Université de Montréal

On Improving Variational Inference with Low-Variance Multi-Sample Estimators

par

Eeshan Gunesh Dhekane

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en informatique

December 17, 2020

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

On Improving Variational Inference with Low-Variance Multi-Sample Estimators

présenté par

Eeshan Gunesh Dhekane

a été évalué par un jury composé des personnes suivantes :

Ioannis Mitliagkas
(président-rapporteur)

Aaron Courville
(directeur de recherche)

Alain Tapp
(membre du jury)

Résumé

Les progrès de l'inférence variationnelle, tels que l'approche de *variational autoencoder* (VI) (Kingma and Welling (2013), Rezende et al. (2014)) et ses nombreuses modifications, se sont avérés très efficaces pour l'apprentissage des représentations latentes de données. *Importance-weighted variational inference* (IWVI) par Burda et al. (2015) améliore l'inférence variationnelle en utilisant plusieurs échantillons indépendants et répartis de manière identique pour obtenir des limites inférieures variationnelles plus strictes. Des articles récents tels que l'approche de *hierarchical importance-weighted autoencoders* (HIWVI) par Huang et al. (2019) et la modélisation de la distribution conjointe par Klys et al. (2018) démontrent l'idée de modéliser une distribution conjointe sur des échantillons pour améliorer encore l'IWVI en le rendant efficace pour l'échantillon. L'idée sous-jacente de ce mémoire est de relier les propriétés statistiques des estimateurs au resserrement des limites variationnelles. Pour ce faire, nous démontrons d'abord une borne supérieure sur l'écart variationnel en termes de variance des estimateurs sous certaines conditions. Nous prouvons que l'écart variationnel peut être fait disparaître au taux de $\mathcal{O}\left(\frac{1}{n}\right)$ pour une grande famille d'approches d'inférence variationelle. Sur la base de ces résultats, nous proposons l'approche de *Conditional-IWVI* (CIWVI), qui modélise explicitement l'échantillonnage séquentiel et conditionnel de variables latentes pour effectuer *importance-weighted variational inference*, et une approche connexe de *Antithetic-IWVI* (AIWVI) par Klys et al. (2018). Nos expériences sur les jeux de données d'analyse comparative, tels que MNIST (LeCun et al. (2010)) et OMNIGLOT (Lake et al. (2015)), démontrent que nos approches fonctionnent soit de manière compétitive, soit meilleures que les références IWVI et HIWVI en tant que le nombre d'échantillons augmente. De plus, nous démontrons que les résultats sont conformes aux propriétés théoriques que nous avons prouvées. En conclusion, nos travaux fournissent une perspective sur le taux d'amélioration de l'inference variationelle avec le nombre d'échantillons utilisés et l'utilité de modéliser la distribution conjointe sur des représentations latentes pour l'efficacité de l'échantillon. ***Mots-clés:*** inférence variationelle, réduction de la variance.

Abstract

Advances in variational inference, such as variational autoencoders (VI) (Kingma and Welling (2013), Rezende et al. (2014)) along with its numerous modifications, have proven highly successful for learning latent representations of data. Importance-weighted variational inference (IWVI) by Burda et al. (2015) improves the variational inference by using multiple i.i.d. samples for obtaining tighter variational lower bounds. Recent works like hierarchical importance-weighted autoencoders (HIWVI) by Huang et al. (2019) and joint distribution modeling by Klys et al. (2018) demonstrate the idea of modeling a joint distribution over samples to further improve over IWVI by making it sample efficient. The underlying idea in this thesis is to connect the statistical properties of the estimators to the tightness of the variational bounds. Towards this, we first demonstrate an upper bound on the variational gap in terms of the variance of the estimators under certain conditions. We prove that the variational gap can be made to vanish at the rate of $\mathcal{O}\left(\frac{1}{n}\right)$ for a large family of VI approaches. Based on these results, we propose the approach of Conditional-IWVI (CIWVI), which explicitly models the sequential and conditional sampling of latent variables to perform importance-weighted variational inference, and a related approach of Antithetic-IWVI (AIWVI) by Klys et al. (2018). Our experiments on the benchmarking datasets MNIST (LeCun et al. (2010)) and OMNIGLOT (Lake et al. (2015)) demonstrate that our approaches perform either competitively or better than the baselines IWVI and HIWVI as the number of samples increases. Further, we also demonstrate that the results are in accordance with the theoretical properties we proved. In conclusion, our work provides a perspective on the rate of improvement in VI with the number of samples used and the utility of modeling the joint distribution over latent representations for sample efficiency in VI.

Keywords: variational inference, variance reduction.

Contents

List of tables

List of figures

List of acronyms and abbreviations

VI	Variational inference
IWVI	Importance-weighted variational inference
HIWVI	Hierarchical importance-weighted variational inference
CIWVI	Conditional importance-weighted variational inference
AIWVI	Antithetic importance-weighted variational inference
ELBO	Evidence lower bound
AE	Autoencoders
MLE	Maximum likelihood estimation
MAP	Maximum a posteriori estimation
SGD	Stochastic gradient descent
VLB	Variational lower bound
LBE	Lower bounded estimator assumption
MCE	Monte-Carlo estimator

Acknowledgement

I thank my advisor, Prof. Aaron Courville, for selecting me into the M.Sc. program at Université de Montréal/Mila, for the excellent supervision throughout my program, for granting me the freedom to work on the research topics of my interest, for sharing his knowledge and ideas throughout my research, and for directing my efforts better.

I thank Chin-Wei Huang for his help in my work presented in this thesis, for insightful discussions on the topic of variational inference, and for his help in better shaping my thesis.

I thank my friends G. S. S. Srinivas Rao and Rahul Patel for the helpful discussions and for all their help in setting up computational resources. I also thank Olexa Bilaniuk for his help in addressing the issues related to the Mila cluster.

I thank Linda Peinthière of Mila and Céline Bégin of Université de Montréal for their help in all the administrative matters associated with my studies and research.

Finally, I thank my parents, Gayatri and Gunesh Dhekane, and my beloved brother Sourish Dhekane for all their support, encouragement, and help throughout my life. I dedicate my efforts and achievements to them for without them, it would not have been possible.

Dedicated to my parents, Gayatri and Gunesh, and my beloved brother Sourish

Chapter 1

Introduction

The problem of unsupervised learning is one of the most important and challenging tasks in the area of machine learning. This is mainly because it is easier, faster, and economic to obtain huge unlabeled datasets rather than having them labeled manually. Further, unsupervised learning of representations seems important in understanding how humans learn about their surroundings; it is evident that humans do not learn the representations of their surroundings solely from supervision. Thus, one of the aims of unsupervised learning is to uncover the latent patterns in the data without any explicit supervision. Ideally, we would like to learn generative models of the data, which can explain how the data is generated. In particular, generative models allow for learning distributions over the data in terms of latent representations. Such generative models can consequently enable sampling from the data distribution, learning missing data components, learning from smaller amounts of data, and so on. Learning such models requires the computation of the posterior distributions over the latent structure conditioned on the input data. This posterior distribution is intractable in many widely used choices of parametrization; it either involves integrals or sums that do not have analytic closed-form solutions or involve computing exponentially many terms, which is computationally infeasible. One of the strategies to mitigate this problem is to formulate the problem of learning the posterior distribution as an optimization problem and perform approximate inference.

Variational inference is such an approximate inference approach that allows learning parametrized generative models of data $\mathbf{x}$ in terms of their latent representations $\mathbf{z}$ when the true posterior $p(\mathbf{z} \mid \mathbf{x})$ is intractable. Advances in variational inference, like variational autoencoders (VAEs) by Kingma and Welling (2013), enable learning of deep and end-to-end trainable models $p_\theta(\mathbf{x}, \mathbf{z})$, which scale well for large datasets. However, VAEs utilize a family $\mathcal{F} = \{q_\phi(\mathbf{z} \mid \mathbf{x}) \mid \phi \in \Phi\}$ of parametrized proposal posterior distributions to model the true posterior, where Φ is the set of all permissible parameters. This approximation results in a biased optimization objective called the evidence lower bound (ELBO); the intractable $p(\mathbf{z} \mid \mathbf{x})$ leads to the computation and optimization of the ELBO given by: $L = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[p_\theta(\mathbf{x} \mid \mathbf{z})] - \mathcal{D}_{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) || p_\theta(\mathbf{z}))$. This ELBO is always a lower bound for the true objective $\log p_\theta(\mathbf{x})$ that we aim to maximize. This bias results in one major problem; the optimization of the biased objective of ELBO leads to learned parameters that are not optimal for the maximization of the data marginal $p_\theta(\mathbf{x})$. Decreasing the gap between $p_\theta(\mathbf{x})$ and the ELBO: $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[p_\theta(\mathbf{x} \mid \mathbf{z})] - \mathcal{D}_{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) || p_\theta(\mathbf{z}))$, called the variational gap, can indeed result in better models of the data. Thus, it is imperative to understand the effect of the variational gap on variational inference and develop techniques that can decrease this gap for better variational inference.

