbf{z}$ |
| $p(\mathbf{z}\|\mathbf{x})$ | The posterior of $\mathbf{z}$ given $\mathbf{x}$ |
| $p(\mathbf{x}\|\mathbf{z})$ | The likelihood given $\mathbf{z}$ |
| $p(\mathbf{x})$ | The marginal distribution |
| $p(\mathbf{z})$ | The prior of $\mathbf{z}$ |
| $q(\mathbf{z}\|\lambda)$ | The (Gaussian) variational distribution with parameter $\lambda$ |
| $\mathcal{L}(\lambda)$ | The objective function |
| $\nabla_\lambda \mathcal{L}(\lambda)$ | The gradient of the objective function for $\lambda$ |
| $\widehat{\nabla}_\lambda \mathcal{L}(\lambda)$ | The reparameterized gradient estimator by MC or RQMC samples |
| $\widehat{\nabla}^{\mathrm{MRG}}_\lambda \mathcal{L}(\lambda)$ | The multi-level reparameterized gradient estimator by Monte Carlo samples |
| $g_\lambda(\epsilon)$ | The inside of reparameterized gradient of ELBO |
| $\mathbb{R}$ | The set of real numbers |
| $\mathbb{N}$ | The set of natural numbers |
| $\mathcal{O}$ | Landau's asymptotic notation |
| $S$ | The parameter space |
| $d$ | The number of dimension on parameter space $S$ |
| $\mathrm{KL}(q\|p)$ | Kullback-Leibler divergence between $q$ and $p$ |
| $\lceil x \rceil$ | The ceil function where $\lceil x \rceil = \min\{k \in \mathbb{N} \| x \leq k\}$ |
| $\top$ | The transpose |
| $\|\cdot\|_2$ | The $l_2$ Euclidean norm of vectors |
| $\mathbb{E}[\cdot]$ | The expectation |
| $\mathbb{V}[\cdot]$ | The variance |

# Chapter 3

# Multi-level Monte Carlo Variational Inference

In this chapter, we explain the proposed method, called MLMCVI. First, we summerize the multi-level Monte Carlo (MLMC) method in Section 3.1. Next, we derive MLMCVI in Sections 3.2 and 3.3. Finally, we theoretically analyze the proposed method in Section 3.4.

## 3.1 Multi-level Monte Carlo Method

The multi-level Monte Carlo (MLMC) method was proposed by Heinrich [38]. This method has been often used in stochastic differential equations for option pricing [26, 11, 71]. In statistics, there are many applications in approximate Bayesian computation [31, 43, 92].

Because of the linearity of expectations, given a sequence $P_0, P_1, \ldots, P_{L-1}$ which approximates $P_L$ with increasing accuracy, we have the simple identity:

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{l=1}^{L} \mathbb{E}[P_l - P_{l-1}]. \tag{3.1}$$

We can thus use the following unbiased estimator for $\mathbb{E}[P_L]$,

$$\mathbb{E}[P_L] \approx N_0^{-1} \sum_{n=1}^{N_0} P_0^{(0,n)} + \sum_{l=1}^{L} \left\{ N_l^{-1} \sum_{n=1}^{N_l} (P_l^{(l,n)} - P_{l-1}^{(l,n)}) \right\}, \tag{3.2}$$

with the inclusion of $l$ in $(l, n)$ indicating that independent samples are used at each level of correction.

If we define $V_0, C_0$ to be the variance and cost of *one sample* of $P_0$, and $\mathbb{V}_l, C_l$ to be the variance and cost of *one sample* of $P_l - P_{l-1}$, then the total variance and cost of Eq.(3.2) are $\sum_{l=0}^{L} N_l^{-1} \mathbb{V}_l$ and $\sum_{l=0}^{L} N_l C_l$, respectively. The MLMC method is described in detail in Heinrich [38] and Giles [26, 28].

Thus, if $Y$ is a multi-level estimator given by

$$Y = \sum_{l=0}^{L} Y_l, \;\; Y_l = N_l^{-1} \sum_{n=1}^{N_l} (P_l^{(l,n)} - P_{l-1}^{(l,n)}),$$

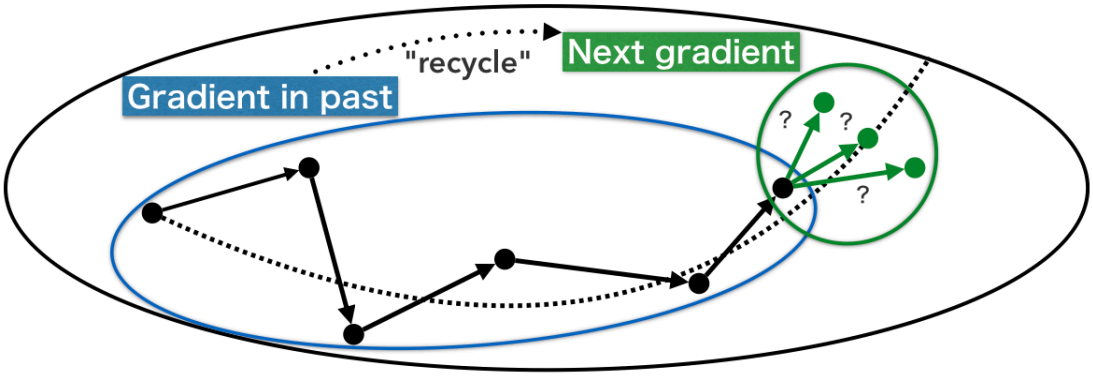


**Figure 3.1:** Concept of the proposed method.

with $P_{-1} \equiv 0$, then

$$\mathbb{E}[Y] = \mathbb{E}[P_L],\ V[Y] = \sum_{l=0}^{L} N_l^{-1} \mathbb{V}_l,\ \mathbb{V}_l \equiv V[P_l - P_{l-1}].$$

This method is widely used, for example, in the infinite-dimensional integration of stochastic differential equations arising in mathematical finance [26] and the large-cost problem of solving elliptic partial differential equations with random coefficients [11].

In a Bayesian framework, Giles et al. [31, 30] applied MLMC to stochastic gradient MCMC algorithms such as the stochastic gradient Langevin dynamics (SGLD), which discretize the stochastic differential equation (SDE) of a posterior based on the multi-level step size and couple them.

## 3.2 Key Idea of Multi-level Monte Carlo Variational Inference (MLMCVI)

The key idea of MLMCVI is to construct a low-variance gradient estimator by using *the information we get as the optimization proceeds* by "recycling" the parameter and gradient from the past (see Figure 3.1).

Equation (2.2) implies that a reparameterized gradient can be applied in the MLMC framework because the expectation always depends on a fixed distribution $p(\epsilon)$, and the linearity of the expectation is available. Moreover, another idea of MLMCVI is that it regards the level "$l$" as the number of iterations "$t$." Thus, applying these ideas, we can "recycle" the parameter and the gradient in the past. When we set,

$$g_{\lambda_t}(\epsilon) = \nabla_{\lambda_t} \log p(\mathbf{x}, \mathcal{T}(\epsilon; \lambda_t)) - \nabla_{\lambda} \log q(\mathcal{T}(\epsilon; \lambda_t)|\lambda_t)),$$

the multi-level reparameterized gradient (MRG) in iteration $T$ is expressed as

$$\nabla_{\lambda_T}^{\mathrm{MRG}} \mathcal{L}(\lambda_T) = \mathbb{E}_{p(\epsilon)}\left[g_{\lambda_0}(\epsilon)\right] + \sum_{t=1}^{T} \left(\mathbb{E}_{p(\epsilon)}\left[g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)\right]\right), \tag{3.3}$$

where $t \in \mathbb{N}$ and $T \in \mathbb{N}$. In the MCVI framework, we need an unbiased estimator of the gradient for stochastic optimization. An unbiased estimator of the MRG in Eq.(3.3) can be immediately obtained as

$$\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_T}\mathcal{L}(\lambda_T) = N_0^{-1}\sum_{n=1}^{N_0} g_{\lambda_0}(\epsilon_{(n,0)}) + \sum_{t=1}^{T}\left(N_t^{-1}\sum_{n=1}^{N_t}\left[g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)})\right]\right), \tag{3.4}$$

where $N_t$ $(t = 0, 1, \ldots, T)$ is the sample size in each iteration, and the MRG estimator is unbiased for $\nabla_{\lambda_T}\mathcal{L}(\lambda_T)$.

## 3.3 Algorithm Derivation

How should the optimal different sample size $N_t$ be estimated for the MRG estimator? To answer this question, we derive the optimal sample size to minimize the *total* variance of the MRG estimator. To derive the optimal $N_t$ properly, we apply the following definition and assumption.

**Definition 3.1** (One-sample Complexity and Variance)**.** *Let $C_t$ and $\mathbb{V}_t$ be the* one-sample *computational complexity and variance of $g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)})$ in iteration $t \geq 1$, respectively.*

Here, if $t = 0$, we define $C_0$ and $\mathbb{V}_0$ as the *one-sample* computational complexity and variance of $g_{\lambda_0}(\epsilon_{(n,0)})$, respectively.

**Assumption 3.1** (Computational Complexity of One-sample Gradient Calculation)**.** *The cost of calculating the one-sample gradient is constant, i.e., $C_0 = c$ and $C_t = 2c$ $(t \geq 1)$, where $c$ is a positive constant.*

Then, the overall cost and variance of $\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_T}\mathcal{L}(\lambda_T)$ are $\sum_{t=0}^{T} N_t C_t$ and $\sum_{t=0}^{T} N_t^{-1}\mathbb{V}_t$, respectively. Assumption 3.1 means that the computational complexity of calculating a gradient estimator per level is $cN_0$ or $2cN_t$ $(t \geq 1)$ when including the sampling cost. From Assumption 3.1, we can establish the following theorem according to standard proof techniques in Giles [26].

**Theorem 3.1** (Optimal Sample Size $N_t$)**.** *Suppose that Assumption 3.1 is satisfied. Then, the total variance is minimized by choosing the sample size per level as*

$$N_t = \begin{cases} \sqrt{\frac{\mathbb{V}_1}{2\mathbb{V}_0}}N_0 & (t = 1), \\ \sqrt{\frac{\mathbb{V}_t}{\mathbb{V}_{t-1}}}N_{t-1} & (t = 2, 3, \ldots, T). \end{cases} \tag{3.5}$$

*Sketch of Proof.* This theorem can be proved by solving a constrained optimization problem that minimizes the overall variance as: $\min_{N_t}\sum_{t=0}^{T} N_t^{-1}\mathbb{V}_t$ s.t. $\sum_{t=0}^{T} N_t C_t = M$, where $M$ is a positive constant value for the total sampling cost. The complete proof is given in Section 4.1. □

Theorem 3.1 indicates that the optimal sample size can be estimated by the ratio of the previous *one-sample* variance to the current one. Therefore, the sample size is determined as

$$N_t = \begin{cases} \lceil\sqrt{\frac{\mathbb{V}_1}{2\mathbb{V}_0}}N_0\rceil & (t = 1), \\ \lceil\sqrt{\frac{\mathbb{V}_t}{\mathbb{V}_{t-1}}}N_{t-1}\rceil & (t = 2, 3, \ldots, T), \end{cases} \tag{3.6}$$

**Algorithm 1** Multi-level Monte Carlo Variational Inference

**Require:** Data $\mathbf{x}$, random variable $\epsilon \sim p(\epsilon)$, transform $\mathbf{z} = \mathcal{T}(\epsilon; \lambda)$, model $p(\mathbf{x}, \mathbf{z})$, variational family $q(\mathbf{z}|\lambda)$
**Ensure:** Variational parameter $\lambda^*$
**Initialize:** $N_0$, $\lambda_0$, $\alpha_0$ and hyperparameter of $\eta$
**for** $t = 0$ to $T$ **do**
  **if** $t = 0$ **then**
    $\epsilon_n \sim p(\epsilon)$ $(n = 1, 2, \ldots, N_0)$ ◁ sampling $\epsilon$
    $\hat{g}_{\lambda_0}(\epsilon_{1:N_0}) = N_0^{-1} \sum_{n=1}^{N_0} g_{\lambda_0}(\epsilon_n)$ ◁ calc. RG estimator
    $\lambda_1 = \lambda_0 - \alpha_0 \hat{g}_{\lambda_0}(\epsilon_{1:N_0})$ ◁ grad-update
  **else**
    **sampling** one $\epsilon$ for sample-size estimation
    **estimate** $N_t$ by Eq.(3.6)
    $\epsilon_n \sim p(\epsilon)$ $(n = 1, 2, \ldots, N_t)$ ◁ sampling $\epsilon$
    $\hat{g}'_{\lambda_t}(\epsilon_{1:N_t}) = N_t^{-1} \sum_{n=1}^{N_t} [g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)})]$ ◁ calc. multi-level term
    $\lambda_{t+1} = \lambda_t + \frac{\eta_t}{\eta_{t-1}}(\lambda_t - \lambda_{t-1}) - \alpha_t \hat{g}'_{\lambda_t}(\epsilon_{1:N_t})$ ◁ grad-update
    **if** $\lambda_{t+1}$ is converged to $\lambda^*$ **then**
      **break**
    **end if**
  **end if**
**end for**
**return** $\lambda^*$

where $\lceil x \rceil = \min\{k \in \mathbb{N} | x \leq k\}$ because $N_t$ is a natural number.

The magnitude of $\mathbb{V}_t$ is a critical issue for sample-size estimation. The behavior of the estimated sample size is analyzed in Section 3.4.2.

The MRG has a problem in that the total cost can become crucially large as $t$ goes to infinity. To bypass this problem, we consider another formulation of the MRG estimator and update-rule on the basis of SGD.

**Lemma 3.1** (Another Formulation of MRG Estimator)**.** *The MRG estimator in iteration $t \geq 1$ can be represented as*

$$\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_t} \mathcal{L}(\lambda_t) = \widehat{\nabla}^{\mathrm{MRG}}_{\lambda_{t-1}} \mathcal{L}(\lambda_{t-1}) + N_t^{-1} \sum_{n=1}^{N_t} \left[ g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)}) \right],$$

*and the update-rule for this estimator in SGD is*

$$\lambda_{t+1} = \lambda_t + \frac{\eta_t}{\eta_{t-1}}(\lambda_t - \lambda_{t-1}) - \alpha_t N_t^{-1} \sum_{n=1}^{N_t} \left[ g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)}) \right].$$

*Proof.* See Section 4.2 for the proof. □

From Theorem 3.1 and Lemma 3.1, the MLMCVI algorithm is derived as Algorithm 1.

## 3.4 Theoretical Analysis

In this section, we analyze the estimated sample size and the effect of the proposed method on the basis of the weighted average norm of the gradient and SNR. In addition, we

compare the results with sampling-based methods such as MCVI and QMCVI [7].

### 3.4.1 Assumptions

To analyze this, we set the following assumptions, which are often considered in the MCVI context [7, 18, 94].

**Assumption 3.2** (Diagonal Gaussian Variational Distribution)**.** *The variational distribution, $q(\mathcal{T}(\epsilon;\lambda)|\lambda)$, is the Gaussian distribution with mean vector* $\mathbf{m}$ *and diagonal covariance matrix* $\Sigma = diag(\mathbf{v})$.

**Assumption 3.3** (Lipschitz Continuity on $\nabla_\lambda \mathcal{L}(\lambda)$)**.** *ELBO $\mathcal{L}(\lambda)$ is a function with Lipschitz continuous derivatives, i.e., $\exists K_1 > 0$ s.t. $\forall \lambda, \bar{\lambda}$:*

$$\|\nabla_\lambda \mathcal{L}(\lambda) - \nabla_{\bar{\lambda}} \mathcal{L}(\bar{\lambda})\|_2^2 \leq K_1 \|\lambda - \bar{\lambda}\|_2^2.$$

Assumption 3.2 has been extensively used in several VI frameworks with a stochastic gradient [48, 70]. Assumption 3.3 means that the ELBO value can not change too fast as $\lambda$ changes.

Furthermore, we assume that the boundedness of $\mathcal{T}(\epsilon;\lambda)$.

**Assumption 3.4** (Boundedness of $\mathcal{T}(\epsilon;\lambda)$)**.** *The reparameterized random variable $\mathcal{T}(\epsilon;\lambda)$ is bounded, i.e., $\exists K_2 > 0$ s.t. $\forall \lambda$ :*

$$\|\mathcal{T}(\epsilon;\lambda)\|_2^2 \leq K_2.$$

Of course, Assumption 3.4 does not always hold. However, we can make it hold by truncating $\mathcal{T}(\epsilon;\lambda)$ to a particular value. One way to truncate $\mathcal{T}(\epsilon;\lambda)$ is to use the proximal operator [61]. The details are described in Section 3.5.

In addition, we focus on the SGD update-rule with decreasing learning rate $\alpha_t$, which is expressed as $\lambda_{t+1} = \lambda_t - \alpha_t \hat{g}_{\lambda_t}(\epsilon_{1:N})$ where $\alpha_t = \alpha_0 \eta_t$ and $\eta_t$ is a learning-rate scheduler. The learning-rate scheduling is essential to guarantee the convergence of stochastic optimization when the gradient estimator is noisy [5]. Moreover, we often assume the following condition on $\alpha_t$ [73].

**Assumption 3.5** (Robbins-Monro Condition)**.** *The learning rate $\alpha_t$ satisfies the following conditions: $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$.*

### 3.4.2 Sample-size Analysis

Here, we theoretically investigate the behavior of the estimated sample size. We first give the following proposition for the *one-sample* gradient variance under certain assumptions.

**Proposition 3.1** (Order of One-sample Gradient Variance)**.** *Suppose that Assumptions 3.2, 3.4 and 3.5 are hold. Then, the expectation of the $l_2$-norm of $g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)$ in iteration $t$ ($t \geq 1$) is bounded:*

$$\mathbb{E}_{p(\epsilon)}\left[\|g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)\|_2^2\right] \leq \alpha_{t-1}^2 N^{-1} K_2 (C_1 + d\delta C_2),$$

*where $\delta, C_1, C_2$ are positive constants and $N$ is the sample size for gradient estimation.*

*Proof.* See Section 4.3 for the proof. □

According to Proposition 3.1, the order of the *one-sample* variance $\mathbb{V}_t$ is $\mathcal{O}(\eta_{t-1}^2)$; therefore, $\mathbb{V}_t \overset{t\to\infty}{\to} 0$. Thus, it seems that fewer samples are required to estimate $\mathbb{E}_{p(\epsilon)}[g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)]$ at finer levels. The following lemma shows that the sample size decreases as optimization proceeds.

**Lemma 3.2** (Behavior of Sample Size)**.** *Suppose that Assumptions 3.1, 3.2, 3.4 and 3.5 hold. Then, the estimated sample size $N_t$ goes to 1 as $t \to \infty$.*

*Proof.* See Section 4.4 for the proof. □

### 3.4.3 Convergence Analysis

Next, we theoretically analyze convergence. Here, we focus on the case in which the number of iteration $t$ is $t \geq 1$ because the MRG estimator appears after the first optimization step.

Before the analysis, we first investigate the order of $\mathbb{V}[\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_t}\mathcal{L}(\lambda_t)]$.

**Lemma 3.3** (Variance of $\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_t}\mathcal{L}(\lambda_t)$)**.** *Suppose that Assumptions 3.1, 3.2, and 3.4 hold. Then, the order of $\mathbb{V}[\widehat{\nabla}^{MRG}_{\lambda_t}\mathcal{L}(\lambda_t)]$ is $\mathcal{O}(\eta_{t-1}^2 N_t^{-1})$.*

*Proof.* See Section 4.5 for the proof. □

From Lemma 3.2, the estimated sample size $N_t$ goes to 1 as optimization proceeds; therefore $\mathbb{V}[\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_t}\mathcal{L}(\lambda_t)]$ goes to 0 because $\eta_t \to 0$ as $t \to \infty$.

In the stochastic optimization literature, Bottou et al. [5] provided comprehensive theorems on the basis of SGD. With the help of those theorems, we can prove the following upper bounds on the norm of the gradient and use them to analyze the effect of the proposed method. We can also compare the upper bounds with those for the MC, RQMC, and MLMC-based methods.

**Theorem 3.2** (Weighted Average Norm of Gradient (MC, RQMC))**.** *Suppose that Assumptions 3.1 and 3.2-3.5 are satisfied. Then, $\forall\lambda_t$, $\mathbb{V}[\hat{g}_{\lambda_t}] \leq \kappa N^{-1}$ (MC) or $\mathbb{V}[\hat{g}_{\lambda_t}] \leq \kappa N^{-2}$ (RQMC), where $\kappa < \infty$ is a positive constant, an upper bound on the norm of the gradient in SGD at iteration $t(= 1, \ldots, T)$ is given as*

$$\frac{1}{A_T}\sum_{t=1}^{T}\alpha_t\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right] \leq \begin{cases} G_T + \frac{\alpha_0^2 K_1}{2A_T}\sum_{t=1}^{T}\eta_t^2\left(\mathbb{E}[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2] + \frac{\kappa}{N}\right) \text{ (MC)}, \\ G_T + \frac{\alpha_0^2 K_1}{2A_T}\sum_{t=1}^{T}\eta_t^2\left(\mathbb{E}[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2] + \frac{\kappa}{N^2}\right) \text{(RQMC)}, \end{cases}$$

*where $A_T = \sum_{t=1}^{T}\alpha_t$, $G_T = \frac{1}{A_T}[\mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*)]$, $\lambda_t$ is iteratively defined in the SGD-update rule, and $\lambda^*$ is the optimal parameter.*

*Proof.* See Section 4.6 for the proof. □

**Theorem 3.3** (Weighted Average Norm of Gradient (MLMC))**.** *Suppose that Assumptions 3.1 and 3.2-3.5 hold. Then, $\forall\lambda_t$, $\mathbb{V}[\widehat{\nabla}^{MRG}_{\lambda_t}\mathcal{L}(\lambda_t)] \leq \kappa\eta_{t-1}N_t^{-1}$, where $\kappa$ is a positive constant, $\kappa < \infty$ and $t \geq 1$, an upper bound on the norm of the gradient in SGD given as*

$$\frac{1}{A_T}\sum_{t=1}^{T}\alpha_t\mathbb{E}[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2] \leq G_T + \frac{\alpha_0^2 K_1}{2A_T}\sum_{t=1}^{T}\eta_t^2\left(\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right] + \frac{\kappa}{N_t}\eta_{t-1}^2\right),$$

*where* $A_T = \sum_{t=1}^{T} \alpha_t$, $G_T = \frac{1}{A_T}[\mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*)]$, $\lambda_t$ *is iteratively defined in the SGD-update rule, and* $\lambda^*$ *is the optimal parameter.*

*Proof.* See Section 4.7 for the proof. □

Theorems 3.2 and 3.3 state that the weighted average norm of the squared gradients converges to zero because of $A_T = \sum_{t=1}^{T} \alpha_t = \infty$ and Assumption 3.5 even if the gradient estimator is noisy. This fact can guarantee that the expectation of the gradient norms of the MC-based, RQMC-based, and MLMC-based methods asymptotically stays around zero. In addition, the difference in convergence speed between these methods depends on the last term in each of these bounds. While the convergence of the MC and RQMC-based methods can be accelerated only by increasing the number of samples, that of the proposed method seems to be accelerated by $N_t^{-1}\eta_{t-1}^2$.

### 3.4.4 SNR Analysis

Finally, we prove a lower bound on the SNR to evaluate the quality of the MC, RQMC, and MLMC-based gradient estimators. The SNR of the gradient estimator is defined in Definition 2.2.4. This indicates that, if SNR $\to 0$, the gradient estimator is dominated by random noise, causing problems with the accuracy of estimation.

**Theorem 3.4** (Signal-to-Noise Ratio Bound)**.** *Suppose that Assumptions 3.1 and 3.2-3.5 hold, and that the expectation of the gradient estimator, the variance of the gradient estimator, and the variances of* $\hat{g}_\lambda(\epsilon_{1:N})$ *are non-zero. Then, we have* $\forall \lambda_t$, $\mathbb{V}[\hat{g}_{\lambda_t}] \leq \kappa N^{-1}$ *(MC),* $\mathbb{V}[\hat{g}_{\lambda_t}] \leq \kappa N^{-2}$ *(RQMC) or* $\mathbb{V}[\widehat{\nabla}_{\lambda_t}^{MRG}\mathcal{L}(\lambda_t)] \leq \kappa \eta_{t-1}^2 N_t^{-1}$ *(MLMC) the upper bound on the SNR in iteration* $t$ *for each method is given by*

$$\mathrm{SNR}(\lambda_t) \geq \begin{cases} \frac{\|\nabla_{\lambda_t}\mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\kappa}} \cdot \sqrt{N} & \text{(MC)}, \\ \frac{\|\nabla_{\lambda_t}\mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\kappa}} \cdot N & \text{(RQMC)}, \\ \frac{\|\nabla_{\lambda_t}\mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\kappa}} \cdot \frac{\sqrt{N_t}}{\eta_{t-1}} & \text{(MLMC)}, \end{cases}$$

*where* $\kappa$ *is a positive constant.*

*Proof.* See Section 4.8 for the proof. □

The above theorem implies that, for the MC and RQMC-based methods, the SNR can be increased only by increasing the initial sample size. It also shows that the SNR value gradually decreases and the gradient estimator is dominated by random noise as the optimization proceeds; that is, the quality of the gradient estimator gradually gets worse because $||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2$ approaches 0. In contrast, in our method, the SNR value can be controlled by not only the sample size but also a factor of the learning-rate scheduler function $\eta_{t-1}$.

## 3.5 Non-Bounded Case in Assumption 3.4

Assumption 3.4 does not always hold because $\mathcal{T}(\epsilon;\lambda)$ itself is not bounded. To overcome this situation, the proximal operator [61] is useful. It is an operator associated with a closed convex function $f$ from a Hilbert space $\mathcal{X}$ to $[-\infty, \infty]$, and is defined as follows:

$$\mathrm{prox}_f(x) = \underset{u \in \mathcal{X}}{\mathrm{argmin}} \left( f(u) + \frac{1}{2}\|u - x\|_2^2 \right).$$

In optimization, the proximal operator has several useful properties such as firmly nonexpansiveness:

$$\|\text{prox}_f(x) - \text{prox}_f(y)\|_2^2 \leq (\text{prox}_f(x) - \text{prox}_f(y))^\top (x - y),$$

where $\forall x, y \in \mathcal{X}$.

In the MLMCVI framework, we want to truncate $\mathcal{T}(\epsilon; \lambda)$ when it becomes too large values to fulfill Assumption 3.4. Thus, we set $f$ as the following indicator function for a some set $S$,

$$\text{I}_S(\mathcal{T}(\epsilon; \lambda)) = \begin{cases} 0 & (\mathcal{T}(\epsilon; \lambda) \in S), \\ +\infty & (\text{otherwise}), \end{cases}$$

where $\text{I}_S(\mathcal{T}(\epsilon; \lambda))$ is closed and convex if $S$ is a closed convex set. From this setting, the proximal operator $f = \text{I}_S$ is the Euclidian projection $P(\cdot)$ on $S$:

$$\begin{aligned} \text{prox}_{\text{I}_S}(\mathcal{T}(\epsilon; \lambda)) &= \underset{u \in S}{\text{argmin}} \, \|u - \mathcal{T}(\epsilon; \lambda)\|_2^2 \\ &= P_S(\mathcal{T}(\epsilon; \lambda)). \end{aligned}$$

From the above and firmly nonexpansiveness in the proximal operator, if $\mathcal{T}^+(\epsilon; \lambda) = \text{prox}_{\text{I}_S}(\mathcal{T}(\epsilon; \lambda))$ and $\mathcal{T}^+(\epsilon; \bar{\lambda}) = \text{prox}_{\text{I}_S}(\mathcal{T}(\epsilon; \bar{\lambda}))$, then,

$$\|\mathcal{T}^+(\epsilon; \lambda) - \mathcal{T}^+(\epsilon; \bar{\lambda})\|_2^2 \leq (\mathcal{T}^+(\epsilon; \lambda) - \mathcal{T}^+(\epsilon; \bar{\lambda}))^\top (\mathcal{T}(\epsilon; \lambda) - \mathcal{T}(\epsilon; \bar{\lambda})),$$

is fulfilled. It implies,

$$\|\mathcal{T}^+(\epsilon; \lambda) - \mathcal{T}^+(\epsilon; \bar{\lambda})\|_2^2 \leq \|\mathcal{T}(\epsilon; \lambda) - \mathcal{T}^+(\epsilon; \bar{\lambda})\|_2^2,$$

from the Cauchy-Schwarz inequality. It means that $\mathcal{T}^+(\epsilon; \lambda)$ is 1-Lipschitz continuous and therefore bounded.

By making use of the proximal operator, we can obtain the samples which do not violate Assumption 3.4. Therefore, all of the theorems and lemmas in this thesis hold in a non-bounded case of $\mathcal{T}(\epsilon; \lambda)$. The MLMCVI algorithm in a non-bounded case is shown in Algorithm 2.

## 3.6 Another Expression of the Estimated Sample Size

From Algorithm 1, we can estimate the number of samples $N_t$ by using Eq. (3.6). However, it is sometimes hard to compute for a high-dimensional model because we have to calculate the variance of two gradients: $g_{\lambda_t}(\epsilon)$ and $g_{\lambda_{t-1}}(\epsilon)$.

One of the ways to bypass this problem is using an upper-bounded value of $N_t$. From the proof of Theorem 3.1, the optimal $N_t$ can be expressed as

$$N_t = \frac{1}{\mu} \sqrt{\frac{\mathbb{V}_t}{C_t}}.$$

From Proposition 3.1, the order of the *one-sample* variance $\mathbb{V}_t$ is $\mathcal{O}(\eta_{t-1}^2)$. Thus, the optimal $N_t$ can be bounded as

$$N_t = \frac{1}{\mu} \sqrt{\frac{\mathbb{V}_t}{C_t}} \leq \frac{1}{\mu} \frac{\eta_{t-1}}{\sqrt{C_t}} < \infty.$$

**Algorithm 2** Multi-level Monte Carlo Variational Inference in Non-bounded Case of $\mathcal{T}(\epsilon;\lambda)$

**Require:** Data $\mathbf{x}$, random variable $\epsilon \sim p(\epsilon)$, transform $\mathbf{z} = \mathcal{T}(\epsilon;\lambda)$, model $p(\mathbf{x},\mathbf{z})$, variational family $q(\mathbf{z}|\lambda)$
**Ensure:** Variational parameter $\lambda^*$
**Initialize:** $N_0$, $\lambda_0$, $\alpha_0$, the hyperparameter of $\eta$, and a convex set $S$
**for** $t = 0$ to $T$ **do**
  **if** $t = 0$ **then**
    $\epsilon_n \sim p(\epsilon)$ $(n = 1, 2, \ldots, N_0)$ ◁ sampling $\epsilon$
    $\mathcal{T}^+(\epsilon_n;\lambda_0) = \text{prox}_{\text{I}_S}(\mathcal{T}(\epsilon_n;\lambda_0))$ ◁ check $\mathcal{T}(\epsilon;\lambda_0)$ value
    $\hat{g}_{\lambda_0}(\epsilon_{1:N_0}) = N_0^{-1}\sum_{n=1}^{N_0} g_{\lambda_0}(\epsilon_n)$ ◁ calc. RG estimator
    $\lambda_1 = \lambda_0 - \alpha_0 \hat{g}_{\lambda_0}(\epsilon_{1:N_0})$ ◁ grad-update
  **else**
    **sampling** one $\epsilon$ for sample-size estimation
    **estimate** $N_t$ by Eq.(3.6)
    $\epsilon_n \sim p(\epsilon)$ $(n = 1, 2, \ldots, N_t)$ ◁ sampling $\epsilon$
    $\mathcal{T}^+(\epsilon_n;\lambda_t) = \text{prox}_{\text{I}_S}(\mathcal{T}(\epsilon_n;\lambda_t))$ ◁ check $\mathcal{T}(\epsilon;\lambda_t)$ value
    $\hat{g}'_{\lambda_t}(\epsilon_{1:N_t}) = N_t^{-1}\sum_{n=1}^{N_t}[g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)})]$ ◁ calc. multi-level term
    $\lambda_{t+1} = \lambda_t + \frac{\eta_t}{\eta_{t-1}}(\lambda_t - \lambda_{t-1}) - \alpha_t \hat{g}'_{\lambda_t}(\epsilon_{1:N_t})$ ◁ grad-update
    **if** $\lambda_{t+1}$ is converged to $\lambda^*$ **then**
      **break**
    **end if**
  **end if**
**end for**
**return** $\lambda^*$

If we use this upper-bounded value instead of $N_t$, i.e.

$$N_t^* = \frac{1}{\mu}\frac{\eta_{t-1}}{\sqrt{C_t}},$$

the ratio of $N_t^*$ and $N_{t-1}^*$ is,

$$\begin{aligned}\frac{N_t^*}{N_{t-1}^*} &= \frac{1}{\mu}\frac{\eta_{t-1}}{\sqrt{C_t}} \cdot \mu\frac{\sqrt{C_{t-1}}}{\eta_{t-2}} \\ &= \frac{\eta_{t-1}}{\eta_{t-2}},\end{aligned}$$

when $t = 2, \ldots, T$. Therefore,

$$N_t^* = \frac{\eta_{t-1}}{\eta_{t-2}} N_{t-1}^*.$$

When $t = 1$,

$$\begin{aligned}\frac{N_1^*}{N_0^*} &= \frac{1}{\mu}\frac{\eta_0}{\sqrt{C_1}} \cdot \mu\sqrt{\frac{C_0}{\mathbb{V}_0}} \\ &= \frac{\eta_0}{\sqrt{\mathbb{V}_0}} \cdot \sqrt{\frac{C_0}{C_1}} \\ &= \frac{\eta_0}{\sqrt{2\mathbb{V}_0}}.\end{aligned}$$

Therefore,

$$\begin{aligned} N_1^* &= \frac{\eta_0}{\sqrt{2\mathbb{V}_0}} N_0^* \\ &= \frac{1}{\sqrt{2\mathbb{V}_0}} N_0^* \ (\because \eta_0 = 1). \end{aligned}$$

Thus, the sample size is estimated by,

$$N_t^* = \begin{cases} \lceil \frac{1}{\sqrt{2\mathbb{V}_0}} N_0^* \rceil & (t = 1), \\ \lceil \frac{\eta_{t-1}}{\eta_{t-2}} N_{t-1}^* \rceil & (t = 2, 3, \ldots, T). \end{cases} \tag{3.7}$$

There are three major learning-rate schedulers: time-based decay, step-based decay, and exponential decay defined as follows [2].