The work on importance-weighted autoencoders (IWAE) by Burda et al. (2015) demonstrates a mechanism to provably improve the ELBO using multiple samples from the proposal posterior, making the ELBO approach $\log p_\theta(\mathbf{x})$ asymptotically with an increasing number of samples. This work also demonstrates that training with the less biased ELBO results in higher values of the data marginal $p_\theta(\mathbf{x})$, resulting in better models of the data. IWAE has led to an increased interest in the approach of using multiple samples for improving variational inference. Our work too is inspired by and builds on this very idea.

1.0.1. Goals and Contributions

In our work, we ask the following questions and investigate their answers.

Q1: How fast does VI improve with the number of samples?

Many recent works have demonstrated theoretical properties of improvements in the variational inference with the number of samples. The measure of this improvement is defined using the variational gap and the aim is to bound it from above as a function of certain properties of the estimators used to estimate the data marginal $p(\mathbf{x})$ and the number of samples. In this direction, many recent approaches have connected the variance of estimators to the variational gap. However, finding a simple upper bound on the variational gap just in terms of the variance of the estimators and the number of samples is a standing challenge. Towards this, we demonstrate a simple bound on the variational gap using only the estimator variance under amicable and justifiable assumptions. This bound enables a convenient reductionism; with our bound, we reduce the problem of improving variational inference (or, decreasing the variational gap) to the problem of designing multi-sample low-variance estimators of the data marginal $\log p(\mathbf{x})$.

Q2: Can modeling joint distribution over samples result in sample-efficient VI? We focus on using multiple samples for improving variational inference and thus, it is natural to search for techniques that allow us to achieve these improvements with as few samples as possible. Recent works like Huang et al. (2019) demonstrate that learning correlation between samples can allow learning hierarchical latent variables, improve the performance of VAEs, and can potentially decrease the sample count. The idea consists of replacing i.i.d. sampling of latent variables, as done in Burda et al. (2015), with the conditional sampling of latent variables, which resonates with the notion of antithetic sampling. However, more importantly, it gives a mechanism to further reduce the estimator variance, thereby decreasing the variational gap and improving the variational inference. Towards this, we propose a generalization of IWAE, which we call the Conditional-IWAE (CIWAE), that enables explicit modeling of correlation between multiple samples. The idea is that instead of the independent sampling $\mathbf{z}^i \overset{\text{iid}}{\sim} q(\cdot \mid \mathbf{x})$, we sample every next latent variable conditioned on previous samples: $\mathbf{z}^i \sim q\left(\cdot \mid \mathbf{z}^{1:i-1}, \mathbf{x}\right)$. This is a highly general form of conditional sampling that models a large family of conditional/hierarchical latent variables approaches. Most importantly, we prove a set of favorable properties of this generalization. We first prove that despite the conditional sampling of latent variables, the corresponding estimators are not only unbiased but are pairwise uncorrelated. Consequently, we prove that despite the modeling of joint distribution over the samples, our approach enjoys the same asymptotic properties as the ones enjoyed by the IWAE and other similar approaches; we prove that the variational lower bound of CIWAE asymptotically approaches $\log p(\mathbf{x})$ at the rate of $\mathcal{O}\left(\frac{1}{n}\right)$.

In addition to CIWAE, we consider one of its special cases, which we call Antithetic-IWAE (AIWAE). In AIWAE, instead of generating all the n i.i.d. samples of IWAE, we only generate $\left\lceil \frac{n}{2} \right\rceil$ i.i.d. samples $\{\mathbf{z}^i\}_{i=1}^{\left\lceil \frac{n}{2} \right\rceil}$ and generate the rest of the samples in a manner resembling the antithetic sampling: $\mathbf{z}^{\left\lceil \frac{n}{2} \right\rceil + i} = 2 \cdot \mu - \mathbf{z}^i \; \forall \; i \in \left\{1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor\right\}$, where μ represents the mean of the i.i.d. samples. We prove that similar to CIWAE, the variational lower bound of AIWAE asymptotically approaches $\log p(\mathbf{x})$ at the rate of $\mathcal{O}\left(\frac{1}{n}\right)$.

For experimentation, we parametrize our CIWAE and AIWAE approaches with neural networks. We perform our experimentation with the benchmarking datasets MNIST (LeCun et al. (2010)) and OMNIGLOT (Lake et al. (2015)) used in the previous state-of-the-art approaches. We demonstrate that the approaches of CIWAE and AIWAE either outperform or perform competitive to the previous state-of-the-art as the number of samples increases. This empirically confirms the hypothesis that modeling the joint distribution over the latent variables leads to better variational inference in a sample-efficient manner.

The rest of the thesis is organized as follows. In Chapter 2, we provide a brief introduction to the basics of machine learning, probabilistic machine learning, and neural network architectures that are relevant for our experimentation. In Chapter 3, we prove a simpler upper bound on the variational gap in terms of the estimator variance and the number of samples. In Chapter 4, we describe the Conditional-IWAE and Anithetic-IWAE approaches, provide their theoretical properties, describe the experimentation, and demonstrate results that support our hypotheses about better variational inference by modeling the joint distribution over samples. Finally, in Chapter 5, we discuss the conclusions of our works and describe some directions for future research.

Chapter 2

Machine Learning Background

2.1. Artificial Intelligence and Machine Learning

We humans possess a remarkable ability called **intelligence**. It allows us to observe the patterns in our surroundings, extract knowledge and learn skills from these observations, use them to predict and control the events in the future, and take decisions to achieve the desired effects. Equipped with our powerful intelligence, we have achieved astonishing feats over the last few thousand years, which set us apart from our biological ancestor great apes, and other animals. We learned to farm our food instead of constantly hunting for it and evolved to form societies. We invented the concepts of currency, trade, art, and religion. We are constantly uncovering the fundamental rules of nature through science and mathematics. We have built machines to ease our efforts and increase our productivity. The inventions of the **computers** and **computer programs/algorithms** are arguably two of the most impactful achievements of the modern era. However, despite our tremendous progress, there remain many problems and phenomena that we are yet to fully understand. Understanding intelligence itself is one of such standing challenges.

One of the ways to understand any phenomenon is to try to *create* it from its components[1]. In the same spirit, the works of Turing (1950) gave rise to the field of **artificial intelligence (AI)**, which is dedicated to developing algorithms for simulating intelligence in machines. Advances in the field of AI can not only enhance our understanding of the notion of intelligence itself but can also improve upon human intelligence. Human intelligence allows

[1] "What I cannot create, I do not understand"– Richard Feynmann.

extremely efficient generalization; it enables learning from experience in a manner such that we can apply the learned knowledge and skills to previously unseen problems and adapt efficiently to new environments. However, although highly powerful, human intelligence is limited by biological constraints, such as the memory capacity of the human brain as well as the rate of neuron firing. A computer program, on the other hand, can potentially be constructed to have memory as well as processing power that is multiple orders of magnitude higher than humans. Thus, advancing the field of AI can help enhance human intelligence by combining its power with the computational benefits of a machine.

Machine learning (ML) is a subfield of AI that aims to develop algorithms that can *learn* from the data to better perform the particular task at hand and generalize for similar but previously unseen data. ML algorithms depend on sufficiently large amounts of data to learn from, powerful parametrized models for learning the data representations, scalable algorithms to train such models, and the computing power to actuate these algorithms. As a result of the availability of massive amounts of data, the research on neural networks and their training via the back-propagation algorithm, and the development of faster compute technologies like graphical processing units (GPUs)/tensor processing units (TPUs), ML approaches have become the state-of-the-art in almost all of the problems of interest in the domain of AI. Consequently, ML forms the contemporary paradigm for the approaches to AI. In the next section, we consider the basic ML concepts and terminologies.