**Definition 3.2** (Time-based Decay Function)**.** *Time-based decay function $\eta_t$ is defined as*

$$\eta_t = \frac{1}{1 + \beta t},$$

*where $\beta$ is the parameter of the degree of decay.*

**Definition 3.3** (Step-based Decay Function)**.** *Step-based decay function $\eta_t$ is defined as*

$$\eta_t = \beta^{\lceil \frac{t}{r} \rceil},$$

*where $\beta$ is the parameter of the degree of decay, and $r$ is the drop-rate parameter.*

**Definition 3.4** (Exponential Decay Function)**.** *Time-based decay function $\eta_t$ is defined as*

$$\eta_t = \exp(-\beta t),$$

*where $\beta$ is the parameter of the degree of decay.*

In these functions, $N_t$ for $t \geq 2$ is estimated as follows:

- Time-based decay: $N_t^* = \lceil \frac{1+\beta(t-2)}{1+\beta(t-1)} N_{t-1}^* \rceil$,
- Step-based decay: $N_t^* = \lceil \beta^{\lceil \frac{1}{r} \rfloor} N_{t-1}^* \rceil$,
- Exponential decay: $N_t^* = \lceil \exp(-\beta) N_{t-1}^* \rceil$,

where $\beta$ and $r$ are the decay and drop-rate parameters.

In this context, we can get another algorithm shown in Algorithm 3. We use this algorithm for experiments in Chapter 5.

**Algorithm 3** Multi-level Monte Carlo Variational Inference in Another Expression of $N_t^*$

**Require:** Data $\mathbf{x}$, random variable $\epsilon \sim p(\epsilon)$, transform $\mathbf{z} = \mathcal{T}(\epsilon; \lambda)$, model $p(\mathbf{x}, \mathbf{z})$, variational family $q(\mathbf{z}|\lambda)$
**Ensure:** Variational parameter $\lambda^*$
**Initialize:** $N_0^*$, $\lambda_0$, $\alpha_0$ and hyperparameter of $\eta$
**for** $t = 0$ to $T$ **do**
  **if** $t = 0$ **then**
    $\epsilon_n \sim p(\epsilon)$ $(n = 1, 2, \ldots, N_0^*)$ $\triangleleft$ sampling $\epsilon$
    $\hat{g}_{\lambda_0}(\epsilon_{1:N_0^*}) = N_0^{*-1} \sum_{n=1}^{N_0^*} g_{\lambda_0}(\epsilon_n)$ $\triangleleft$ calc. RG estimator
    **pick up** one-sample $g_{\lambda_0}(\epsilon)$ from $\hat{g}_{\lambda_0}(\epsilon_{1:N_0^*})$ and calc. $\mathbb{V}_0$
    $\lambda_1 = \lambda_0 - \alpha_0 \hat{g}_{\lambda_0}(\epsilon_{1:N_0^*})$ $\triangleleft$ grad-update
  **else**
    **estimate** $N_t^*$ by Eq. (3.7)
    $\epsilon_n \sim p(\epsilon)$ $(n = 1, 2, \ldots, N_t^*)$ $\triangleleft$ sampling $\epsilon$
    $\hat{g}'_{\lambda_t}(\epsilon_{1:N_t^*}) = N_t^{*-1} \sum_{n=1}^{N_t^*} [g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)})]$ $\triangleleft$ calc. multi-level term
    $\lambda_{t+1} = \lambda_t + \frac{\eta_t}{\eta_{t-1}}(\lambda_t - \lambda_{t-1}) - \alpha_t \hat{g}'_{\lambda_t}(\epsilon_{1:N_t^*})$ $\triangleleft$ grad-update
    **if** $\lambda_{t+1}$ is converged to $\lambda^*$ **then**
      **break**
    **end if**
  **end if**
**end for**
**return** $\lambda^*$

# Chapter 4

# Proofs

In this chapter, we show the proof of theorems, lemmas and propositions introduced in this thesis.

## 4.1 Proof of Theorem 3.1

*Proof.* Now we consider the constrained minimization problem as follow:

$$\min_{N_t} \sum_{t=0}^{T} N_t^{-1}\mathbb{V}_t \quad \text{s.t.} \quad \sum_{t=0}^{T} N_t C_t = M,$$

where $M$ is constant.

For a fixed cost, the variance is minimized by choosing $N_t$ to minimize

$$f(N_t) = \sum_{t=0}^{T} N_t^{-1}\mathbb{V}_t + \mu^2\left(\sum_{t=0}^{T} N_t C_t - M\right), \tag{4.1}$$

for some value of the Lagrange multiplier $\mu^2$ ($\mu > 0$). Thus,

$$\frac{\partial f(N_t)}{\partial N_t} = -N_t^{-2}\mathbb{V}_t + \mu^2 C_t = 0.$$

Then, we obtain

$$N_t = \frac{1}{\mu}\sqrt{\frac{\mathbb{V}_t}{C_t}} \ (\because \mu > 0, C_t > 0, \mathbb{V}_t > 0). \tag{4.2}$$

Now, we substitute (4.2) for (4.1),

$$\begin{aligned} f(N_t) &= \sum_{t=0}^{T} \mu\sqrt{\mathbb{V}_t C_t} + \sum_{t=0}^{T} \mu\sqrt{\mathbb{V}_t C_t} - \mu^2 M \\ &= 2\mu \sum_{t=0}^{T} \sqrt{\mathbb{V}_t C_t} - \mu^2 M. \end{aligned} \tag{4.3}$$

We differentiate $f(N_t)$ in Eq. (4.3) by respect to $\mu$,

$$\frac{\partial f}{\partial \mu} = 2\sum_{t=0}^{T}\sqrt{\mathbb{V}_t C_t} - 2\mu M = 0.$$

Therefore,

$$\mu = \frac{1}{M}\sum_{t=0}^{T}\sqrt{\mathbb{V}_t C_t}. \tag{4.4}$$

Thus, we substitute (4.4) for (4.2),

$$N_t = \frac{M}{\sum_{t=0}^{T}\sqrt{\mathbb{V}_t C_t}}\sqrt{\frac{\mathbb{V}_t}{C_t}}. \tag{4.5}$$

Now we consider the ratio of $N_t$ to $N_{t-1}$,
(i) if $t = 2, \ldots, T$,

$$\begin{aligned}
\frac{N_t}{N_{t-1}} &= \frac{M}{\sum_{t=0}^{T}\sqrt{\mathbb{V}_t C_t}}\sqrt{\frac{\mathbb{V}_t}{C_t}} \cdot \frac{\sum_{t=0}^{T}\sqrt{\mathbb{V}_t C_t}}{M}\sqrt{\frac{C_{t-1}}{\mathbb{V}_{t-1}}} \\
&= \sqrt{\frac{\mathbb{V}_t / C_t}{\mathbb{V}_{t-1}/C_{t-1}}} \\
&= \sqrt{\frac{\mathbb{V}_t / 2c}{\mathbb{V}_{t-1}/2c}} \\
&= \sqrt{\frac{\mathbb{V}_t}{\mathbb{V}_{t-1}}}.
\end{aligned}$$

(ii) If $t = 1$,

$$\begin{aligned}
\frac{N_1}{N_0} &= \frac{M}{\sum_{t=0}^{T}\sqrt{\mathbb{V}_t C_t}}\sqrt{\frac{\mathbb{V}_1}{C_1}} \cdot \frac{\sum_{t=0}^{T}\sqrt{\mathbb{V}_t C_t}}{M}\sqrt{\frac{C_0}{\mathbb{V}_0}} \\
&= \sqrt{\frac{\mathbb{V}_1 / C_1}{\mathbb{V}_0/C_0}} \\
&= \sqrt{\frac{\mathbb{V}_1 / 2c}{\mathbb{V}_0/c}} \\
&= \sqrt{\frac{\mathbb{V}_1}{2\mathbb{V}_0}}.
\end{aligned}$$

According to these results, the optimal sample size $N_t$ $(t = 2, \ldots, T-1)$ is

$$N_t = \sqrt{\frac{\mathbb{V}_t}{\mathbb{V}_{t-1}}} N_{t-1},$$

and

$$N_1 = \sqrt{\frac{\mathbb{V}_1}{2\mathbb{V}_0}} N_0.$$

Therefore, we obtain

$$N_t = \begin{cases} \sqrt{\frac{\mathbb{V}_1}{2\mathbb{V}_0}} N_0 & (t=1), \\ \sqrt{\frac{\mathbb{V}_t}{\mathbb{V}_{t-1}}} N_{t-1} & (t=2,3,\ldots,T), \end{cases}$$

and the claim is proved. □

## 4.2 Proof of Lemma 3.1

*Proof.* MRG in iteration $T$ can be rewritten as

$$\begin{aligned} \nabla^{\mathrm{MRG}}_{\lambda_T}\mathcal{L}(\lambda_T) &= \mathbb{E}_{p(\epsilon)}[g_{\lambda_0}(\epsilon)] + \sum_{t=1}^{T}\left(\mathbb{E}_{p(\epsilon)}\Big[g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)\Big]\right) \\ &= \mathbb{E}_{p(\epsilon)}[g_{\lambda_0}(\epsilon)] + \sum_{t=1}^{T-1}\left(\mathbb{E}_{p(\epsilon)}\Big[g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)\Big]\right) + \mathbb{E}_{p(\epsilon)}\Big[g_{\lambda_T}(\epsilon) - g_{\lambda_{T-1}}(\epsilon)\Big] \\ &= \nabla^{\mathrm{MRG}}_{\lambda_{T-1}}\mathcal{L}(\lambda_{T-1}) + \mathbb{E}_{p(\epsilon)}\Big[g_{\lambda_T}(\epsilon) - g_{\lambda_{T-1}}(\epsilon)\Big]. \end{aligned}$$

By constructing the unbiased estimator in the above $\nabla^{\mathrm{MRG}}_{\lambda_T}\mathcal{L}(\lambda_T)$,

$$\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_T}\mathcal{L}(\lambda_T) = \widehat{\nabla}^{\mathrm{MRG}}_{\lambda_{T-1}}\mathcal{L}(\lambda_{T-1}) + N_T^{-1}\sum_{n=1}^{N_T}\Big[g_{\lambda_T}(\epsilon_{(n,T)}) - g_{\lambda_{T-1}}(\epsilon_{(n,T)})\Big].$$

Further, according to the SGD update-rule, we have

$$\begin{aligned} \lambda_{T+1} &= \lambda_T - \alpha_T\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_T}\mathcal{L}(\lambda_T) \\ &= \lambda_T - \alpha_T\left(\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_{T-1}}\mathcal{L}(\lambda_{T-1}) + N_T^{-1}\sum_{n=1}^{N_T}\Big[g_{\lambda_T}(\epsilon_{(n,T)}) - g_{\lambda_{T-1}}(\epsilon_{(n,T)})\Big]\right) \\ &= \lambda_T - \alpha_T\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_{T-1}}\mathcal{L}(\lambda_{T-1}) - \alpha_t N_T^{-1}\sum_{n=1}^{N_T}\Big[g_{\lambda_T}(\epsilon_{(n,T)}) - g_{\lambda_{T-1}}(\epsilon_{(n,T)})\Big] \\ &= \lambda_T - \alpha_0\eta_T\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_{T-1}}\mathcal{L}(\lambda_{T-1}) - \alpha_T N_T^{-1}\sum_{n=1}^{N_T}\Big[g_{\lambda_T}(\epsilon_{(n,T)}) - g_{\lambda_{T-1}}(\epsilon_{(n,T)})\Big] \\ &= \lambda_T - \frac{\eta_T}{\eta_{T-1}}\alpha_0\eta_{T-1}\widehat{\nabla}^{\mathrm{MRG}}_{\lambda_{T-1}}\mathcal{L}(\lambda_{T-1}) - \alpha_T N_T^{-1}\sum_{n=1}^{N_T}\Big[g_{\lambda_T}(\epsilon_{(n,T)}) - g_{\lambda_{T-1}}(\epsilon_{(n,T)})\Big] \\ &= \lambda_T + \frac{\eta_T}{\eta_{T-1}}(\lambda_T - \lambda_{T-1}) - \alpha_T N_T^{-1}\sum_{n=1}^{N_T}\Big[g_{\lambda_T}(\epsilon_{(n,T)}) - g_{\lambda_{T-1}}(\epsilon_{(n,T)})\Big]. \end{aligned}$$

If we change $T$ to $t$, the claim is proved. □

## 4.3 Proof of Proposition 3.1

*Proof.* According to Assumption 3.4, we obtain the Lipshitzs condition given by

$$||g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)||_2^2 \leq K_2||\mathcal{T}(\epsilon;\lambda_t) - \mathcal{T}(\epsilon;\lambda_{t-1})||_2^2.$$

Now, the all of variational parameters is a single vector $\lambda = (\mathbf{m}, \mathbf{v})$. Therefore, according to Assumption 3.2, the tranceformation $\mathcal{T}(\epsilon; \lambda_t)$ can be written as

$$\mathcal{T}(\epsilon; \lambda_t) = \mathbf{m}_t + \mathbf{v}_t \cdot \epsilon.$$

Thus, we obtain

$$\begin{aligned} ||g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)||_2^2 &\leq K_2 ||\mathcal{T}(\epsilon; \lambda_t) - \mathcal{T}(\epsilon; \lambda_{t-1})||_2^2 \\ &= K_2 \|(\mathbf{m}_t + \mathbf{v}_t \cdot \epsilon) - (\mathbf{m}_{t-1} + \mathbf{v}_{t-1} \cdot \epsilon)\|_2^2 \\ &= K_2 \|(\mathbf{m}_t - \mathbf{m}_{t-1}) + (\mathbf{v}_t - \mathbf{v}_{t-1}) \cdot \epsilon\|_2^2. \end{aligned} \tag{4.6}$$

Because of the SGD update-rule with decreasing learning-rate,

$$\begin{aligned} K_2 \|(\mathbf{m}_t - \mathbf{m}_{t-1}) + (\mathbf{v}_t - \mathbf{v}_{t-1}) \cdot \epsilon\|_2^2 &= K_2 \| - \alpha_{t-1} \hat{g}_{\mathbf{m}_{t-1}(\epsilon_{1:N})} - \alpha_{t-1} \hat{g}_{\mathbf{v}_{t-1}(\epsilon_{1:N})} \cdot \epsilon\|_2^2 \\ &= \alpha_{t-1}^2 K_2 \|\hat{g}_{\mathbf{m}_{t-1}(\epsilon_{1:N})} + \hat{g}_{\mathbf{v}_{t-1}(\epsilon_{1:N})} \cdot \epsilon\|_2^2. \end{aligned}$$

By plugging the above in Eq.(4.6) and taking the expectation, we obtain

$$\begin{aligned} \mathbb{E}_{p(\epsilon)}\left[||g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)||_2^2\right] &\leq \alpha_{t-1}^2 K_2 \mathbb{E}_{p(\epsilon)}\left[\|\hat{g}_{\mathbf{m}_{t-1}(\epsilon_{1:N})} + \hat{g}_{\mathbf{v}_{t-1}(\epsilon_{1:N})} \cdot \epsilon\|_2^2\right] \\ &\leq \alpha_{t-1}^2 K_2 \mathbb{E}_{p(\epsilon)}\left[\|\hat{g}_{\mathbf{m}_{t-1}(\epsilon_{1:N})}\|_2^2 + \|\hat{g}_{\mathbf{v}_{t-1}(\epsilon_{1:N})} \cdot \epsilon\|_2^2\right] \text{ (triangle inequality)} \\ &\leq \alpha_{t-1}^2 K_2 \mathbb{E}_{p(\epsilon)}\left[\|\hat{g}_{\mathbf{m}_{t-1}(\epsilon_{1:N})}\|_2^2 + \|\hat{g}_{\mathbf{v}_{t-1}(\epsilon_{1:N})}\|_2^2 \cdot \|\epsilon\|_2^2\right] \text{ (Cauchy-Schwarz inequality).} \end{aligned}$$

Since $\epsilon \overset{\text{i.i.d.}}{\sim} p(\epsilon) \in \mathbb{R}^d$, the expectation of $\|\epsilon\|_2^2$ is obtained as

$$\begin{aligned} \mathbb{E}_{p(\epsilon)}\left[\|\epsilon\|_2^2\right] &= \mathbb{E}_{p(\epsilon)}[\epsilon_{(1)}^2 + \epsilon_{(2)}^2 + \cdots + \epsilon_{(d)}^2] \\ &= \mathbb{E}_{p(\epsilon)}[\epsilon_{(1)}^2] + \mathbb{E}_{p(\epsilon)}[\epsilon_{(2)}^2] + \cdots + \mathbb{E}_{p(\epsilon)}[\epsilon_{(d)}^2] \\ &= d\mathbb{E}_{p(\epsilon)}[\epsilon_{(1)}^2]. \end{aligned}$$

Therefore, if we consider $\mathbb{E}_{p(\epsilon)}[\epsilon_{(1)}^2] \leq \delta (\geq 0)$,

$$\mathbb{E}_{p(\epsilon)}\left[||g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)||_2^2\right] \leq \alpha_{t-1}^2 K_2 \left(\mathbb{E}_{p(\epsilon)}\left[\|\hat{g}_{\mathbf{m}_{t-1}(\epsilon_{1:N})}\|_2^2\right] + d\mathbb{E}_{p(\epsilon)}[\epsilon_{(1)}^2]\mathbb{E}_{p(\epsilon)}\left[\|\hat{g}_{\mathbf{v}_{t-1}(\epsilon_{1:N})}\|_2^2\right]\right) \tag{4.7}$$

$$\leq \alpha_{t-1}^2 K_2 \left(\mathbb{E}_{p(\epsilon)}\left[\|\hat{g}_{\mathbf{m}_{t-1}(\epsilon_{1:N})}\|_2^2\right] + d\delta\mathbb{E}_{p(\epsilon)}\left[\|\hat{g}_{\mathbf{v}_{t-1}(\epsilon_{1:N})}\|_2^2\right]\right). \tag{4.8}$$

Here, if we use $N$ samples for gradient estimation,

$$\begin{aligned} \mathbb{E}_{p(\epsilon)}\left[\|\hat{g}_{\mathbf{m}_{t-1}(\epsilon_{1:N})}\|_2^2\right] &= \mathbb{E}_{p(\epsilon)}\left[\|\frac{1}{N}\sum_{n=1}^{N} g_{\mathbf{m}_{t-1}(\epsilon_n)}\|_2^2\right] \\ &= \frac{1}{N^2} \cdot N\mathbb{E}_{p(\epsilon)}\left[\|g_{\mathbf{m}_{t-1}}(\epsilon_1)\|_2^2\right] \\ &= \frac{1}{N}\mathbb{E}_{p(\epsilon)}\left[\|g_{\mathbf{m}_{t-1}}(\epsilon_1)\|_2^2\right] \\ &= \mathcal{O}(N^{-1}). \end{aligned}$$

The same results are obtained for the term $\mathbb{E}_{p(\epsilon)}[\|\hat{g}_{\mathbf{v}_{t-1}(\epsilon_{1:N})}\|_2^2]$).

According to this, Eq.(4.7) can be rewritten as

$$\begin{aligned}\mathbb{E}_{p(\epsilon)}\left[||g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)||_2^2\right] &\leq \alpha_{t-1}^2 K_2(N^{-1}C_1 + d\mathbb{E}_{p(\epsilon)}[\epsilon_{(1)}^2]N^{-1}C_2) \\ &\leq \alpha_{t-1}^2 N^{-1} K_2(C_1 + d\delta C_2) \\ &= \alpha_0 \eta_{t-1}^2 N^{-1} K_2(C_1 + d\delta C_2) \\ &= \mathcal{O}(\eta_{t-1}^2 N^{-1}),\end{aligned}$$

where $C_1, C_2$ are positive constances.

Thus, as $t \to \infty$, we can see that $\mathbb{E}_{p(\epsilon)}[||g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)||_2^2] = \mathcal{O}(\eta_{t-1}^2 N^{-1})$. Furthermore, $\mathbb{V}_t$ is typically similar in the magnitude to $\mathbb{E}_{p(\epsilon)}[||g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)||_2^2]$ [28] because

$$\begin{aligned}\mathbb{V}[g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)] &= \mathbb{E}_{p(\epsilon)}\left[\left\|g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)\right\|_2^2\right] - \left\|\mathbb{E}_{p(\epsilon)}\left[g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)\right]\right\|_2^2 \\ &\leq \mathbb{E}_{p(\epsilon)}\left[\left\|g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)\right\|_2^2\right].\end{aligned}$$

Therefore, we obtain the fact that $\mathbb{V}[g_{\lambda_t}(\epsilon) - g_{\lambda_{t-1}}(\epsilon)] = \mathcal{O}(\eta_{t-1}^2 N^{-1})$ (as $t \to \infty$). Thus, if we use one-sample to estimate the gradient estimator, the order of $\mathbb{V}_t$ is $\mathcal{O}(\eta_{t-1}^2)$.

According the fact that $\alpha_t \overset{t\to\infty}{\to} 0$, the one sample variance $\mathbb{V}_t$ goes to 0 asymptotically as iteration proceeds.

Thus, the claim is proved. □

## 4.4 Proof of Lemma 3.2

*Proof.* According to Theorem 3.1,

$$\begin{aligned}N_T &= \left\lceil \sqrt{\frac{\mathbb{V}_T}{\mathbb{V}_{T-1}}} N_{T-1} \right\rceil \\ &= \left\lceil \sqrt{\frac{\mathbb{V}_T}{\mathbb{V}_{T-1}}} \cdot \sqrt{\frac{\mathbb{V}_{T-1}}{\mathbb{V}_{T-2}}} \cdot \sqrt{\frac{\mathbb{V}_{T-2}}{\mathbb{V}_{T-3}}} \cdots \sqrt{\frac{\mathbb{V}_2}{\mathbb{V}_1}} N_1 \right\rceil \\ &= \left\lceil \sqrt{\frac{\mathbb{V}_T}{\mathbb{V}_{T-1}}} \cdot \sqrt{\frac{\mathbb{V}_{T-1}}{\mathbb{V}_{T-2}}} \cdot \sqrt{\frac{\mathbb{V}_{T-2}}{\mathbb{V}_{T-3}}} \cdots \sqrt{\frac{\mathbb{V}_2}{\mathbb{V}_1}} \cdot \sqrt{\frac{\mathbb{V}_1}{2\mathbb{V}_0}} N_0 \right\rceil \\ &= \left\lceil \sqrt{\frac{\mathbb{V}_T}{2\mathbb{V}_0}} N_0 \right\rceil.\end{aligned}$$

Remembering the result of Proposition 3.1, the order of $\mathbb{V}_T$ is $\mathcal{O}(\eta_{T-1}^2)$. Therefore, $\mathbb{V}_T \leq c\eta_{T-1}^2$ where $c$ is a positive constant and

$$\begin{aligned}N_T &\leq \left\lceil \sqrt{\frac{c\eta_{T-1}^2}{2\mathbb{V}_0}} N_0 \right\rceil \\ &= \left\lceil \eta_{T-1} \sqrt{\frac{c}{2\mathbb{V}_0}} N_0 \right\rceil.\end{aligned}$$

$\mathbb{V}_0 < \infty$ and $\eta_{T-1} \to 0$; therefore $N_T$ goes to 1 as $T \to \infty$. Because of $0 \leq t \leq T$, the claim is proved. □

## 4.5 Proof of Lemma 3.3

*Proof.* According to Proposition 3.1, the order of $\mathbb{V}_t$ is $\mathcal{O}(\eta_{t-1}^2)$. Therefore,

$$\mathbb{V}_t \leq \kappa_t \eta_{t-1}^2,$$

where $\kappa_t$ is a positive constant at the iteration $t$.

Then, we obtain the following inequality:

$$\begin{aligned}
\mathbb{V}[\widehat{\nabla}_{\lambda_t}^{\mathrm{MRG}} \mathcal{L}(\lambda_t)] &= \alpha_t^{-2} \mathbb{V}[-\alpha_t \widehat{\nabla}_{\lambda_t} \mathcal{L}(\lambda_t)] \\
&= \alpha_t^{-2} \mathbb{V}\left[ \frac{\eta_t}{\eta_{t-1}} (\lambda_t - \lambda_{t-1}) - \alpha_t N_t^{-1} \sum_{n=1}^{N_t} \left[ g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)}) \right] \right] \\
&= \alpha_t^{-2} \mathbb{V}\left[ - \alpha_t N_t^{-1} \sum_{n=1}^{N_t} \left[ g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)}) \right] \right] \\
&= \mathbb{V}\left[ N_t^{-1} \sum_{n=1}^{N_t} \left[ g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)}) \right] \right] \\
&= N_t^{-2} \mathbb{V}\left[ \sum_{n=1}^{N_t} \left[ g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)}) \right] \right] \\
&= N_t^{-2} N_t \mathbb{V}\left[ g_{\lambda_t}(\epsilon_{(1,t)}) - g_{\lambda_{t-1}}(\epsilon_{(1,t)}) \right] \\
&= N_t^{-1} \mathbb{V}\left[ g_{\lambda_t}(\epsilon_{(1,t)}) - g_{\lambda_{t-1}}(\epsilon_{(1,t)}) \right] \\
&\leq N_t^{-1} \kappa_t \eta_{t-1}^2.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{V}[\widehat{\nabla}_{\lambda_t}^{\mathrm{MRG}} \mathcal{L}(\lambda_t)] &\leq N_t^{-1} \kappa_t \eta_{t-1}^2 \\
&= \kappa_t \cdot \eta_{t-1}^2 N_t^{-1} \\
&= \mathcal{O}(\eta_{t-1}^2 N_t^{-1}).
\end{aligned}$$

and the claim is proved. □

## 4.6 Proof of Theorem 3.2

*Proof.* By Assumption 3.3, we have that $\mathcal{L}(\lambda) \leq \mathcal{L}(\bar{\lambda}) + \nabla_{\bar{\lambda}} \mathcal{L}(\bar{\lambda})^\top (\lambda - \bar{\lambda}) + \frac{1}{2} K_1 ||\lambda - \bar{\lambda}||_2^2, \forall \lambda, \bar{\lambda}$. By using the SGD-update rule, we obtain $\lambda_{t+1} - \lambda_t = -\alpha_t \hat{g}_{\lambda_t}(\epsilon_{1:N})$. Thus, when we set $\lambda = \lambda_{t+1}$ and $\bar{\lambda} = \lambda_t$, this assumption can be expressed as

$$\begin{aligned}
\mathcal{L}(\lambda_{t+1}) - \mathcal{L}(\lambda_t) &\leq \nabla_{\lambda_t} \mathcal{L}(\lambda_t)^\top (\lambda_{t+1} - \lambda_t) + \frac{1}{2} K_1 ||\lambda_{t+1} - \lambda_t||_2^2 \\
&= -\alpha_t \nabla_{\lambda_t} \mathcal{L}(\lambda_t)^\top \hat{g}_{\lambda_t}(\epsilon_{1:N_t}) + \frac{\alpha_t^2 K_1}{2} ||\hat{g}_{\lambda_t}(\epsilon_{1:N})||_2^2.
\end{aligned}$$

Taking the expectation by $\epsilon_{1:N} \sim p(\epsilon)$, we obtain

$$\mathbb{E}_{p(\epsilon_{1:N})}[\mathcal{L}(\lambda_{t+1}) - \mathcal{L}(\lambda_t)] \leq -\alpha_t \nabla_{\lambda_t} \mathcal{L}(\lambda_t)^\top \mathbb{E}_{p(\epsilon_{1:N})}[\hat{g}_{\lambda_t}(\epsilon_{1:N})] + \frac{\alpha_t^2 K_1}{2} \mathbb{E}_{p(\epsilon_{1:N})}\left[ \|\hat{g}_{\lambda_t}(\epsilon_{1:N})\|_2^2 \right].$$

Using the fact that $\mathbb{E}_{p(\epsilon_{1:N})}[||\hat{g}_{\lambda_t}(\epsilon_{1:N})||_2^2] = \mathbb{V}[\hat{g}_{\lambda_t}(\epsilon_{1:N})] + ||\mathbb{E}_{p(\epsilon_{1:N})}[\hat{g}_{\lambda_t}(\epsilon_{1:N})]||_2^2$ and $\mathbb{E}_{p(\epsilon_{1:N})}[\hat{g}_{\lambda_t}(\epsilon_{1:N})] = \nabla_{\lambda_t}\mathcal{L}(\lambda_t)$, we obtain

$$\mathbb{E}_{p(\epsilon_{1:N})}[\mathcal{L}(\lambda_{t+1}) - \mathcal{L}(\lambda_t)] \leq -\alpha_t ||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2 + \frac{\alpha_t^2 K_1}{2}\left(\mathbb{V}[\hat{g}_{\lambda_t}(\epsilon_{1:N})] + \left\|\mathbb{E}_{p(\epsilon_{1:N})}[\hat{g}_{\lambda_t}(\epsilon_{1:N})]\right\|_2^2\right).$$

Further, re-using the fact that $\mathbb{E}_{p(\epsilon_{1:N})}[\hat{g}_{\lambda_t}(\epsilon_{1:N})] = \nabla_{\lambda_t}\mathcal{L}(\lambda_t)$, the above equation can be rewritten as

$$\mathbb{E}_{p(\epsilon_{1:N})}[\mathcal{L}(\lambda_{t+1}) - \mathcal{L}(\lambda_t)] \leq \frac{\alpha_t^2 K_1}{2}\mathbb{V}[\hat{g}_{\lambda_t}(\epsilon_{1:N})] + \left(\frac{\alpha_t^2 K_1}{2} - \alpha_t\right)||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2.$$

Summing for $t = 1, 2, \ldots, T$ and taking the total expectation,

$$\mathbb{E}[\mathcal{L}(\lambda_T) - \mathcal{L}(\lambda_1)] \leq \frac{K_1}{2}\sum_{t=1}^{T}\alpha_t^2\mathbb{E}\left[\mathbb{V}[\hat{g}_{\lambda_t}(\epsilon_{1:N})]\right] + \sum_{t=1}^{T}\left(\frac{\alpha_t^2 K_1}{2} - \alpha_t\right)\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right].$$

From the fact that $\mathcal{L}(\lambda^*) - \mathcal{L}(\lambda_1) \leq \mathbb{E}[\mathcal{L}(\lambda_T) - \mathcal{L}(\lambda_1)]$, where $\lambda_1$ is determinitic and $\lambda^*$ is the optimal parameter, we have the following inequality by dividing the inequality by $A_T = \sum_{t=1}^{T}\alpha_t$:

$$\frac{1}{A_T}[\mathcal{L}(\lambda^*) - \mathcal{L}(\lambda_1)] \leq \frac{K_1}{2A_T}\sum_{t=1}^{T}\alpha_t^2\mathbb{E}\left[\mathbb{V}[\hat{g}_{\lambda_t}(\epsilon_{1:N})]\right] + \frac{1}{A_T}\sum_{t=1}^{T}\left(\frac{\alpha_t^2 K_1}{2} - \alpha_t\right)\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right].$$

Therefore, we can obtain

$$\frac{1}{A_T}\sum_{t=1}^{T}\alpha_t\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right] \leq \frac{1}{A_T}[\mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*)] + \frac{K_1}{2A_T}\sum_{t=1}^{T}\alpha_t^2\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right] + \frac{K_1}{2A_T}\sum_{t=1}^{T}\alpha_t^2\mathbb{E}\left[\mathbb{V}[\hat{g}_{\lambda_t}(\epsilon_{1:N})]\right].$$

If we estimate the gradient values by MC samples, we can see $\mathbb{V}[\hat{g}_{\lambda_t}(\epsilon_{1:N})] \leq \kappa N^{-1}$.