2.2. Machine Learning Basics

2.2.1. Supervision in Learning

Datasets are collections of data points and form one of the most important components of any ML algorithm. ML algorithms are designed to *adapt* in a data-driven manner so that they can *learn* from the data. Certain datasets contain data points in the form of pairs of *inputs* along with their *labels*, in which case they are termed as *labeled* datasets. For instance, the MNIST dataset by LeCun et al. (2010) contains the images of digits as inputs and the digit identity as the label. However, labeling of datasets is a demanding task and requires significant human effort, which is evident from the curation of the ImageNet dataset (Deng et al. (2009)). On the other hand, thanks to the age of "big data", *unlabeled* datasets containing only the inputs without any annotations are readily available. Below, we formalize these notions of labeled and unlabeled datasets.

Definition 2.2.1 (Datasets). A dataset D is a collection of data points $\mathbf{t}^{(i)}$: $D = \left\{\mathbf{t}^{(i)}\right\}_i$. For a labeled dataset, the data point is a tuple of inputs $\mathbf{x}^{(i)}$ and labels $y^{(i)}$: $\mathbf{t}^{(i)} = \left(\mathbf{x}^{(i)}, y^{(i)}\right)$, whereas for an unlabeled dataset, the data point consists of just the inputs $\mathbf{x}^{(i)}$: $\mathbf{t}^{(i)} = \mathbf{x}^{(i)}$. We denote by $\mathcal{X}, \mathcal{Y}$ the spaces of inputs and labels respectively.

One of the ways in which an ML algorithm can learn is with supervision and using a labeled dataset. One can attempt to learn a *predictive function* that can process an input data point to generate the corresponding label. Learning such a predictive function for the labels from the inputs using a labeled dataset is called **supervised learning**.

Definition 2.2.2 (Supervised Learning). A supervised learning algorithm attempts to learn a predictive function $f : \mathcal{X} \longrightarrow \mathcal{Y}$ using a given labeled dataset $D = \left\{\mathbf{x}^{(i)}, y^{(i)}\right\}_i$ such that when provided with unseen but similar inputs $\mathbf{x}^* \in \mathcal{X}$, the function can correctly predict the corresponding label $y^* \in \mathcal{Y}$ as $f\left(\mathbf{x}^*\right)$.

Remark 2.2.3 (Classification and Regression). Most of the time, supervised learning algorithms are used either to predict the class of the inputs, which can take one of the finitely many distinct values, or to predict a scalar-valued function of the inputs, which can take values from a continuous interval. The former case, where the labels are discrete valued $\left(y^{(i)} \in \mathcal{Y}, \text{ with } |\mathcal{Y}| = n \in \mathbb{N}\right)$, is called the classification problem. The latter case, where the label can take a continuous/ranged value $\left(y^{(i)} \in \mathcal{Y} \subseteq \mathbb{R}\right)$, is called the regression problem.

However, as seen before, curating massive labeled datasets is a highly demanding task. Thus, it is imperative to have algorithms that can learn even from unlabeled datasets. Learning with only an unlabeled dataset in order to uncover the latent structure in the inputs is referred to as **unsupervised learning**. This notion is formalized below.

Definition 2.2.4 (Unsupervised Learning). An unsupervised learning algorithm attempts to learn the latent structure/representation $\mathbf{z} \in \mathcal{Z}$ for the input $\mathbf{x} \in \mathcal{X}$, where $\mathcal{Z}$ represents the space of latent structure/representation.

Remark 2.2.5 (**Different Approaches to Unsupervised Learning**)**.** Note that the latent representations that unsupervised approaches aim to learn are ought to have certain desirable characteristics, based on which they are classified. For instance, one can aim to group the inputs into different collections such that the inputs from the same collection are similar to one another and the inputs from different collections differ from one another. This approach is called **clustering**. Alternatively, one can aim to learn a lower dimensional representation, or a *manifold*, for the inputs. Such approaches are called **dimensionality reduction** or **manifold learning**. Another general approach is to learn a distribution over the space of inputs such that it assigns higher values only at the input data points. This approach is called **density estimation**, and forms the basis of our work.
Note that in certain cases, one can learn using a dataset that is only partially labeled. Such learning that leverages both the labeled and the unlabeled datasets is termed **semi-supervised learning**. Another approach is to utilize one part of the unlabeled data as the inputs and treat the rest of the parts of the data as the corresponding labels to learn its representations. This approach is called **self-supervised learning**. Also, there exists a different formulation of learning called **reinforcement learning** in which the learner is modeled as an agent that acts in an environment to optimize its reward.

2.2.2. Parametrization, Hyper-Parameters, and Capacity

In the ML paradigm, **parametrized functions** are used to obtain the representations of the data. **Parameters** refer to all the trainable scalars of the function, which are learned using the given dataset. In particular, the predictive function f in the case of supervised learning approaches is invariably chosen to be a parametric function $f = f_\theta : \mathcal{X} \longrightarrow \mathcal{Y}$ with the parameter $\theta \in \Theta$, where Θ represents the set of all permissible parameters. Similarly, the latent representations $\mathbf{z}$ in the case of unsupervised learning approaches are modeled in terms of parametrized functions $g_\phi(\mathbf{x})$ of the inputs $\mathbf{x}$ with the parameter $\phi \in \Phi$ chosen from the set Φ of all permissible parameters. For example, suppose that we have a dataset of points $D = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_i$ with $\mathcal{X} = \mathcal{Y} = \mathbb{R}$. We want to parametrize the predictive function f for this regression problem as a polynomial of degree at most d. Then, the family of parametrized functions would be as follows.

$$\mathcal{F}_d = \left\{f_{(a_0,\ldots,a_d)} : \mathbb{R} \longrightarrow \mathbb{R} \mid f_{(a_0,\ldots,a_d)}(\mathbf{x}) = \textstyle\sum_{0 \le i \le d} a_i \cdot \mathbf{x}^i,\ a_i \in \mathbb{R}\ \forall\ i \in \{0, \ldots, d\}\right\} \quad (2.2.1)$$

Note that we can learn the *optimal* values of the coefficients a_i $(i \in \{0, \dots, d\})$ using the given dataset D, which makes them the parameters of the approach. However, note that choosing the value of the degree d is up to the designer of the approach; it can not be learned from the given dataset D. Such scalars that define the choice of the parametric family but whose value can not be learned from the given dataset are called the **hyper-parameters** of the approach. Thus, the degree d is a hyper-parameter of the above approach.

Hyper-parameters control the *richness* of the family of parametrized functions. This is evident from the above example; we have $\mathcal{F}_m \subset \mathcal{F}_n$ for natural numbers m, n with $m < n$, which makes $\mathcal{F}_n$ *richer* than $\mathcal{F}_m$. This effective richness of the family of parametrized functions is called the **capacity** of the approach. Hence, it is evident that the hyper-parameters of any ML approach control its capacity.

2.2.3. Performance Measure and Searching Optimal Parameters

Since ML approaches adapt their parameters from the data, one must be able to perform the following two steps: *i.* measure the performance, or the *goodness*, of the corresponding model with any parameter value, and *ii.* search for *better* parameter values than the current one based on the performance measure. The performance measure of any ML algorithm is calculated using a **learning objective**, commonly referred to as a **loss function**, which is defined in terms of the parametrized model, the inputs, and the labels (if they are available). As the name suggests, the lower the value of the loss function, the better is the performance of the model and vice versa. The notion of the loss function is formalized below.

Definition 2.2.6 (Loss Function). In a supervised learning setting with parametrizd predictive function f_θ and given dataset $D = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_i$, the loss function L_{sup} is a function of the terms from the set: $\{f_\theta(\mathbf{x}), y \mid (\mathbf{x}, y) \in D\}$. In an unsupervised learning setting with the latent representation $\mathbf{z}$ modeled in terms of parametrized function g_ϕ, the loss function L_{unsup} is a function of the terms from the set: $\{\mathbf{x}, g_\phi(\mathbf{x}) \mid \mathbf{x} \in D\}$.

Observe that with a given dataset D, the loss function L becomes the function of the parameter. In the context of Definition 2.2.6 above, $L_{\text{sup}} = L_{\text{sup}}(\theta)$ for the supervised learning setting and $L_{\text{unsup}} = L_{\text{unsup}}(\phi)$ for the unsupervised setting. Thus, the problem of searching for the optimal parameters translates to finding the `argmin` of the loss function. In ML, we invariably select loss functions that are differentiable with respect to their parameters. If the intended measure of performance is non-differentiable, we use its differentiable *surrogates* as the loss functions. Having a differentiable loss function $L = L(\theta)$ enables the computation of its **gradient** $\nabla_\theta L$ with respect to the parameters θ. Since the gradient of any function with respect to its parameters points towards the *direction* of the fastest growth of the function, the parameters of a learnable models can be improved by changing them by small amounts in the *opposite* direction of the gradient. This gives the so-called **gradient-descent algorithm** (Cauchy (1847)) for optimizing parameters.