Therefore,

$$\begin{aligned}
\frac{1}{A_T}\sum_{t=1}^{T}\alpha_t\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right] &\leq \frac{1}{A_T}[\mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*)] + \frac{K_1}{2A_T}\sum_{t=1}^{T}\alpha_t^2\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right] + \frac{K_1}{2A_T}\sum_{t=1}^{T}\alpha_t^2\mathbb{E}\left[\mathbb{V}[\hat{g}_{\lambda_t}(\epsilon_{1:N})]\right] \\
&\leq \frac{1}{A_T}[\mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*)] + \frac{K_1}{2A_T}\sum_{t=1}^{T}\alpha_t^2\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right] + \frac{\kappa \cdot K_1}{2A_T N}\sum_{t=1}^{T}\alpha_t^2 \\
&= \frac{1}{A_T}[\mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*)] + \frac{K_1}{2A_T}\sum_{t=1}^{T}\alpha_t^2\left(\mathbb{E}[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2] + \frac{\kappa}{N}\right) \\
&= \frac{1}{A_T}[\mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*)] + \frac{\alpha_0^2 K_1}{2A_T}\sum_{t=1}^{T}\eta_t^2\left(\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right] + \frac{\kappa}{N}\right) \ (\because \alpha_t = \alpha_0 \cdot \eta_t).
\end{aligned}$$

If we estimate the gradient values by RQMC samples, as with the above, we obtain

$$\frac{1}{A_T}\sum_{t=1}^{T}\alpha_t\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right] \leq \frac{1}{A_T}[\mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*)] + \frac{\alpha_0^2 K_1}{2A_T}\sum_{t=1}^{T}\eta_t^2\left(\mathbb{E}\left[||\nabla_{\lambda_t}\mathcal{L}(\lambda_t)||_2^2\right] + \frac{\kappa}{N^2}\right).$$

Thus, the claim is proved. □

## 4.7 Proof of Theorem 3.3

*Proof.* By taking the same result from the proof of Theorem 3.2, we obtain

$$\mathcal{L}(\lambda_{t+1}) - \mathcal{L}(\lambda_t) \leq -\alpha_t \nabla_{\lambda_t} \mathcal{L}(\lambda_t)^\top \hat{g}_{\lambda_t}(\epsilon_{1:N_t}) + \frac{K_1}{2} || - \alpha_t \hat{g}_{\lambda_t}(\epsilon_{1:N_t})||_2^2.$$

Taking expectation by $\epsilon_{1:N} \sim p(\epsilon)$, we obtain

$$\mathbb{E}_{p(\epsilon_{1:N_t})}[\mathcal{L}(\lambda_{t+1}) - \mathcal{L}(\lambda_t)] \leq -\alpha_t \nabla_{\lambda_t} \mathcal{L}(\lambda_t)^\top \mathbb{E}_{p(\epsilon_{1:N_t})}[\hat{g}_{\lambda_t}(\epsilon_{1:N_t})] + \frac{K_1}{2} \mathbb{E}_{p(\epsilon_{1:N})}\Big[|| - \alpha_t \hat{g}_{\lambda_t}(\epsilon_{1:N_t})||_2^2\Big].$$

Using the fact that $\mathbb{E}_{p(\epsilon_{1:N})}[||-\alpha_t \hat{g}_{\lambda_t}(\epsilon_{1:N})||_2^2] = \mathbb{V}[-\alpha_t \hat{g}_{\lambda_t}(\epsilon_{1:N})] + ||\mathbb{E}_{p(\epsilon_{1:N})}[-\alpha_t \hat{g}_{\lambda_t}(\epsilon_{1:N})]||_2^2$ and $\mathbb{E}_{p(\epsilon_{1:N_t})}[\hat{g}_{\lambda_t}(\epsilon_{1:N_t})] = \nabla_{\lambda_t} \mathcal{L}(\lambda_t)$, we obtain

$$\mathbb{E}_{p(\epsilon_{1:N_t})}[\mathcal{L}(\lambda_{t+1}) - \mathcal{L}(\lambda_t)] \leq -\alpha_t ||\nabla_{\lambda_t} \mathcal{L}(\lambda_t)||_2^2 + \frac{K_1}{2}\Big(\mathbb{V}[-\alpha_t \hat{g}_{\lambda_t}(\epsilon_{1:N_t})] + \alpha_t^2 ||\mathbb{E}_{p(\epsilon_{1:N_t})}[\hat{g}_{\lambda_t}(\epsilon_{1:N_t})]||_2^2\Big). \tag{4.9}$$

According to Lemma 3.1 and the proof of this,

$$-\alpha_t \hat{g}_{\lambda_t}(\epsilon_{1:N_t}) = \frac{\eta_t}{\eta_{t-1}}(\lambda_t - \lambda_{t-1}) - \alpha_t \hat{g}'_{\lambda_t}(\epsilon_{1:N_t}),$$

where $\hat{g}'_{\lambda_t}(\epsilon_{1:N_t}) = N_t^{-1} \sum_{n=1}^{N_t} [g_{\lambda_t}(\epsilon_{(n,t)}) - g_{\lambda_{t-1}}(\epsilon_{(n,t)})]$. Therefore,

$$\begin{aligned}
\mathbb{V}[-\alpha_t \hat{g}_{\lambda_t}(\epsilon_{1:N_t})] &= \mathbb{V}\left[\frac{\eta_t}{\eta_{t-1}}(\lambda_t - \lambda_{t-1}) - \alpha_t \hat{g}'_{\lambda_t}(\epsilon_{1:N_t})\right] \\
&= \alpha_t^2 \mathbb{V}[\hat{g}'_{\lambda_t}(\epsilon_{1:N_t})] \\
&\leq \alpha_t^2 \cdot \kappa \eta_{t-1}^2 N_t^{-1}.
\end{aligned}$$

Thus, Eq.(4.9) can be rewritten as

$$\begin{aligned}
\mathbb{E}_{p(\epsilon_{1:N_t})}[\mathcal{L}(\lambda_{t+1}) - \mathcal{L}(\lambda_t)] &\leq -\alpha_t ||\nabla_{\lambda_t} \mathcal{L}(\lambda_t)||_2^2 + \frac{K_1}{2}\Big(\mathbb{V}[-\alpha_t \hat{g}_{\lambda_t}(\epsilon_{1:N_t})] + \alpha_t^2 ||\mathbb{E}_{p(\epsilon_{1:N_t})}[\hat{g}_{\lambda_t}(\epsilon_{1:N_t})]||_2^2\Big) \\
&\leq -\alpha_t ||\nabla_{\lambda_t} \mathcal{L}(\lambda_t)||_2^2 + \frac{K_1}{2}\Big(\alpha_t^2 \cdot \kappa \eta_{t-1}^2 N_t^{-1} + \alpha_t^2 ||\mathbb{E}_{p(\epsilon_{1:N_t})}[\hat{g}_{\lambda_t}(\epsilon_{1:N_t})]||_2^2\Big) \\
&= -\alpha_t ||\nabla_{\lambda_t} \mathcal{L}(\lambda_t)||_2^2 + \frac{\alpha_t^2 K_1}{2}\Big(\kappa \eta_{t-1}^2 N_t^{-1} + ||\mathbb{E}_{p(\epsilon_{1:N_t})}[\hat{g}_{\lambda_t}(\epsilon_{1:N_t})]||_2^2\Big).
\end{aligned}$$

Further, re-using the fact that $\mathbb{E}_{p(\epsilon_{1:N})}[\hat{g}_{\lambda_t}(\epsilon_{1:N})] = \nabla_{\lambda_t} \mathcal{L}(\lambda_t)$, the above equation can be rewritten as

$$\mathbb{E}_{p(\epsilon_{1:N_t})}[\mathcal{L}(\lambda_{t+1}) - \mathcal{L}(\lambda_t)] \leq \frac{\kappa K_1 \alpha_t^2}{2N_t} \cdot \eta_{t-1}^2 + \left(\frac{\alpha_t^2 K_1}{2} - \alpha_t\right) ||\nabla_{\lambda_t} \mathcal{L}(\lambda_t)||_2^2.$$

Summing for $t = 1, 2, \ldots, T$ and taking the total expectation,

$$\mathbb{E}[\mathcal{L}(\lambda_T) - \mathcal{L}(\lambda_1)] \leq \frac{\kappa K_1}{2} \sum_{t=1}^{T} \frac{\alpha_t^2}{N_t} \eta_{t-1}^2 + \sum_{t=1}^{T} \left(\frac{\alpha_t^2 K_1}{2} - \alpha_t\right) \mathbb{E}\Big[||\nabla_{\lambda_t} \mathcal{L}(\lambda_t)||_2^2\Big].$$

From the fact that $\mathcal{L}(\lambda^*) - \mathcal{L}(\lambda_1) \leq \mathbb{E}[\mathcal{L}(\lambda_T) - \mathcal{L}(\lambda_1)]$, we have the following inequality by dividing the inequality by $A_T = \sum_{t=1}^{T} \alpha_t$:

$$\frac{1}{A_T}[\mathcal{L}(\lambda^*) - \mathcal{L}(\lambda_1)] \leq \frac{\kappa K_1}{2A_T} \sum_{t=1}^{T} \frac{\alpha_t^2}{N_t} \eta_{t-1}^2 + \frac{1}{A_T} \sum_{t=1}^{T} \left( \frac{\alpha_t^2 K_1}{2} - \alpha_t \right) \mathbb{E}\left[ ||\nabla_{\lambda_t} \mathcal{L}(\lambda_t)||_2^2 \right].$$

Therefore, we can obtain

$$\begin{aligned}
\frac{1}{A_T} \sum_{t=1}^{T} \alpha_t ||\nabla_{\lambda_t} \mathcal{L}(\lambda_t)||_2^2 &\leq \frac{1}{A_T}[\mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*)] + \frac{K_1}{2A_T} \sum_{t=1}^{T} \alpha_t^2 \mathbb{E}\left[ ||\nabla_{\lambda_t} \mathcal{L}(\lambda_t)||_2^2 \right] + \frac{\kappa K_1}{2A_T} \sum_{t=1}^{T} \frac{\alpha_t^2}{N_t} \eta_{t-1}^2 \\
&= \frac{1}{A_T}[\mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*)] + \frac{K_1}{2A_T} \sum_{t=1}^{T} \alpha_t^2 \left( \mathbb{E}\left[ ||\nabla_{\lambda_t} \mathcal{L}(\lambda_t)||_2^2 \right] + \frac{\kappa}{N_t} \eta_{t-1}^2 \right). \\
&= \frac{1}{A_T}[\mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*)] + \frac{\alpha_0^2 K_1}{2A_T} \sum_{t=1}^{T} \eta_t^2 \left( \mathbb{E}\left[ ||\nabla_{\lambda_t} \mathcal{L}(\lambda_t)||_2^2 \right] + \frac{\kappa}{N_t} \eta_{t-1}^2 \right).
\end{aligned}$$

Thus, the claim is proved. □

## 4.8 Proof of Theorem 3.4

*Proof.* Firstly, we focus on the MC and RQMC-based methods. Because of the definition of SNR and the fact that $\mathbb{E}_{p(\epsilon)}[\hat{g}_\lambda(\epsilon_{1:N})] = \nabla_{\lambda_t} \mathcal{L}(\lambda_t)$, we obtain

$$\begin{aligned}
\mathrm{SNR}(\lambda) &= \frac{\|\mathbb{E}_{p(\epsilon_{1:N})}[\hat{g}_\lambda(\epsilon_{1:N})]\|_2^2}{\sqrt{\mathbb{V}[\hat{g}_\lambda(\epsilon_{1:N})]}} \\
&= \frac{\|\nabla_\lambda \mathcal{L}(\lambda)\|_2^2}{\sqrt{\mathbb{V}[\hat{g}_\lambda(\epsilon_{1:N})]}}.
\end{aligned}$$

By using the order of gradient variance in each methods, we obtain the following lower bounds:

$$\begin{aligned}
\mathrm{SNR}(\lambda_t) &= \frac{\|\nabla_{\lambda_t} \mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\mathbb{V}[\hat{g}_{\lambda_t}(\epsilon_{1:N})]}} \\
&\geq \begin{cases} \frac{\|\nabla_{\lambda_t} \mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\kappa N^{-1}}} & \text{(MC)}, \\ \frac{\|\nabla_{\lambda_t} \mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\kappa N^{-2}}} & \text{(RQMC)}, \end{cases} \\
&= \begin{cases} \frac{\|\nabla_{\lambda_t} \mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\kappa}} \cdot \sqrt{N} & \text{(MC)}, \\ \frac{\|\nabla_{\lambda_t} \mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\kappa}} \cdot N & \text{(RQMC)}. \end{cases}
\end{aligned}$$

Secondly, we show the SNR bound of our method. SNR can be expressed as

$$\begin{aligned}
\mathrm{SNR}(\lambda_t) &= \frac{\|\mathbb{E}_{p(\epsilon_{1:N})}[\hat{g}_\lambda(\epsilon_{1:N})]\|_2^2}{\sqrt{\mathbb{V}[\hat{g}_\lambda(\epsilon_{1:N})]}} \\
&= \frac{\|\nabla_{\lambda_t} \mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\alpha_t^{-2} \mathbb{V}[\alpha_t \hat{g}_\lambda(\epsilon_{1:N})]}}.
\end{aligned}$$

According to the proof of Theorem 3.3,

$$\begin{aligned}\mathbb{V}[\alpha_t \hat{g}_\lambda(\epsilon_{1:N})] &= \mathbb{V}[-\alpha_t \hat{g}_\lambda(\epsilon_{1:N})] \\ &\leq \alpha_t^2 \cdot \kappa \eta_{t-1}^2 N_t^{-1}.\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}\mathrm{SNR}(\lambda_t) &= \frac{\|\nabla_{\lambda_t} \mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\alpha_t^{-2} \mathbb{V}[\alpha_t \hat{g}_\lambda(\epsilon_{1:N})]}} \\ &\geq \frac{\|\nabla_{\lambda_t} \mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\alpha_t^{-2} \alpha_t^2 \cdot \kappa \eta_{t-1}^2 N_t^{-1}}} \\ &= \frac{\|\nabla_{\lambda_t} \mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\kappa \eta_{t-1}^2 N_t^{-1}}} \\ &= \frac{\|\nabla_{\lambda_t} \mathcal{L}(\lambda_t)\|_2^2}{\sqrt{\kappa}} \cdot \frac{\sqrt{N_t}}{\eta_{t-1}}.\end{aligned}$$

Theorem 3.4 holds. □

# Chapter 5

# Experiments

We experimentally analyzed the performance of the proposed method by using three different models: hierarchical linear regression, Bayesian logistic regression, and Bayesian neural network (BNN) regression.

We compared the proposed method with baseline methods in terms of the performance of optimization and prediction by using the ELBO and the log-likelihood on the training and test data sets.

In addition, we used the empirical gradient variance to compare the performance of variance reduction. Finally, we checked the quality of the gradient estimator on the basis of the empirical SNR.

## 5.1 Experimental Settings

The model setting and data were as follows.

### 5.1.1 Hierarchical Linear Regression

We applied hierarchical linear regression to toy data generated from the same generating process of this model. Here, we set a Gaussian hyperprior on $\mu'$, and lognormal hyperpriors on the variance of intercepts $\sigma'$ and the noise $\epsilon$.

The generative process of this model is as follows.

$$
\begin{aligned}
\mu' &\sim \mathcal{N}(0, 10^2), && \text{weight hyper prior} \\
\sigma' &\sim \text{LogNormal}(0.5), && \text{weight hyper prior} \\
\epsilon &\sim \text{LogNormal}(0.5), && \text{noise} \\
\mathbf{b}_i &\sim \mathcal{N}(\mu', \sigma'), && \text{weights} \\
y_i &\sim \mathcal{N}(\mathbf{x}_i^\top \mathbf{b}_i, \epsilon). && \text{output distribution}
\end{aligned}
$$

We set $I = 100$ and $k = 10$, where $k$ denotes the dimension of the data $x_i$ and $I$ is the number of observations. In this setting, the dimension of the whole parameter space is $d = I \times k + k + 2 = 1012$, and this model is approximated by a variational diagonal Gaussian distribution.

We optimized the ELBO of the MC-based and the RQMC-based method by using the Adam optimizer [47] and of the MLMC-based by using the SGD optimizer with learning-rate scheduler $\eta$. To compare the performance of the baseline methods with that of the

proposed method, we used 100 initial MC or RQMC samples. Furthermore, we compared the empirical variance and SNR of these methods by using 100 or 10 initial MC or RQMC samples for inference. In the optimization step, we used $\eta$ as the step-decay function and set the hyperparameter $\{\beta, r\}$ for sample size estimation to $\{0.5, 100\}$. Finally, we set the initial learning rate as 0.01.

### 5.1.2 Bayesian Logistic Regression

We applied it to the brest cancer data set in the UCI repository[1]. Here, we set a standard Gaussian hyper prior on $\mu'$, and an inverse gamma hyper prior (weak information prior) on the variance of weights $\sigma'$.

The generative process of this model is as follows.

$$\begin{aligned}
\sigma' &\sim \text{Gamma}(0.5, 0.5), && \text{weight hyper prior} \\
\mu' &\sim \mathcal{N}(0, 1), && \text{weight hyper prior} \\
\mathbf{z}_i &\sim \mathcal{N}(\mu', 1/\sigma'), && \text{weights} \\
\sigma(x) &= \frac{1}{1 + \exp(-x)}, && \text{Sigmoid function} \\
y_i &\sim \text{Bernoulli}(\sigma(\mathbf{x}_i^\top \mathbf{z}_i)). && \text{output distributions}
\end{aligned}$$

In these settings, the dimension of the whole parameter space is $d = 11$, and this model is also approximated by a variational diagonal Gaussian distribution.

To optimize the ELBO, we used the Adam optimizer for the MC and the RQMC-based methods and the SGD optimizer with learning-rate scheduler $\eta$ for the MLMC-based method.

To compare the performance of the baseline methods with that of the proposed method, we used 100 initial MC or RQMC samples. Furthermore, we compared the empirical variance and SNR of these methods by using 100 or 10 initial MC or RQMC samples for inference. In the optimization step, we used $\eta$ as the step-decay function and set the hyperparameter $\{\beta, r\}$ for sample size estimation to $\{0.5, 100\}$. Finally, we set the initial learning rate as 0.001.

### 5.1.3 Bayesian Neural Network Regression

We applied a bayesian neural network regression (BNN-regression) model to the wine-quality-red data set, which are included the wine-quality data set in the UCI repository[2].

The network consists of a 50-unit hidden layer with ReLU activations. In addition, we set a normal prior over each weight and placed an inverse Gamma hyperprior over each weight prior, and also set an inverse Gamma prior to the observed variance.

The generate process of this model is as follows.

$$\begin{aligned}
\alpha &\sim \text{Gamma}(1., 0.1), && \text{weight hyper prior} \\
\tau &\sim \text{Gamma}(1., 0.1), && \text{noise hyper prior} \\
w_i &\sim \mathcal{N}(0, 1/\alpha), && \text{weights} \\
y &\sim \mathcal{N}(\phi(\mathbf{x}, \mathbf{w}), 1/\tau). && \text{output distributions}
\end{aligned}$$

In this settings, $\phi(\mathbf{x}, \mathbf{w})$ is a multi-layer perceptron which maps input data $\mathbf{x}$ to output $y$ by using the set of weights $w$, and the set of parameters are expressed as $\theta := (\mathbf{w}, \alpha, \tau)$.

[1] https://archive.ics.uci.edu/ml/data sets/Breast+Cancer
[2] https://archive.ics.uci.edu/ml/data sets/Wine+Quality

The model exhibits a posterior of dimension $d = 653$ and was applied to a 100-row data set subsampled from the wine-red data set.

We approximated the posterior of this model by using a variational diagonal Gaussian distribution, and we used the learning-rate scheduler $\eta$ as the step-decay function and set the hyperparameter $\{\beta, r\}$ for sample size estimation to $\{0.5, 100\}$.

To optimize the ELBO, we used the Adam optimizer for the MC and the RQMC-based methods and the SGD optimizer with learning-rate scheduler $\eta$ for the MLMC-based method.

To compare the performance of the baseline methods with that of the proposed method, we used 100 initial MC or RQMC samples. Furthermore, we compared the empirical variance and SNR of these methods by using 100 or 10 initial MC or RQMC samples for inference. Finally, we set the initial learning rate to 0.001.

## 5.2 Results

The all of experimental results are shown in from Figure 5.1 to 5.6. From these results, we found that the proposed method (solid blue lines) converged faster than the baseline methods did in all settings. Furthermore, the proposed method sometimes obtained values closer to the optimal ELBO and log-likelihood for the test data set as shown in Figure 5.4. In addition, the empirical gradient variance and SNR of the proposed method (dashed and solid blue lines, respectively) exhibited lower values than the baseline methods do.

Moreover, as stated in regard to Lemma 3.2, the proposed method performed well even when the sample size was reduced as the optimization proceeded (see Figure 5.7). It shows that the proposed method is able to optimize the ELBO efficiently while saving the computational cost.

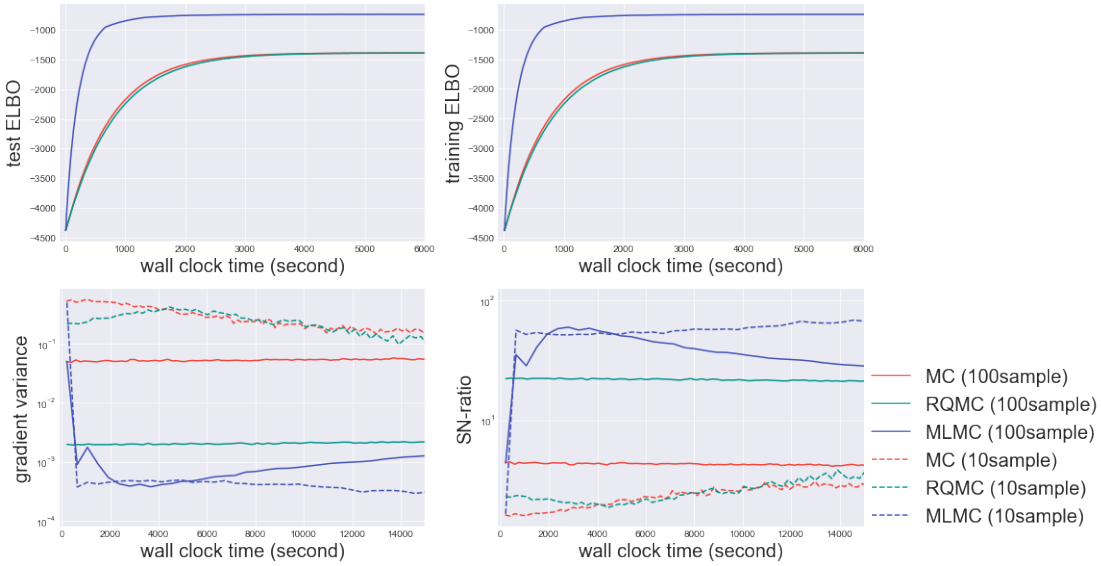


**Figure 5.1:** Experimental results of a hierarchical linear regression. On the first row, test ELBO (higher is better) and traning ELBO (higher is better) is lined up. On the second row, empirical gradient variance (lower is better) and empirical SNR (lower is better) is lined up.

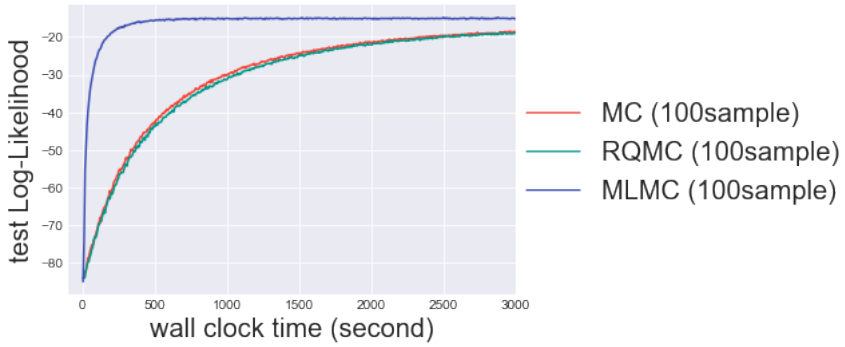


**Figure 5.2:** Experimental results of log-likelihood on test data set in a hierarchical linear regression (higher is better).

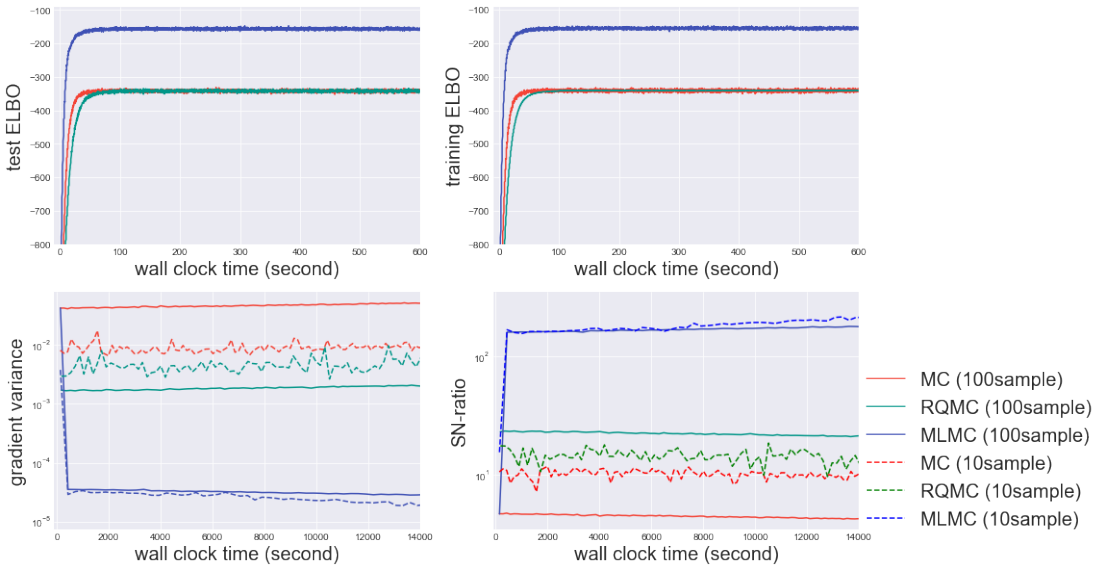


**Figure 5.3:** Experimental results of a bayesian logistic regression. On the first row, test ELBO (higher is better) and traning ELBO (higher is better) is lined up. On the second row, empirical gradient variance (lower is better) and empirical SNR (lower is better) is lined up.

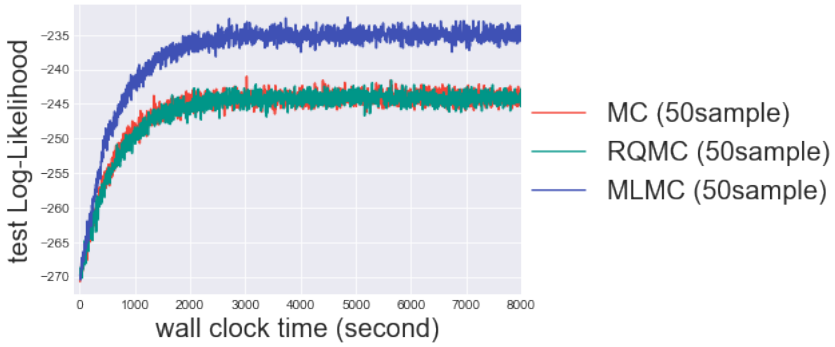


**Figure 5.4:** Experimental results of log-likelihood on test data set in a bayesian logistic regression (higher is better).

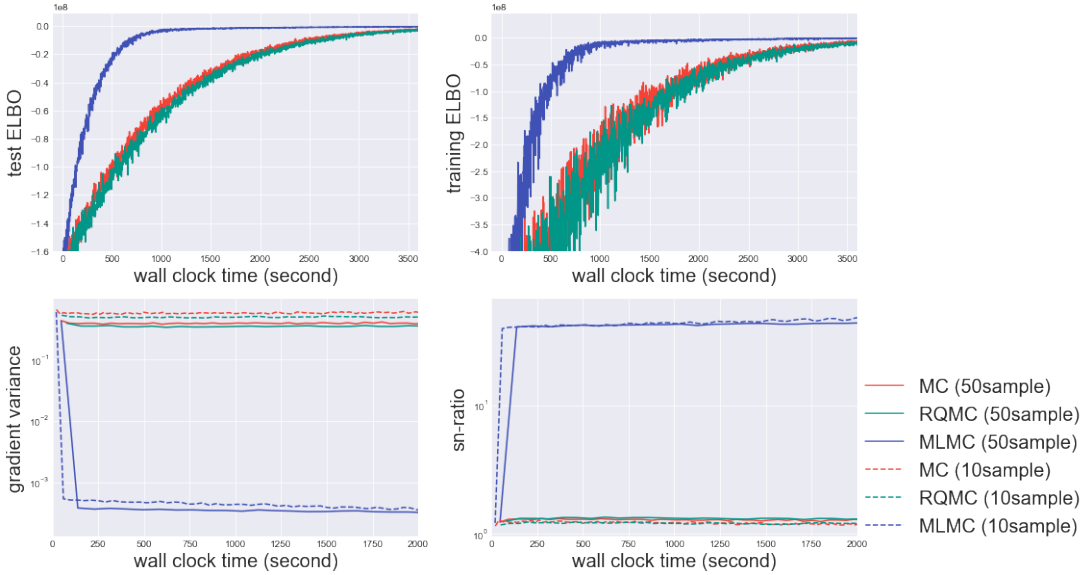


**Figure 5.5:** Experimental results of a bayesian neural network regression. On the first row, test ELBO (higher is better) and traning ELBO (higher is better) is lined up. On the second row, empirical gradient variance (lower is better) and empirical SNR (lower is better) is lined up.

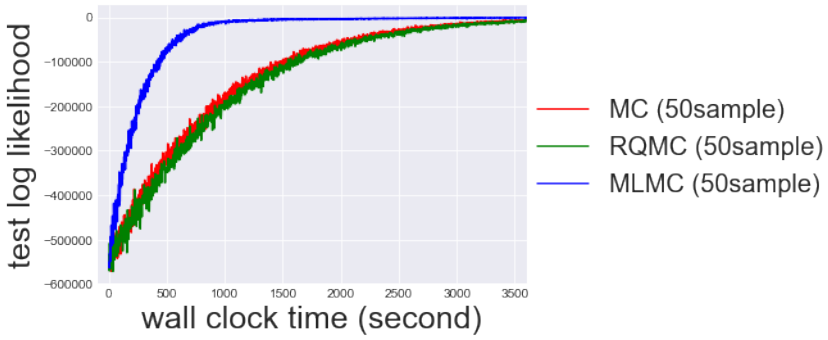


**Figure 5.6:** Experimental results of log-likelihood on test data set in a bayesian neural network regression (higher is better).

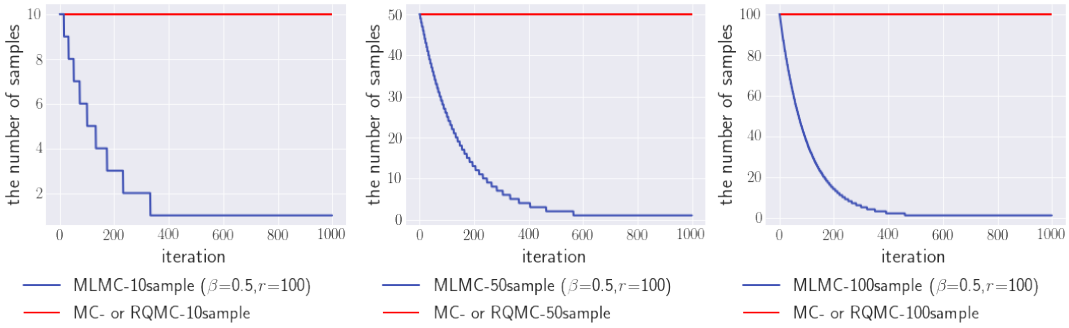


**Figure 5.7:** Reduction results of random variable samples for a gradient estimation, when $\beta = 0.5, r = 100$.

# Chapter 6

# Conclusion and Future Work

We have proposed multi-level Monte Carlo variational inference (MLMCVI), a novel framework of variance and sample-size reduction for MCVI with a reparameterized gradient estimator. In the MLMCVI framework, the optimal number of samples and update schemes are naturally derived, and they provide the minimum total variance per optimization step. The proposed method is easily integrated into automated inference libraries with an auto-differential tool.

Furthermore, we showed theoretical properties of the proposed method that the convergence of the weighted-average gradient norm is accelerated and that the deterioration of the quality of the MRG estimator can be controlled by factor $\eta_{t-1}$. Through three experiments, we also confirmed that the proposed method achieves better or competitive performance in terms of the speed of convergence, variance reduction, and SNR when compared with baseline methods. Moreover, we found that the proposed method sometimes yields values closer to the optimum in terms of the log-likelihood on a test data set.

Because of using pure MC samples, the proposed method can be combined with variance reduction techniques such as control variates and importance sampling based on the reparameterization trick. Moreover, it may be possible to use the RQMC sampling in the proposed method by extending the studies by Giles and Waterhouse [29] and Gerstner and Noll [25]. Also, the current proposed method focuses on the SGD-based update rules; we will extend it to other optimizers (e.g., Adam) in our future study.

In this work, we focused on the Gaussian variational distribution and analyzed the proposed method. However, it could be easily extended to a broader class of distributions by using the generalized reparameterization gradient [79]. We thus plan to extend the proposed method to more general variational distributions and to elucidate additional theoretical properties.

# Bibliography

[1] Saeed Asadi Bagloee, Madjid Tavana, Mohsen Asadi, and Tracey Oliver. Autonomous vehicles: challenges, opportunities, and future implications for transportation policies. *Journal of Modern Transportation*, 24(4):284–303, 12 2016.

[2] Yoshua Bengio. *Practical Recommendations for Gradient-Based Training of Deep Architectures*, pages 437–478. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[3] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518), 2017.

[4] Thomas Bonald and Richard Combes. A minimax optimal algorithm for crowdsourcing. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 4352–4360, 2017.

[5] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

[6] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2): 88–97, 2009.

[7] Alexander Buchholz, Florian Wenzel, and Stephan Mandt. Quasi-Monte Carlo variational inference. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 668–677, 2018.

[8] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[9] Bob Carpenter, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software, Articles*, 76(1): 1–32, 2017.

[10] Do Kook Choe and Eugene Charniak. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336, 11 2016.

[11] K. A. Cliffe, M. B. Giles, Robert Scheichl, and Aretha L Teckentrup. Multilevel monte carlo methods and applications to elliptic pdes with random coefficients. *Computing and Visualization in Science*, 14(1):3–15, 2011.

[12] Siamak Zamani Dadaneh, Xiaoning Qian, and Mingyuan Zhou. Bnp-seq: Bayesian nonparametric differential expression analysis of sequencing count data. *Journal of the American Statistical Association*, 113(521):81–94, 2018.

[13] Timo M Deist, Andrew Patti, Zhaoqi Wang, David Krane, Taylor Sorenson, and David Craft. Simulation-assisted machine learning. *Bioinformatics*, 35(20):4072–4080, 03 2019.

[14] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31:105–112, 03 2009.

[15] Josef Dick, Frances Y. Kuo, and Ian H. Sloan. High-dimensional integration : The quasi-monte carlo way. *Acta Numeruca*, 22:133–288, 2013.

[16] Joshua V. Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A. Saurous. Tensorflow distributions. *arXiv preprint*, arXiv:1711.10604, 2017.

[17] Minh Doan and Anne E. Carpenter. Leveraging machine vision in cell-based diagnostics to do more with less. *Nature Materials*, 18(5):414–418, 2019.

[18] Justin Domke. Provable gradient variance guarantees for black-box variational inference. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2019.

[19] Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216 – 222, 1987.

[20] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7): 2121–2159, 2011.

[21] Alan E. Gelfand and Adrian F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990.

[22] Alan E. Gelfand and Adrian F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990.

[23] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, 12 1984.

[24] Anastasis Georgoulas, Jane Hillston, and Guido Sanguinetti. Unbiased bayesian inference for population markov jump processes via random truncations. *Statistics and Computing*, 27(4):991–1002, 2017.

[25] Thomas Gerstner and Marco Noll. Randomized multilevel quasi-monte carlo path simulation. *Recent Developments in Computational Finance*, pages 349–369, 2013.