Definition 2.2.7 (Gradient Descent (GD)). Consider an ML algorithm with parameters $\theta \in \Theta$, Θ being the set of all parameters, and loss function $L(\theta)$. Then, the problem of learning can be converted into the following optimization problem:

Learning Problem: Find parameter $\theta^* \in \Theta$ such that $\theta^* = \arg\min_{\theta\in\Theta} L(\theta)$

To minimize $L(\theta)$, we use the gradient descent algorithm in order to learn new better parameters θ' from the current parameters θ as follows:

Gradient Descent Algorithm: $\theta' \longleftarrow \theta - \eta \cdot \nabla_\theta L(\theta)$

Here, η is the **step-size** of the parameter update and it is a hyper-parameter. It dictates the amount by which the current parameters θ should be updated in the direction opposite to the gradient $\nabla_\theta L(\theta)$.

Remark 2.2.8 (Step-Size and Learning Dynamics). The step-size η is crucial in the learning dynamics. Too small a step-size can slow down the parameter updates, thereby slowing down the algorithm. On the other hand, too large a step-size can make the parameter values diverge. Thus, η is chosen after trying a set of proposal values and then selecting the best. Step-size schedules are also used for the better learning of parameters.

2.2.4. Generalization and Regularization

One of the most important aspects of learning algorithms is that they should be able to **generalize**. Generalization refers to the ability of a learning algorithm to perform well not only on the dataset used for its training but also perform equally well on *similar yet previously unseen* data points. If an algorithm performs well on the data that is used to train it but does not perform well on new data points, it is said to have **overfit** to the training data. Overfitting usually happens when the algorithm ends up modeling the randomness and extreme peculiarities in the inputs rather than learning the patterns in them. Such a condition is also termed as **memorization**, wherein the algorithm ends up memorizing peculiarities of the inputs. On the other extreme, the algorithm may not be capable of learning patterns sufficiently and ends up performing poorly even on the training data itself. In such cases, the algorithm is said to have **underfit** to the training data.

There is a connection between the capacity of a model, the size of the training dataset, and the generalization performance of the corresponding algorithm. Low capacity models are associated with function families that are not sufficiently rich. Thus, such models end up underfitting on larger training datasets. High capacity models are associated with overly expressive function families. Thus, such models end up memorizing and overfitting to smaller training datasets. Thus, it is important to ensure the right amount of training dataset and the right capacity of models in an ML algorithm for it to generalize better.

There are several ways to improve the generalization performance of ML models. Since hyper-parameters control the capacity of ML algorithms, **hyper-parameter tuning** can help allocate the correct capacity to the model and improve its generalizability. Another widely used approach is **regularization**, where additional constraints are imposed on the model parameters to enhance its generalization performance. Regularization is usually carried out by adding to the loss function a regularization term, which is also a function of the model parameters. This term penalizes different parameter choices differently, which results in preference of certain parameter choices over others during training. Thus, regularization gives a mechanism to introduce prior knowledge into the optimization process.

Definition 2.2.9 (Regularization Term). Consider a parametrized model with parameters $\theta \in \Theta$, which is trained with the loss function $L(\theta)$. A **regularizer** is a function R such that $R : \Theta \longrightarrow \mathbb{R}$. The model is then trained with the regularized loss $\widetilde{L}(\theta) = L(\theta) + \lambda \cdot R(\theta)$. The term $R(\theta)$ is the **regularization term**, and λ is the hyper-parameter controlling the effect of regularization.

Remark 2.2.10 (**$\mathcal{L}^2-$Regularizer and $\mathcal{L}^1-$Regularizer**)**.** The most commonly used regularizers are the $\mathcal{L}^2-$regularizer and the $\mathcal{L}^1-$regularizer. The $\mathcal{L}_L^2-$regularizer, given by $R(\theta) = \|\theta\|_2^2$, encourages the parameters to be closer to the origin and the $\mathcal{L}_1-$regularizer, given by $R(\theta) = \|\theta\|_1$, encourages learning of sparse parameters.

However, regularization is not restricted to the addition of regularizer terms to the loss function and can take many other forms. One such widely used technique is the **early stopping**. The core idea of early stopping is that the data available for training is split into train and validation splits. The model is trained until its performance improves not only on the train split but also on the validation split. As soon as the validation performance starts to deteriorate while the training performance keeps improving, the training is stopped. Another widely used technique is **data augmentation**, in which we add *plausible* perturbed versions of the input data points to the training dataset. This effectively increases the size of the training dataset, which can mitigate the problem of overfitting in high capacity models. In addition, there is a multitude of other regularization approaches like **noise injection** (Bishop (1995)), **ensemble techniques** like bagging (Breiman (1996)), and **dropout** (Srivastava et al. (2014)).

This concludes our discussion of the basic concepts and terminologies of ML. For further details, we encourage the readers to the excellent references books: Goodfellow et al. (2016), Bishop (2006), and Russell and Norvig (2002). Having considered the basics of ML, we turn our attention to its probabilistic formulations.

2.3. Probabilistic Machine Learning

2.3.1. The Need of Probabilistic Modeling

There are multiple sources of **noise** and **uncertainty** in any ML approach. Some of these sources are deliberate and beneficial, while some others are unavoidable and may even prove detrimental to the approach. Thus, there is a need to model and understand these sources of randomness in order to exploit the beneficial ones and to mitigate the detrimental ones. This is the reason for considering the probabilistic models of ML. Towards this, we begin with a discussion of some of these sources of randomness in ML algorithms.

- **The data can be noisy**. The dataset available for training of a model is often noisy. Usually, this noise is a result of the sensors used to collect the data. However, noise can also be added to the data points for the regularization of the model through data augmentation techniques. Thus, there is a need to model the noise in data and its consequences on the learning of the model.
- **The predictions of a model can involve uncertainties**. Certain tasks are better formulated with models that output a probability distribution over possible outputs. For instance, a spam detection algorithm may output the probability that an input e-mail is a spam or not. In some ML approaches, the intended output is better modeled as a **sample** conditioned on the inputs. For instance, in **reinforcement learning** problems, the agent is required to choose an action based on its observations. In **bandit learning** problems, an agent needs to learn to perform a downstream task from weak feedback signals. In such cases, sampling of outputs is known to enable better learning of the models. However, in problems like medical diagnosis, the output of an ML approach may have a direct impact on important aspects of human life. In such cases, it is imperative to estimate the uncertainties of the model predictions.
- **The optimization algorithms can be stochastic**. It is often required to have stochastic versions of optimization algorithms for scaling up the training of ML approaches. As we will see later, the gradient descent algorithm is often computationally expensive. For scaling up ML approaches, its stochastic version, the **stochastic gradient descent algorithm**, is used.
- **The estimation of model parameters can involve uncertainties**. As a result of the aforementioned uncertainties, the estimation of model parameters can involve uncertainties, which sometimes need to be estimated.
- **Most importantly, almost all ML problems have probabilistic formulations**. For instance, the supervised learning problem of learning a predictive function f_θ of inputs $\mathbf{x}$ to predict the labels y can be formulated as the problem of learning a parametrized conditional distribution $p_\theta(y \mid \mathbf{x})$ over the labels y given the inputs $\mathbf{x}$. Similarly, unsupervised learning problems can be formulated as the problems of

learning parametrized distributions over the inputs $\mathbf{x}$, possibly along with their latent representations $\mathbf{z}$, such as $p_\theta(\mathbf{x})$, $p_\theta(\mathbf{z} \mid \mathbf{x})$, or $p_\theta(\mathbf{x} \mid \mathbf{z})$.

In view of these points, the utility of modeling the uncertainties in ML approaches is evident. Thus, we begin with our discussion of the probabilistic formulation of ML.

2.3.2. Directed Graphical Models

One of the core themes of probabilistic models of ML approaches is that they view all the involved variables as **random variables/vectors** and model their probability distributions.