[26] Michael B. Giles. Multi-level monte carlo path simulation. *Operations Research*, 56(3): 607–617, 2008.

[27] Michael B. Giles. Multilevel monte carlo methods. *Monte Carlo and Quasi-Monte Carlo Methods 2012*, 56(3):79–98, 2013.

[28] Michael B. Giles. Multilevel monte carlo methods. *Acta Numeruca*, 24:259–328, 2015.

[29] Michael B. Giles and Ben J. Waterhouse. Multilevel quasi-monte carlo path simulation. In *Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics*, pages 165–181, 2009.

[30] Michael B. Giles, Mateusz B. Majka, Lukasz Szpruch, Sebastian J. Vollmer, and Konstantinos C. Zygalakis. Multi-level monte carlo methods for the approximation of invariant measures of stochastic differential equations. *Statistics and Computing*, 2019.

[31] Mike Giles, Tigran Nagapetyan, Lukasz Szpruch, Sebastian Vollmer, and Konstantinos Zygalakis. Multilevel monte carlo for scalable bayesian computations. *arXiv preprint arXiv:1609.06144*, 2016.

[32] Paul Glasserman. Monte carlo methods in financial engineering. *Stochastic Modelling and Applied Probability*, 53(1):XIII,596, 2003.

[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[34] Ehsan Hajiramezanali, Siamak Zamani Dadaneh, Alireza Karbalayghareh, Mingyuan Zhou, and Xiaoning Qian. Bayesian multi-domain learning for cancer subtype discovery from next-generation sequencing count data. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 9115–9124, 2018.

[35] J. H. Halton. Algorithm 247: Radical-inverse quasi-random point sequence. *Commun. ACM*, 7(12):701–702, 12 1964.

[36] A.E. Hassanien and D.A. Oliva. *Advances in Soft Computing and Machine Learning in Image Processing*. Studies in Computational Intelligence. Springer International Publishing, 2017.

[37] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970.

[38] Stefan Heinrich. Multilevel monte carlo methods. In *Proceedings of the Third International Conference on Large-Scale Scientific Computing-Revised Papers*, pages 58–67, 2001.

[39] Philipp Hennig. Fast probabilistic optimization from noisy gradients. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 62–70, 2013.

[40] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

[41] Dietmar Jannach. *Recommender systems : an introduction*. Cambridge University Press, 2011.

[42] Dominik Janzing and Bernhard Schoelkopf. Causal inference using the algorithmic markov condition. *Information Theory, IEEE Transactions on Information Theory*, 56: 5168 – 5194, 11 2010.

[43] Ajay Jasra, Seongil Jo, David Nott, Christine Shoemaker, and Raul Tempone. Multilevel monte carlo in approximate bayesian computation. *arXiv preprint arXiv:1702.03628*, 2017.

[44] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2): 183–233, 1999.

[45] Hao-Cheng Kao, Kai-Fu Tang, and Edward Y. Chang. Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 2305–2313, 2018.

[46] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 5574–5584, 2017.

[47] Diederick P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[48] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[49] Samory Kpotufe, Eleni Sgouritsa, Dominik Janzing, and Bernhard Schölkopf. Consistency of causal inference under the additive noise model. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML)*, volume 32 of *Proceedings of Machine Learning Research*, pages 478–486, 22–24 Jun 2014.

[50] Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and David Blei. Automatic variational inference in stan. In *Advances in Neural Information Processing Systems 28*, pages 568–576, 2015.

[51] David Kurtz, Mohammad Shahrokh Esfahani, Florian Scherer, Joanne Soo, Michael Jin, Chih Liu, Aaron Newman, Ulrich Dührsen, Andreas Hüttmann, Olivier Casasnovas, Jason Westin, Matthais Ritgen, Sebastian Böttcher, Anton Langerak, Mark Roschewski, Wyndham Wilson, Gianluca Gaidano, Davide Rossi, Jasmin Bahlo, and Ash Alizadeh. Dynamic risk profiling using serial tumor biomarkers for personalized outcome prediction. *Cell*, 178, 07 2019.

[52] Pierre L'Ecuyer and Christiane Lemieux. Recent advances in randomized quasi-monte carlo methods. *Modeling uncertainty*, pages 419–474, 2005.

[53] Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Directional analysis of stochastic gradient descent via von mises-fisher distributions in deep learning. *arXiv preprint arXiv:1810.00150*, 2018.

[54] Christian Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer, New York, NY, USA, 2009.

[55] Gunther Leobacher and Friedrich Pillichshammer. *Introduction to quasi-Monte Carlo integration and applications*. Compact Textbooks in Mathematics. Springer, 2014.

[56] Ximing Li, Changchun Li, Jinjin Chi, and Jihong Ouyang. Variance reduction in black-box variational inference by adaptive importance sampling. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJICAI)*, pages 2404–2410, 2018.

[57] Qiang Liu, Jian Peng, and Alexander T Ihler. Variational inference for crowdsourcing. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25 (NeurIPS)*, pages 692–700, 2012.

[58] Anne-Marie Lyne, Mark Girolami, Yves Atchadé, Heiko Strathmann, and Daniel Simpson. On russian roulette estimates for bayesian inference with doubly-intractable likelihoods. *Statistical Science*, 30(4):443–467, 2015.

[59] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[60] Andrew Miller, Nick Foti, Alexander D' Amour, and Ryan P Adams. Reducing reparameterization gradient variance. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 3708–3718, 2017.

[61] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

[62] Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, 1992.

[63] Harald Niederreiter. Random number generation and quasi-monte carlo methods. 1992.

[64] John Paisley, David M. Blei, and Michael I. Jordan. Variational bayesian inference with stochastic search. In *Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML)*, pages 1363–1370, 2012.

[65] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 8024–8035, 2019.

[66] Robert Price. A useful theorem for nonlinear devices having gaussian inputs. *IRE Trans. Information Theory*, 4:69–72, 1958.

[67] Tom Rainforth, Adam R. Kosiorek, Tuan Anh Le, Chris J. Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 4274–4282, 2018.

[68] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 814–822, 2014.

[69] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 8 2010.

[70] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1278–1286, 2014.

[71] Chang-han Rhee and Peter W. Glynn. Unbiased estimation with square root convergence for SDE models. *Operations Research*, 63(5):1026–1043, 2015.

[72] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems: introduction and challenges. In *Recommender systems handbook*, pages 1–34. Springer, 2015.

[73] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[74] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag, 2005.

[75] M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1): 139–140, January 2010. URL https://www.ncbi.nlm.nih.gov/pubmed/19910308.

[76] Geoffrey Roeder, Yuhuai Wu, and David Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 2017.

[77] Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo Method*. Wiley Publishing, 3rd edition, 2016.

[78] F. J. R. Ruiz, M. K. Titsias, and D. M. Blei. Overdispersed black-box variational inference. In *Uncertainty in Artificial Intelligence (UAI)*, 2016.

[79] F. J. R. Ruiz, M. K. Titsias, and D. M. Blei. The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, 2016.

[80] J. Sakaya and A. Klami. Importance sampled stochastic optimization for variational inference. In *Uncertainty in Artificial Intelligence (UAI)*, 2017.

[81] Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 3067–3075, 2017.

[82] I. M. Sobol. Distribution of points in a cube and approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Phisics*, 7:86–112, 1967.

[83] Michalis K. Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pages 1971–1979, 2014.

[84] Michalis K. Titsias and Miguel Lázaro-Gredilla. Local expectation gradients for black box variational inference. In *Advances in Neural Information Processing Systems 28 (NeurIPS)*, pages 2638–2646, 2015.

[85] Seiya Tokui and Issei Sato. Reparameterization trick for discreate variables. *arXiv preprint*, arXiv:1611.01239, 2016.

[86] Dustin Tran, Matthew D. Hoffman, Rif A. Saurous, Eugene Brevdo, Kevin Murphy, and David M. Blei. Deep probabilistic programming. In *International Conference on Learning Representations (ICLR)*, 2017.

[87] Minh-Ngoc Tran, David J Nott, and Robert Kohn. Variational bayes with intractable likelihood. *Journal of Computational and Graphical Statistics*, 26(4):873–882, 2017.

[88] George Tucker, Dieterich Lawson, Shixiang Gu, and Christopher Maddison. Doubly reparameterized gradient estimators for monte carlo objectives. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[89] Oriol Vinyals, Ł ukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2773–2781. Curran Associates, Inc., 2015.

[90] I. D. Vrontos, P. Dellaportas, and D. N. Politis. Full bayesian inference for garch and egarch models. *Journal of Business & Economic Statistics*, 18(2):187–198, 2000.

[91] Eric-Jan Wagenmakers, Maarten Marsman, Tahira Jamil, Alexander Ly, Josine Verhagen, Jonathon Love, Ravi Selker, Quentin F. Gronau, Martin Šmíra, Sacha Epskamp, Dora Matzke, Jeffrey N. Rouder, and Richard D. Morey. Bayesian inference for psychology. part i: Theoretical advantages and practical ramifications. *Psychonomic Bulletin & Review*, 25(1):35–57, Feb 2018.

[92] David J. Warne, Ruth E. Baker, and Matthew J. Simpson. Multilevel rejection sampling for approximate bayesian computation. *Computational Statistics & Data Analysis*, 124: 71–86, 2018.

[93] Bruce Western. Bayesian analysis for sociologists: An introduction. *Sociological Methods & Research*, 28(1):7–34, 1999.

[94] Ming Xu, Matias Quiroz, Robert Kohn, and Scott A. Sisson. Variance reduction properties of the reparameterization trick. In *Proceedings of the 22th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

[95] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *arXiv preprint*, arXiv:1711.05597, 2017.

[96] Yuchen Zhang, Xi Chen, Denny Zhou, and Michael L. Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *Journal of Machine Learning Research*, 17(102):1–44, January 2016.

# Appendix A

# Additional Information on MLMC

We introduce the additional information on MLMC to help understand the background, the proposed method, algorithm, and theoretical analysis.

## A.1 Sampling Method and Multi-level Monte Carlo (MLMC)

When we approximate posterior distributions, Monte Carlo methods are often used for estimating expectation of intractable objects with several random samples. The mean squared error (MSE) of approximation with random samples is a rate of $\mathcal{O}(N^{-1})$, and an accuracy of $\epsilon$ requires $N = \mathcal{O}(\epsilon^{-2})$ samples. This rate can be too high for application. One approach to addressing this high cost is the use of the QMC or RQMC methods, in which the samples are not chosen randomly and independently, instead of being selected very carefully to reduce the error [28]. In the best cases, the error rate is $\mathcal{O}(N^{-2}\log N^{2d-2})$. There are many reviews about the QMC approach provided by Niederreiter [62], L'Ecuyer and Lemieux [52] and Leobacher and Pillichshammer [55].

Another approach to improving the computational efficiency is the MLMC method proposed by Heinrich [38]. This method has been often used in stochastic differential equations for options pricing [26, 11, 71]. In statistics, there are many applications in approximate Bayesian computation [31, 43, 92]. In a Bayesian framework, Giles et al. [31, 30] applied MLMC to stochastic gradient MCMC algorithms such as the stochastic gradient Langevin dynamics (SGLD), which discretize the posterior of SDE based on the multi-level step size and couple them.

Because of the linearity of expectations, given a sequence $P_0, P_1, \ldots, P_{L-1}$ which approximates $P_L$ with increasing accuracy, we have the simple identity:

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{l=1}^{L} \mathbb{E}[P_l - P_{l-1}]. \tag{A.1}$$

We can thus use the following unbiased estimator for $\mathbb{E}[P_L]$,

$$\mathbb{E}[P_L] \approx N_0^{-1} \sum_{n=1}^{N_0} P_0^{(0,n)} + \sum_{l=1}^{L} \left\{ N_l^{-1} \sum_{l=1}^{N_l} (P_l^{(l,n)} - P_{l-1}^{(l,n)}) \right\}, \tag{A.2}$$

with the inclusion of $l$ in $(l, n)$ indicating that independent samples are used at each level of correction.

If we define $V_0, C_0$ to be the variance and cost of *one sample* of $P_0$, and $\mathbb{V}_l, C_l$ to be the variance and cost of *one sample* of $P_l - P_{l-1}$, then the total variance and cost of Eq.(A.2) are $\sum_{l=0}^{L} N_l^{-1}\mathbb{V}_l$ and $\sum_{l=0}^{L} N_l C_l$, respectively.

Thus, if $Y$ is a multi-level estimator given by

$$Y = \sum_{l=0}^{L} Y_l,\ Y_l = N_l^{-1} \sum_{n=1}^{N_l} (P_l^{(l,n)} - P_{l-1}^{(l,n)}),$$

with $P_{-1} \equiv 0$, then

$$\mathbb{E}[Y] = \mathbb{E}[P_L],\ V[Y] = \sum_{l=0}^{L} N_l^{-1}\mathbb{V}_l,\ \mathbb{V}_l \equiv V[P_l - P_{l-1}].$$

## A.2 Control Variates and Relationship to Two-Level MLMC

One of the classic methods to reduce the variance of Monte Carlo samples is the control variates method [32]. When we want to estimate $\mathbb{E}[f]$ and there is a function $h$ which is high correlated to $f$ with a known expectation $\mathbb{E}[h]$, we can use the unbiased estimator for $\mathbb{E}[f]$ which are consisted from $N$ i.i.d. samples $\omega^{(n)}$ as follows,

$$\hat{f}(\omega^{(n)}) = N^{-1} \sum_{n=1}^{N} \{f(\omega^{(n)}) - a(h(\omega^{(n)}) - \mathbb{E}[h])\}. \tag{A.3}$$

Then, the variance of $\hat{f}(\omega^{(n)})$ is expressed as

$$\mathbb{V}[\hat{f}(\omega^{(n)})] = \mathbb{V}[f(\omega^{(n)})] + a^2\mathbb{V}[h(\omega^{(n)})] - 2a\mathrm{Cov}(f(\omega^{(n)}), h(\omega^{(n)})),$$

and the optimal value for $a$ is $\rho\sqrt{V[f]/V[h]}$, where $\rho$ is the correlation between $f$ and $h$. So, the variance of this estimator is reduced by a factor of $1 - \rho^2$ (see Giles [26]).

Two-level MLMC method is similar to this method. According to Giles [27], if we want to estimate $\mathbb{E}[P_1]$ but it is much cheaper to simulate $P_0$ which approximates $P_1$, then since

$$\mathbb{E}[P_1] = \mathbb{E}[P_0] + \mathbb{E}[P_1 - P_0], \tag{A.4}$$

we can use the unbiased two-level estimator given by

$$N_0^{-1} \sum_{n=1}^{N_0} P_0^{(n)} + N_1^{-1} \sum_{n=1}^{N_1} \left( P_1^{(n)} - P_0^{(n)} \right). \tag{A.5}$$

There are two different points from control variates methods: the value of $\mathbb{E}[P_0]$ is *unknown* and $a$ takes one.

# Appendix B

# Additional Experimental Results on Various Initial Learning-rate

In this part, we show the additional experimental results of the ELBO and log-likelihood values on test dataset. Here, we set the various initial learning-rate to confirm the affection of our variance reduction method.

From these results, we found that the proposed method archived better performances on every $\alpha_0$.

## B.1 Hierarchical Linear Regression

### B.1.1 $\alpha_0 = 0.1$

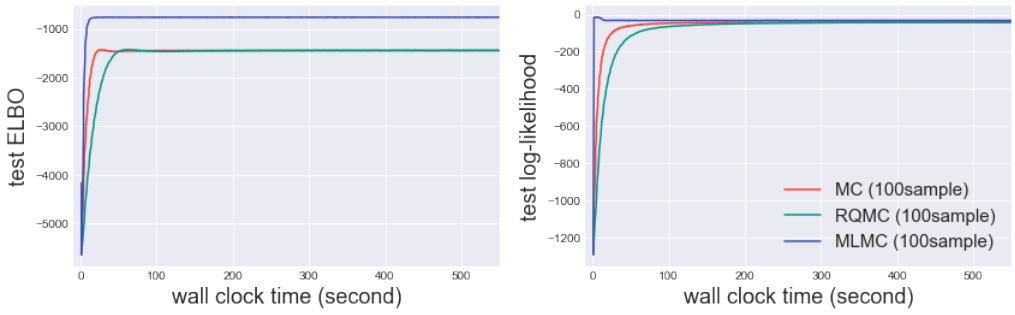


**Figure B.1:** Experimental results when the initial learning rate $\alpha_0 = 0.1$. Test ELBO (higher is better) and test log-likelihood (higher is better) are lined up from left.

### B.1.2 $\alpha_0 = 0.01$

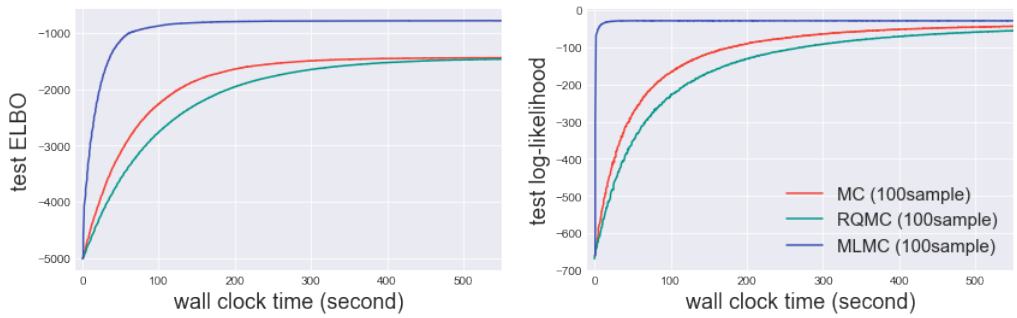


**Figure B.2:** Experimental results when the initial learning rate $\alpha_0 = 0.01$. Test ELBO (higher is better) and test log-likelihood (higher is better) are lined up from left.