Definition 2.3.1 (**Data and Latent Representations**)**.** We denote by X and Z the random variables corresponding to the observable data and their corresponding latent representations. We assume that the dataset D is generated by taking samples from a fixed underlying distribution $p_{\text{data}}(\cdot)$ corresponding to the random variable X. Further, we assume that the samples taken from $p_{\text{data}}(\cdot)$ for generating the dataset D are **independent and identically distributed (i.i.d.)**. We denote this as follows:

$$D = \left\{ \mathbf{x}^{(i)} \mid \mathbf{x}^{(i)} \overset{\text{iid}}{\sim} p_{\text{data}}(\cdot) \right\}$$

Now, we define the probabilistic formulation of a model of the data. Depending on the choice of the problem, we are interested in either modeling only the observed data or both the observed data and its latent representations. For instance, in supervised learning, both the inputs $\mathbf{x}$ and labels y are observable and we want to learn *conditional* parametrized model $p_\theta(y \mid \mathbf{x})$. In density estimation, we want to learn a parametrized model $p_\theta(\mathbf{x})$ of only the data $\mathbf{x}$ and in the problem of uncovering the latent structure of the data, such as clustering, we are interested in parametrized models $p_\theta(\mathbf{z} \mid \mathbf{x})$ of the latent representations $\mathbf{z}$ conditioned on the inputs $\mathbf{x}$. Thus, in general, we are interested in learning the joint probability distribution over *all* the involved random variables. However, we can *simplify* this problem to learning parametrized conditional distributions, like $p_\theta(y \mid \mathbf{x})$ or $p_\theta(\mathbf{z} \mid \mathbf{x})$, or parametrized marginal distributions, like $p_\theta(\mathbf{x})$, depending on the exact problem.

Definition 2.3.2 (**Probabilistic Model**)**.** A parametrized probabilistic model of the data is a parametrized joint probability distribution $p_\theta(\mathbf{x}, \mathbf{z})$ over all involved random variables, $\mathbf{x}$ being the variables observable and $\mathbf{z}$ being the latent variables.

Note that the specification of a model needs additional information in the form of the way different observable and latent random variables depend on each other. Such a specification of the model is called the probabilistic graphical model, which models each of the random variables as a node of a graph and specifies the dependencies in terms of the directed edges from one node to the another. We restrict the choice of the graph to be a **directed acyclic graphs**. With this, we formalize the notion of directed graphical models below.

Definition 2.3.3 (**Directed Graphical Models (DGM)**)**.** Let $G = (X, E)$ be a directed acyclic graph (DAG). Let the set of vertices $X = \left\{X^{(1)}, \ldots, X^{(n)}\right\}$ be the set of all random variables of an ML approach. Let $E = \{(\mathbf{v} \longrightarrow \mathbf{w}) \mid \mathbf{v}, \mathbf{w} \in X\}$ be the set of directed edges of G. Consider the parent function $\pi : X \longrightarrow 2^X$ over the graph G, which inputs a vertex of the graph and returns the set of its parents, i.e., $\forall\ \mathbf{x} \in X$, $\pi(\mathbf{x}) = \{\mathbf{u} \mid \mathbf{u} \in X, (\mathbf{u} \longrightarrow \mathbf{x}) \in E\}$. Then, the directed graphical model (DGM) corresponding to the DAG G entails that the model $p\left(X^{(1)}, \ldots, X^{(n)}\right)$ factorizes as follows:

$$p\left(X^{(1)}, \ldots, X^{(n)}\right) = \textstyle\prod_{i=1}^{n} p\left(X^{(i)} \mid \pi\left(X^{(i)}\right)\right)$$

Thus, the DGM states that in the model, the random variable $X^{(i)}$ is dependent on only those random variables that form its parent in the DAG G in the sense that $X^{(i)}$ is conditionally independent of any other random variables given $\pi\left(X^{(i)}\right)$.

2.3.3. Estimation and Inference

The problem of learning in probabilistic models can take many forms but there are two major, but slightly different, themes. As seen before, the probabilistic model is defined as a parametrized distribution $p_\theta(\mathbf{x}, \mathbf{z})$ over all the observable variables $\mathbf{x}$ and latent variables $\mathbf{z}$. In the first theme, we assume that we have access to the datasets corresponding to the values of all the involved variables. Then, the aim is to estimate the *value* of the parameter θ

of the model, which is called the **estimation** problem. In the second theme, we only assume that we have access to observable data. Then, the aim is to predict the *distribution* over the latent variables $\mathbf{z}$ from the observable variables $\mathbf{x}$, which is called the **inference** problem.

2.3.4. Maximum Likelihood Estimation

We begin with the simpler problem of estimation in a probabilistic model. We are given with a dataset $D = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}$ and a parametrized model $p_\theta(\mathbf{x})$ for this dataset. Our aim is to train the model with the dataset to estimate the *optimal* parameters. In **maximum likelihood estimation (MLE)**, we assume that the optimal parameter is the one that *explains* the given data well. Thus, we set the learning objective to be the maximization of the dataset likelihood $p_\theta(D)$, or equivalently, the **dataset log-likelihood** $\log p_\theta(D)$.

Definition 2.3.4 (**Maximum Likelihood Estimation (MLE)**)**.** Given a parametrized model $p_\theta(\mathbf{x})$ and dataset $D = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}$, the optimal parameter $\hat{\theta}^{\text{MLE}}$ corresponding to the maximum likelihood estimation is defined as:

$$\begin{aligned}\hat{\theta}^{\text{MLE}} &= \arg\max_{\theta\in\Theta} \log p_\theta(D) \equiv \arg\max_{\theta\in\Theta} \frac{1}{n}\log p_\theta(D)\\ &= \arg\max_{\theta} \log p_\theta\left(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}\right) =^{*} \arg\max_{\theta\in\Theta} \frac{1}{n}\sum\nolimits_{i=1}^{n} \log p_\theta\left(\mathbf{x}^{(i)}\right)\end{aligned}$$

Here, Θ is the set of permissible parameters and the equality marked $(*)$ follows from the i.i.d. data assumption, as given in Definition 2.3.1.

Now, note that GD can be used to minimize the *negative* of dataset log-likelihood in order to compute $\hat{\theta}^{\text{MLE}}$ from randomly initialized θ. In particular, the update step becomes:

$$\begin{aligned}&\textbf{Loss: } L(\theta) = \frac{1}{n}\log p_\theta(D) = \frac{1}{n}\sum\nolimits_{i=1}^{n} \log p_\theta\left(\mathbf{x}^{(i)}\right)\\ &\textbf{Update: } \theta \longleftarrow \theta - \eta\cdot\nabla_\theta L(\theta) = \theta - \eta\cdot\frac{1}{n}\sum\nolimits_{i=1}^{n} \nabla_\theta \log p_\theta\left(\mathbf{x}^{(i)}\right)\end{aligned} \tag{2.3.1}$$

Thus, each update step of the GD for computing $\hat{\theta}^{\text{MLE}}$ requires computing n gradients: $\nabla_\theta \log p_\theta\left(\mathbf{x}^{(i)}\right)$. This step is computationally expensive for large datasets and thus, it *does not scale up*; it is computationally infeasible to use GD for training models on large datasets. Here, we deliberately introduce uncertainty in the update step for scaling it up.

Definition 2.3.5 (**Stochastic Gradient Descent (SGD)**)**.** The **stochastic gradient descent (SGD)** update samples a **batch** $\{\mathbf{x}^i\}_{i=1}^B$ of size B from the dataset D, with $B \ll n$, and performs the update on the model parameters.

$$\theta \longleftarrow \theta - \eta \cdot \frac{1}{B} \sum\nolimits_{\mathbf{x}^i \overset{\text{iid}}{\sim} D,\ i \in \{1,\dots,B\}} \nabla_\theta \log p_\theta \left(\mathbf{x}^i\right)$$

Note that the SGD approach utilizes a surrogate loss $\hat{L}_\theta^{\text{SGD}}$ evaluated only on a sampled batch, and then uses its gradients to update the parameters θ, which is formalized below.

$$\begin{aligned} &\textbf{Loss: } \hat{L}_\theta^{\text{SGD}} = \frac{1}{B} \sum\nolimits_{\mathbf{x}^i \overset{\text{iid}}{\sim} D,\ i \in \{1,\dots,B\}} \nabla_\theta \log p_\theta \left(\mathbf{x}^i\right) \\ &\textbf{Update: } \theta \longleftarrow \theta - \eta \cdot \nabla_\theta \hat{L}_\theta^{\text{SGD}} = \theta - \eta \cdot \frac{1}{B} \sum\nolimits_{\mathbf{x}^i \overset{\text{iid}}{\sim} D,\ i \in \{1,\dots,B\}} \nabla_\theta \log p_\theta \left(\mathbf{x}^i\right) \end{aligned} \tag{2.3.2}$$

This introduction of sampling is an example of the need for stochasticity in a beneficial manner; in the SGD updates, it enables scaling up of the algorithm for large datasets via performing updates with relatively smaller batches of data.