### B.1.3 $\alpha_0 = 0.05$

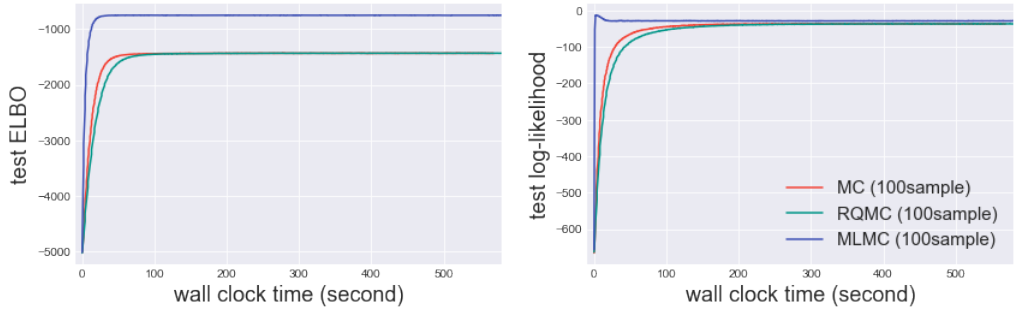


**Figure B.3:** Experimental results when the initial learning rate $\alpha_0 = 0.05$. Test ELBO (higher is better) and test log-likelihood (higher is better) are lined up from left.

## B.2 Bayesian Logistic Regression

### B.2.1 $\alpha_0 = 0.1$

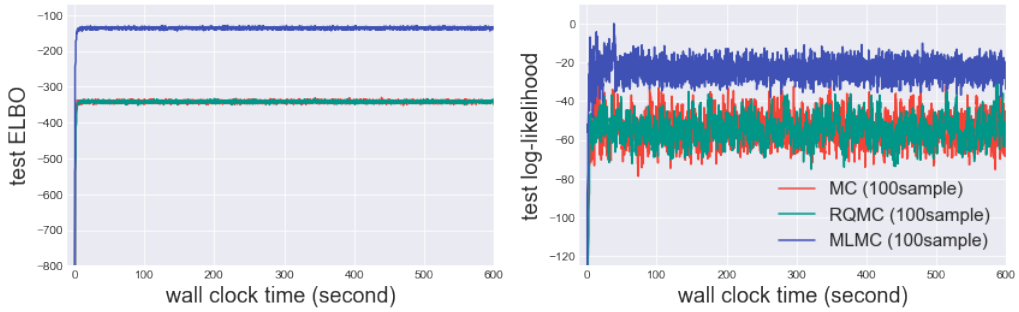


**Figure B.4:** Experimental results when the initial learning rate $\alpha_0 = 0.1$. Test ELBO (higher is better) and test log-likelihood (higher is better) are lined up from left.

### B.2.2 $\alpha_0 = 0.01$

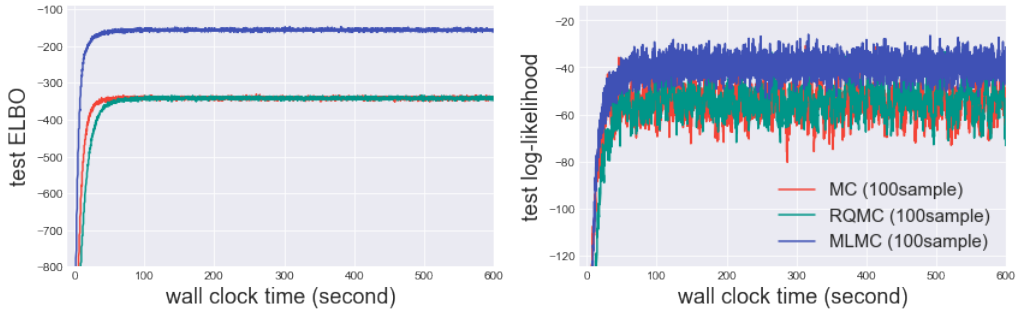


**Figure B.5:** Experimental results when the initial learning rate $\alpha_0 = 0.01$. Test ELBO (higher is better) and test log-likelihood (higher is better) are lined up from left.

### B.2.3 $\alpha_0 = 0.05$

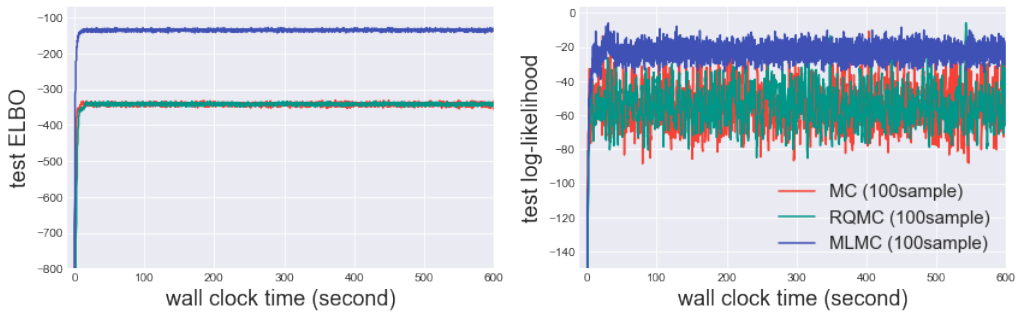


**Figure B.6:** Experimental results when the initial learning rate $\alpha_0 = 0.05$. Test ELBO (higher is better) and test log-likelihood (higher is better) are lined up from left.

## B.3 Bayesian Neural Network Regression

### B.3.1 $\alpha_0 = 0.05$

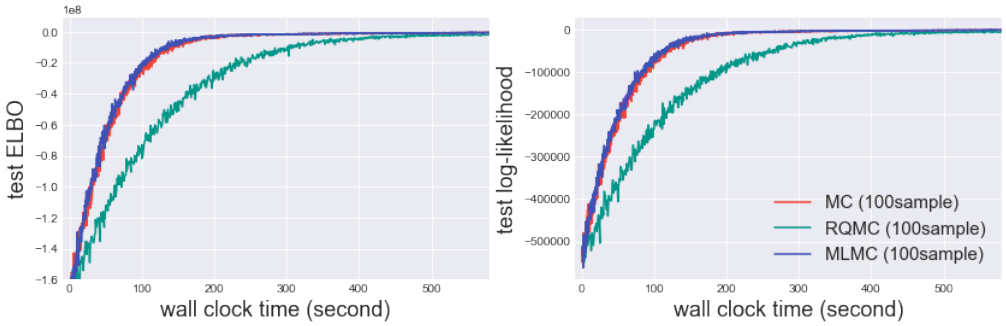


**Figure B.7:** Experimental results when the initial learning rate $\alpha_0 = 0.05$. Test ELBO (higher is better) and test log-likelihood (higher is better) are lined up from left.

### B.3.2 $\alpha_0 = 0.001$

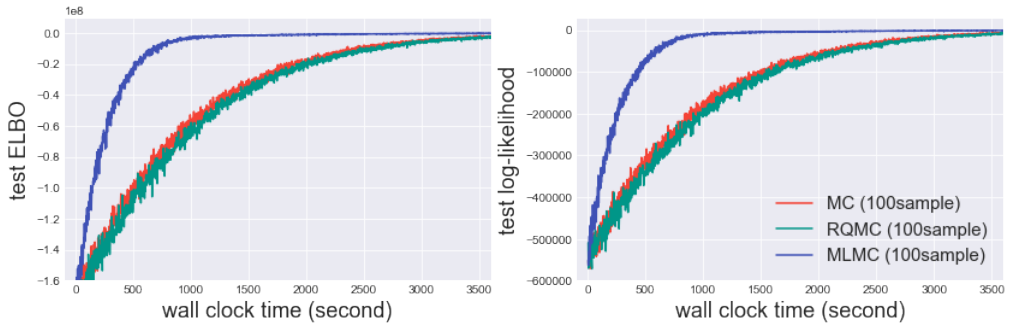


**Figure B.8:** Experimental results when the initial learning rate $\alpha_0 = 0.001$. Test ELBO (higher is better) and test log-likelihood (higher is better) are lined up from left.

### B.3.3 $\alpha_0 = 0.0001$

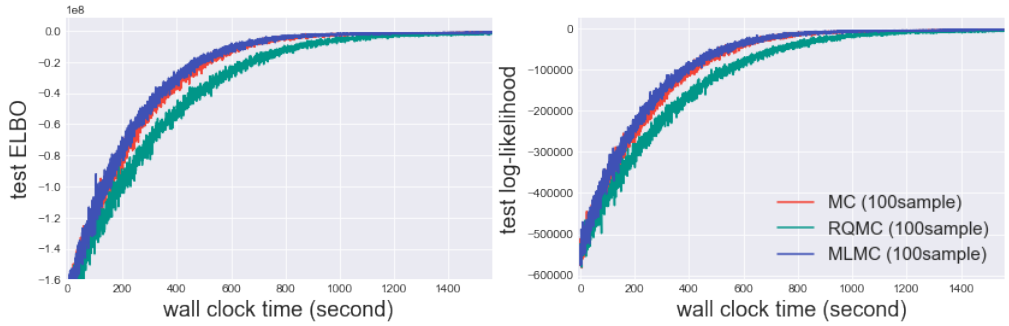


**Figure B.9:** Experimental results when the initial learning rate $\alpha_0 = 0.0001$. Test ELBO (higher is better) and test log-likelihood (higher is better) are lined up from left.

# Appendix C

# Experiment on Image Dataset

In this chapter, we show the experimental results on the large image data to confirm the predictive performance on the basis of the test log-likelihood. We use a bayesian logistic regression for the Fashion-MNIST data set.

## C.1 Details of Model and Experimental Results on Image Dataset

Here, we set a standard Gaussian hyper prior on $\mu'$, and an inverse gamma hyper prior (weak information prior) on the variance of weights $\sigma'$. Thus, the generate process of this model is:

$$
\begin{aligned}
&\sigma' \sim \text{Gamma}(0.5, 0.5), && \text{weight hyper prior}\\
&\mu' \sim \mathcal{N}(0, 1), && \text{weight hyper prior}\\
&\mathbf{z}_i \sim \mathcal{N}(\mu', 1/\sigma'), && \text{weights}\\
&\sigma(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}, && \text{Softmax function}\\
&y \sim \text{Categorical}(\sigma(\phi(\mathbf{x}_i^\top \mathbf{z}_i))). && \text{output distributions}
\end{aligned}
$$

In these settings, the dimension of the whole parameter space is $d = 786$, and this model is also approximated by a variational diagonal Gaussian distribution.

We optimized the ELBO of the MC-, the RQMC-, and the MLMC-based methods by using the SGD optimizer with a learning-rate scheduler $\eta$. To compare the performance of the baseline methods with that of the proposed method, we used 100 initial MC or RQMC samples. In the optimization step, we used $\eta$ as the step-decay function and set the hyperparameter $\{\beta, r\}$ for sample size estimation to $\{0.5, 100\}$. Finally, we set the initial learning rate as 0.01, 0.005, or 0.001.

From the results in Figure C.1, we find that, even if we use large-scale image-data sets such as Fashion-MNIST, the proposed method achieves to yield values closer to the optimum in terms of the log-likelihood on the test data set.

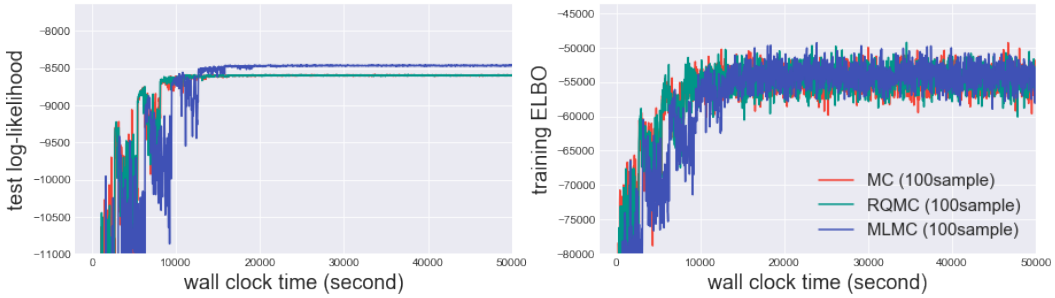


**Figure C.1:** Experimental results when the initial learning rate $\alpha_0 = 0.01$. Test log-likelihood (higher is better) and traning ELBO (higher is better) are lined up from left.

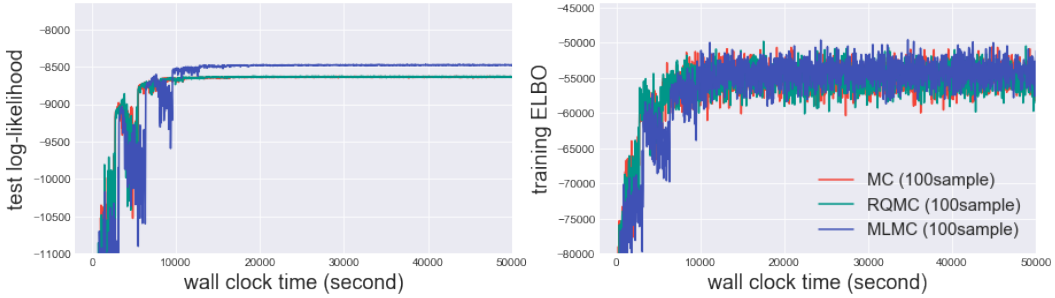


**Figure C.2:** Experimental results when the initial learning rate $\alpha_0 = 0.005$. Test log-likelihood (higher is better) and traning ELBO (higher is better) are lined up from left.

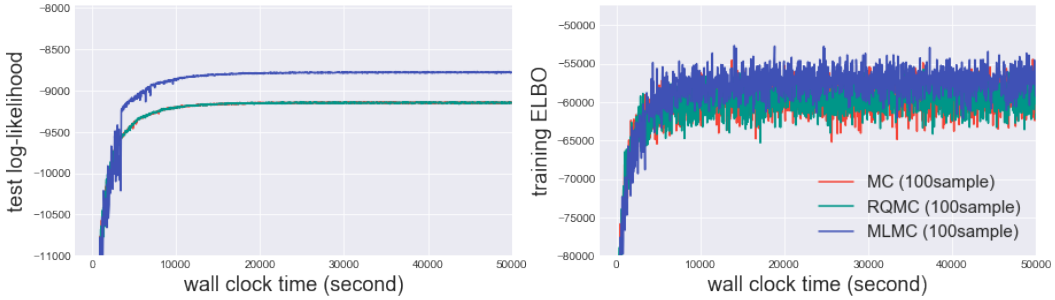


**Figure C.3:** Experimental results when the initial learning rate $\alpha_0 = 0.001$. Test log-likelihood (higher is better) and traning ELBO (higher is better) are lined up from left.

### C.1.1 Experimental Results on Data-size Change

From the experimental results, we can see that the proposed method sometimes obtain a better optimum in sight of the log-likelihood on the test dataset. Then, how much training data is needed to achieve a better performance than that of the benchmark method? To confirm this, we conduct the following experiments.

Firstly, we use the same settings on the bayesian logistic regression experiments in Chapter 5 and Appendix 5.1.2. Secondly, we split 80% of data as a training dataset and the rest as a test dataset. In addition, we also divided training data into 10%, 20%, 40%, 60%, and 80%, and calculated each test log-likelihood values.

The results are shown in Figure C.4. From this, we can see that, even if 20% training data is reduced, our method achieves almost equal performance compared to that of the benchmark method on full training data.

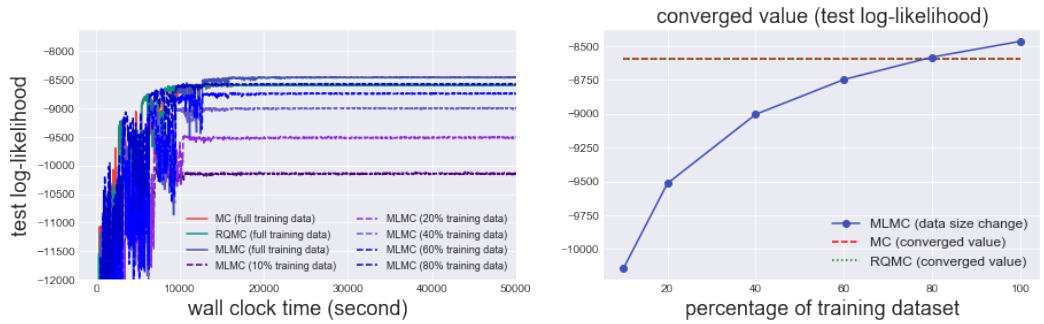


**Figure C.4:** Experimental results when the size of training data changes. Test log-likelihood (higher is better) and the converged value of it (higher is better) per each percentages are lined up from left.