2.3.5. Bayesian Inference

From Definition 2.3.4 and Equations 2.3.1, 2.3.2, it is evident that MLE has many limitations.

- $\hat{\theta}^{\text{MLE}}$ **is a point estimate of the optimal parameters**; we obtain one value of the parameter based on the dataset and the optimization algorithm. Thus, the point estimate does not model the uncertainty in the estimated parameter value, which is a result of the data noise as well as the stochasticity in the optimization.
- $\hat{\theta}^{\text{MLE}}$ **does NOT allow for the incorporation of extra information known about the parameters**. In particular, it does not allow the incorporation of *prior knowledge* about the parameters into the optimization process.

The second point of incorporating the prior knowledge is taken care of by the **maximum a posteriori estimation (MAP)**, which assumes that the optimal parameter is the one that is best *explained by* the given data. Thus, we set the learning objective to be the maximization of the distribution $p(\theta \mid D)$, or equivalently, the maximization of $\log p(\theta \mid D)$.

Definition 2.3.6 (Maximum A Posteriori Estimation (MAP)). Given a model with parameter θ, dataset $D = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}$, and a distribution over the parameters $p(\theta)$ representing the prior knowledge of the parameters, the optimal parameter $\hat{\theta}^{\text{MAP}}$ corresponding to the maximum a posteriori estimation is defined as:

$$\begin{aligned}\hat{\theta}^{\text{MAP}} &= \arg\max_{\theta\in\Theta} \log p(\theta \mid D) =^{*} \arg\max_{\theta\in\Theta} \log \frac{p(D \mid \theta)\cdot p(\theta)}{p(D)} \\ &\equiv^{\dagger} \arg\max_{\theta\in\Theta} \log p(D \mid \theta)\cdot p(\theta) = \arg\max_{\theta\in\Theta} \log p(D \mid \theta) + \log p(\theta)\end{aligned}$$

Here, Θ is the set of all permissible parameters. The equality $(*)$ follows from **Bayes' rule**, and the equality $(\dagger)$ follows from the fact that $p(D)$ is a constant during the optimization of θ.

The benefit of MAP estimation over MLE is that when we have the prior knowledge over the parameters θ in the form of a distribution $p(\theta)$, it can be incorporated into the optimization process. However, the MAP estimate $\hat{\theta}^{\text{MAP}}$ is also a point estimate and thus, similar to the MLE, it does not model the uncertainty over the estimated parameter value either. To model the uncertainty in the estimated parameter value, or any other variable of interest, it is important to model a distribution over it. This is the so-called **Bayesian inference**.

Definition 2.3.7 (Bayesian Inference). Consider a variable of interest θ, a model parameter or the latent representation of data, for which we want to model the uncertainty. Let the prior knowledge of θ be available in terms of a distribution $p(\theta)$. Suppose we have access to the data D and a model $p(\theta \mid D)$ representing the likelihood of the data given θ. Then, Bayesian inference refers to the inference technique of modeling the distribution over θ given the data D as follows.

$$p(\theta \mid D) = \frac{p(D \mid \theta)\cdot p(\theta)}{p(D)} =^{*} \frac{p(D \mid \theta)\cdot p(\theta)}{\int_{\theta'} p(D \mid \theta')\cdot p(\theta')\,\mathrm{d}\theta'}$$

Here, $p(\theta)$ is referred to as the **prior**, $p(D \mid \theta)$ as the **likelihood**, $p(D)$ as the **data marginal**, and $p(\theta \mid D)$ as the **posterior**. Further, the equality $(*)$ follows from using the marginalization of the variable θ' in the denominator. If the involved variables are discrete, the integral sign of the equality above should be replaced by the summation.

Remark 2.3.8 (MAP Estimation and Bayesian Inference). Bayesian inference essentially refers to the usage of Bayes' rule in order to model the *complete* posterior distribution over the variable of interest θ using the given data D. Thus, from Definitions 2.3.6 and 2.3.7, we can see that the MAP point estimate is just the `argmax` of the complete posterior distribution $p(\theta \mid D)$ modeled using Bayesian inference.

It is clear that Bayesian inference generalizes point estimates, like MAP, by modeling the complete posterior distribution. However, Bayesian inference does require the computation of the data marginal term $p(D) = \int_{\theta'} p(D \mid \theta') \cdot p(\theta') \, \mathrm{d}\theta'$. It so happens that in many popular choices of prior and likelihood models, the computation of this marginal term, and hence that of the posterior term, is not easy. Often, in the case of continuous variables, the marginal does not have an *analytic solution*, which means that there is no closed-form expression for the integral. Also, in the case of discrete variables, the corresponding summation may be computationally expensive as it might involve computing exponentially many terms. These conditions are referred to as *intractabilities*. Thus, the data marginal $p(D)$, and consequently the posterior $p(\theta \mid D)$, is computationally **intractable** for many widely used parametrizations. Thus, despite being highly powerful, Bayesian learning does require different techniques that help overcome these intractabilities.

One of the approaches to tackle the intractabilities in the evaluation of the posterior distribution is to convert this problem of learning the posterior distribution into an optimization problem. The core idea is that if the exact posterior distribution is the solution of an optimization problem, then we can *approximately solve* that optimization problem to learn an approximation to the true posterior distribution. **Variational inference (VI)** is one such approach, which forms the basis of our research work and so, we discuss it next.

2.3.6. Variational Inference

Variational inference (VI) is an approximate inference approach for generative latent variable models. VI utilizes a family of **parametrized proposal distributions** in order to model the true intractable posterior distribution. Since we use a proposal distribution to model a true posterior distribution, it is natural to set their KL-divergence as the optimization objective. This is what is done in VI, which is formalized below.

Definition 2.3.9 (Variational Inference (VI)). Consider the task of learning the latent representations $\mathbf{z}$ of input data $\mathbf{x}$ by learning the posterior distribution $p(\mathbf{z} \mid \mathbf{x})$, which is intractable. Consider a family $\mathcal{F} = \{q_\phi(\mathbf{z} \mid \mathbf{x}) \mid \phi \in \Phi\}$ of parametrized proposal distributions $q_\phi(\mathbf{z} \mid \mathbf{x})$ for modeling the true posterior distribution $p(\mathbf{z} \mid \mathbf{x})$, where Φ is the set of all permissible parameters. Then, VI defines the approximate inference problem as an optimization problem, where the task is to find the optimal parameter $\phi^* \in \Phi$ that minimizes the KL-divergence $\mathcal{D}_{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) || p(\mathbf{z} \mid \mathbf{x}))$.

VI Objective: Find $\phi^* \in \Phi$ such that $\phi^* = \arg\min_{\phi\in\Phi} \mathcal{D}_{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) || p(\mathbf{z} \mid \mathbf{x}))$, i.e.,

$$\text{Minimize } \mathcal{D}_{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) || p(\mathbf{z} \mid \mathbf{x})) \text{ with respect to } \phi \in \Phi \text{ to obtain } \phi^*$$
$$\text{so that } q_{\phi^*}(\mathbf{z} \mid \mathbf{x}) \text{ is the desired approximation to the true posterior } p(\mathbf{z} \mid \mathbf{x}).$$

Remark 2.3.10 (Amortization). One can also utilize a parametrized proposal family $\mathcal{F} = \{q_\phi(\mathbf{z}) \mid \phi \in \Phi\}$ for modeling the posterior $p(\mathbf{z} \mid \mathbf{x})$ and define the VI objective as the task of finding $\phi^\dagger \in \Phi$ such that $\phi^\dagger = \arg\min_{\phi\in\Phi} \mathcal{D}_{KL}(q_\phi(\mathbf{z}) || p(\mathbf{z} \mid \mathbf{x}))$. In this case, we would be implicitly allowing the learning of different optimal parameters for the input data $\mathbf{x}$ independent of the other input data points. On the other hand, the VI objective of Definition 2.3.9, we force the optimal parameters to be *shared* across data points by trying to learn to optimize parameters for the entire dataset. This later approach of trying to model a conditional proposal distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$ rather than modeling the proposal $q_\phi(\mathbf{z})$ individually for each data point is known as **amortization**. Naturally, amortization slightly lowers the performance of the VI approach but allows it to be scalable to large datasets, where learning optimal parameters per data point is infeasible. Thus, we will focus on the amortized version of the VI approach throughout our discussions, unless specified otherwise.

From Definition 2.3.9, it is clear that the optimization objective $\mathcal{D}_{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) || p(\mathbf{z} \mid \mathbf{x}))$ can not be computed directly as it is defined in terms of the intractable posterior $p(\mathbf{z} \mid \mathbf{x})$.

Thus, we perform manipulations to obtain an equivalent trainable optimization objective.

$$\begin{aligned}\mathcal{D}_{KL}\left(q_\phi\left(\mathbf{z} \mid \mathbf{x}\right)||p\left(\mathbf{z} \mid \mathbf{x}\right)\right) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{q_\phi\left(\mathbf{z} \mid \mathbf{x}\right)}{p\left(\mathbf{z} \mid \mathbf{x}\right)}\right] =^{*} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{q_\phi\left(\mathbf{z} \mid \mathbf{x}\right)}{\frac{p(\mathbf{x}|\mathbf{z})\cdot p(\mathbf{z})}{p(\mathbf{x})}}\right] \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{q_\phi\left(\mathbf{z} \mid \mathbf{x}\right)}{p\left(\mathbf{z}\right)}\right] - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p\left(\mathbf{x} \mid \mathbf{z}\right)\right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p\left(\mathbf{x}\right)\right] \\ &=^{\dagger} \mathcal{D}_{KL}\left(q_\phi\left(\mathbf{z} \mid \mathbf{x}\right)||p\left(\mathbf{z}\right)\right) - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p\left(\mathbf{x} \mid \mathbf{z}\right)\right] + \log p\left(\mathbf{x}\right)\end{aligned} \tag{2.3.3}$$

Here, the equality $(*)$ follows from Bayes' rule and the equality $(\dagger)$ follows from the fact that $\log p\left(\mathbf{x}\right)$ is a constant in the expectation with respect to $q_\phi\left(\mathbf{z} \mid \mathbf{x}\right)$.

Now, from Equation 2.3.3, we can see that the minimization of $\mathcal{D}_{KL}\left(q_\phi\left(\mathbf{z} \mid \mathbf{x}\right)||p\left(\mathbf{z} \mid \mathbf{x}\right)\right)$ is equivalent to the maximization of the expression $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p\left(\mathbf{x} \mid \mathbf{z}\right)\right] - \mathcal{D}_{KL}\left(q_\phi\left(\mathbf{z} \mid \mathbf{x}\right)||p\left(\mathbf{z}\right)\right)$ as the term $\log p\left(\mathbf{x}\right)$ is a constant in the optimization with respect to $\phi \in \Phi$. Further, we can utilize parametrized **decoder** $p_\theta\left(\mathbf{x} \mid \mathbf{z}\right)$ and **prior** $p_\theta\left(z\right)$ in the final expression, which allows us to compute and optimize it using gradient descent/ascent methods. This equivalent optimization objective is called the **evident lower bound (ELBO)**.

Definition 2.3.11 (**Evidence Lower Bound (ELBO)**)**.** The expression $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta\left(\mathbf{x} \mid \mathbf{z}\right)\right] - \mathcal{D}_{KL}\left(q_\phi\left(\mathbf{z} \mid \mathbf{x}\right)||p_\theta\left(\mathbf{z}\right)\right)$ is called the evidence lower bound (ELBO). The $q_\phi\left(\mathbf{z} \mid \mathbf{x}\right)$ is called the **encoder**, the **inference model**, or the **posterior**, $p_\theta\left(\mathbf{x} \mid \mathbf{z}\right)$ is called the **decoder**, and $p_\theta\left(\mathbf{z}\right)$ is the parametrized **prior**.

2.4. Neural Networks

One of the major factors responsible for the advent of deep learning is the advances in (deep) neural networks and their optimization. **Neurons** are computational units inspired from the biological neurons, which are capable of *learning* based on *experience.* Neurons perform computations defined in terms of trainable parameters and non-linearities to produce outputs from their inputs. **Neural networks** consist of *layers* of such trainable neurons, where the outputs of the current layers are the inputs to the next layer. Given the inputs, the outputs of the neural network are computed and are compared against the ground-truth values to compute the *loss*, which is the measure of performance. The updates of the trainable parameters can be computed using the **backpropagation algorithm**, where the updates are computed sequentially from the last layer towards the first layer using the **chain rule**. Thus, neural networks are considered to be capable of **end-to-end learning**. **Deep neural networks** usually refer to neural networks with many layers of neurons and they form the state-of-the-art approaches in many important machine learning problems. In this section, we discuss some of these neural networks that we use in our experimentation.

2.4.1. Feed-Forward Neural Networks

Feed-forward neural networks (FFNN) consist of layers of neurons such that every neuron of a layer is connected with every neuron of the next layer. For the layer i, a linear combination of its inputs $\mathbf{a}^{(i)}$ is first computed, which is then passed through a non-linearity σ in order to obtain the output $\mathbf{a}^{(i+1)}$ of this layer. We will refer to this computation carried out by a neural network as its **forward pass**. The forward pass of the FFNN is given below.

Definition 2.4.1 (Forward Pass of Feed-Forward Neural Network).
The computations in layer i of a FFNN: $\mathbf{a}^{(i+1)} = \sigma^{(i)}\left(\mathbf{W}^{(i)} \cdot \mathbf{a}^{(i)} + \mathbf{b}^{(i)}\right)$, where

$\mathbf{a}^{(i)} \in \mathbb{R}^{d^{(i)}}$ is d^i-dimensional input of layer i

$\mathbf{a}^{(i+1)} \in \mathbb{R}^{d^{(i+1)}}$ is $d^{i+1}-$dimensional output of the layer i

$\mathbf{W}^{(i)} \in \mathbb{R}^{d^{(i+1)} \times d^i}$ is the **weight matrix** and $\mathbf{b}^{(i)} \in \mathbb{R}^{d^{i+1}}$ is the **bias** of the layer i

$\sigma^{(i)} : \mathbb{R}^{d^{(i+1)}} \longrightarrow \mathbb{R}^{d^{(i+1)}}$ is the **non-linearity** of layer i

Remark 2.4.2 (Non-Linearities and Examples). Non-linearities in neural networks are essential. Without non-linearities, a FFNN computes only a linear function of its inputs and it is impossible to model sufficiently many function families using only the linear functions. The most widely used non-linearities are **sigmoid**, **tanh**, and **rectified linear units (ReLU)**. In our experimentation, we use the sigmoid and tanh activations as well as a variant, named **exponential linear unit (ELU)**, of the ReLU activation.

$$\text{For } x \in \mathbb{R}, \quad \begin{array}{ll} \operatorname{sigmoid}(x) = \frac{1}{1+e^{-x}}, & \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \\ \operatorname{ReLU}(x) = \begin{cases} x & \text{if } 0 < x \\ 0 & \text{otherwise} \end{cases} & \operatorname{ELU}(x) = \begin{cases} x & \text{if } 0 < x \\ \alpha \cdot (e^x - 1) & \text{otherwise} \end{cases} \end{array} \tag{2.4.1}$$

2.4.2. Convolutional Neural Networks

In order to process data with specific structures and patterns, it is imperative to utilize neural networks with appropriate **stuctural priors**. Structural prior refers to the choices in the neural network architecture that cater to the specific characteristics of the input data. One of the major sources of data in real-world applications is visual inputs like images and videos. The processing of visual inputs requires many considerations, which also point at the limitations of processing images with FFNNs.

- **Images are represented as 2-dimensional arrays and contain spatial patterns**. Note that although FFNNs are powerful function approximators, they must input data in the form of a fixed dimensional vector. Consequently, if one intends to process images, where the data has a 2-dimensional structure and the patterns, one needs to convert this data into a fixed-length vector format. This *flattening* of the data into a vector can lead to loss of the spatial information in the input images.
- **Images are extremely high-dimensional inputs**. Usually, images contain approximately 10^4 to 10^5 *pixels.* If the image is colored, each pixel contains 3 scalar values corresponding to the three channels R, G, and B. Thus, images are extremely high-dimensional data, and processing them directly with FFNNs would require an extremely large number of trainable parameters, which is computationally expensive.
- **Images contain information that is usually invariant to certain transformations**. For instance, images transformed with small translations, scaling, rotation, and illumination usually do not change the output of the neural network for the required task at hand. It is not guaranteed that the parametrization in FFNNs can cater for and exploit these desired invariances.

Convolutional neural networks (CNNs) aim to incorporate these invariances into the neural network architectures. CNNs consist of convolutional layers, that are often followed by pooling layers. Instead of layers of neurons that are connected to all the neurons of the next layer, the convolutional layers incorporate **parameter sharing** by using **kernels** to process the images. Kernels are small parametrized matrices, which are discrete-convolved with the input features to the layer to generate the output features. The **pooling** layers are used to subsample every local-region of an input feature based on certain criteria to induce an invariance to small translations as well as to further reduce the number of trainable parameters. The forward pass of a convolutional layer is given below.

Definition 2.4.3 (Forward Pass of Convolutional Layer).

The computations for one kernel in a convolutional layer: $\mathbf{a}^{(\text{out})} = \mathbf{K} * \mathbf{a}^{(\text{in})} + \mathbf{b}$,

where $\mathbf{a}^{(\text{in})}$ is the 2-dimensional feature map input to the layer

$\mathbf{a}^{(\text{out})}$ is the 2-dimensional feature map output from the layer

$\mathbf{K} \in \mathbb{R}^{k \times k}$ is the kernel and $\mathbf{b}$ is the **bias** of the layer

$*$ is the **convolution operation**: $\left(\mathbf{K} * \mathbf{a}^{(\text{in})}\right)_{n,m} = \sum_{0 \leq i,j \leq k-1} \mathbf{a}^{(\text{in})}_{n+i,m+j} \cdot \mathbf{K}_{k-i,k-j}$

2.4.3. Residual Neural Networks

Despite the great success of deep neural networks, it was observed that the training of very deep neural networks was notoriously hard. As the networks grew deeper, obtaining informative gradients in the initial layers during the training process was difficult. However, in principle, training of an $(n+1)-$layered neural network $\mathcal{N}_{n+1}$ should result in a network that performs equally well, if not better, than the training of an $n-$layered neural network $\mathcal{N}_n$. This is because appending an identity layer to an $n-$layered neural network gives an $(n+1)-$layered neural network with identical performance. This insight led to the notion of **skip-connections** in neural networks, which decompose the problem of learning deeper layer representations into the problem of learning a residual non-linear change to the identity. In particular, the output of a skip-connection layer computes the feature from the inputs, which are passed through a non-linearity, and are added to the input itself in order to generate the outputs. Such neural networks are referred to as the **residual neural networks**, for which the generic forward pass is described below.

Definition 2.4.4 (**Forward Pass of Residual Neural Network Block**).

The computations of a residual layer: $\mathbf{a}^{(\text{out})} = \sigma\left(f_\phi\left(\mathbf{a}^{(\text{in})}\right)\right) + \mathbf{a}^{(\text{in})}$, where

$\mathbf{a}^{(\text{in})}$ is the input to and $\mathbf{a}^{(\text{out})}$ is the output of the residual layer

f_ϕ is the feature extractor and σ is the non-linearity of the layer

$\sigma\left(f_\phi\left(\mathbf{a}^{(\text{in})}\right)\right)$ is the residual non-linear change to the skip-connectioned input $\mathbf{a}^{(\text{in})}$

2.4.4. Recurrent Neural Networks

Many real-world sources of data are variable-length sequences that additionally possess temporal structures and patterns. For example, videos are sequences of images, audio signals are sequences of sound patterns, sentences are sequences of words, and in our particular case, we aim at sequentially sampling of latent variables. **Recurrent neural networks (RNNs)** are the neural network architectures that incorporate the required structural prior for learning temporal structures and patterns in the data. In particular, for a sequence of data inputs $\left\{x^{(t)}\right\}_{t\geq 0}$, recurrent neural networks enable modeling parametrized conditional distributions over the next input given the previous inputs, which are of the form[2] $p_\theta\left(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)}, \ldots, \mathbf{x}^{(1)}\right)$. The vanilla RNNs, which were initially used to learn such models, achieve this modeling by keeping track of a hidden state $\mathbf{h}^{(t)}$ at each time instant t $(t \geq 0)$, which models the context required to process the current input. These RNNs process the input $\mathbf{x}^{(t+1)}$ along with this state $\mathbf{h}^{(t)}$ to predict the next input $\mathbf{x}^{(t+2)}$ as well as to update the hidden state to $\mathbf{h}^{(t+1)}$. However, these models were observed to suffer from the problem of **vanishing or exploding gradients**. Since RNNs involve performing repeated computations with the same shared weights, the gradients with repsect to the involved parametrization are difficult to maintain in a bounded region. This results in one of the major challenges in such modeling of temporal patterns; vanilla RNNs can not readily learn patterns over long-term intervals. **Long short-term memory networks (LSTMs)** and **gated recurrent units (GRUs)** are the extensions of vanilla RNNs that introduce the notion of processing the inputs and the hidden state via **gates** in order to mitigate the

[2]In the context of natural language processing, where each $\mathbf{x}^{(t)}$ is a word, this conditional model refers to the **language model**.

issues of vanishing or exploding gradients, which in turn allows for learning fairly long-term dependencies. In our experimentation, we utilize the GRUs as they perform equally well as the LSTMs but have simpler gating structure, which is described below.

Definition 2.4.5 (**Forward Pass of Gated Recurrent Unit**).

Reset gate r: $\mathbf{r}^{(t)} = \text{sigmoid}\left(\mathbf{W}^{(\text{ir})} \cdot \mathbf{x}^{(t)} + \mathbf{b}^{(\text{ir})} + \mathbf{W}^{(\text{hr})} \cdot \mathbf{h}^{(t-1)} + \mathbf{b}^{(\text{hr})}\right)$

Update gate z: $\mathbf{z}^{(t)} = \text{sigmoid}\left(\mathbf{W}^{(\text{iz})} \cdot \mathbf{x}^{(t)} + \mathbf{b}^{(\text{iz})} + \mathbf{W}^{(\text{hz})} \cdot \mathbf{h}^{(t-1)} + \mathbf{b}^{(\text{hz})}\right)$

New gate n: $\mathbf{n}^{(t)} = \tanh\left(\mathbf{W}^{(\text{in})} \cdot \mathbf{x}^{(t)} + \mathbf{b}^{(\text{in})} + \mathbf{r}^{(t)} \star \left(\mathbf{W}^{(\text{hn})} \cdot \mathbf{h}^{(t-1)} + \mathbf{b}^{(\text{hn})}\right)\right)$

Hidden state h: $\mathbf{h}^{(t)} = \left(1 - \mathbf{z}^{(t)}\right) \star \mathbf{n}^{(t)} + \mathbf{z}^{(t)} \star \mathbf{h}^{(t-1)}$

Here, $\mathbf{W}$ and $\mathbf{b}$ are appropriately shaped weights and biases corresponding to the gates and $\star$ represents the Hadamard product.

2.4.5. Backpropagation

The update of parameters in deep neural networks having multiple layers can be efficiently carried out using the **backpropagation algorithm**. The backpropagation algorithm computes the updates in parameter values with **chain rule** applied sequentially from the last layer towards the first layer. This allows the usage of already computed gradients to be reused for the parameter updates in the lower layers in a recursive manner.

This completes our discussion of the relevant neural network architectures. In the next Chapter, we begin with the first contribution of our work. We derive the theoretical properties of variational inference based on multi-sample estimators and prove a simpler bound on their variational gap.

Chapter 3

Properties of Multi-Sample Variational Inference and their Variational Gaps

3.1. Estimator View of Variational Inference

In this section, we consider an estimator-based view of the variational inference approach. Previously, we have seen that variational inference is an optimization problem that tries to learn a proposal $q_\phi(\mathbf{z})$, or $q_\phi(\mathbf{z} \mid \mathbf{x})$ if amortized, for the true intractable posterior $p(\mathbf{z} \mid \mathbf{x})$. The parameter ϕ can be learned by setting the optimization objective as the minimization of the KL-divergence term $\mathcal{D}_{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) || p(\mathbf{z} \mid \mathbf{x}))$, which is equivalent to the maximization of the evidence lower bound (ELBO) $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[p(\mathbf{x} \mid \mathbf{z})] - \mathcal{D}_{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) || p(\mathbf{z}))$. However, we can arrive at the same ELBO optimization objective in another manner, which we demonstrate below. Note that one of the ways of learning unsupervised representations of data X is by modeling a probability distribution over the data. Mathematically, we want to learn a parametrized distribution $p_\theta(\cdot)$ over the space of data points $\mathcal{X}$, which can input a sample $\mathbf{x} \in \mathcal{X}$ and output its probability density or probability mass depending on whether the random variable of the data X is continuous or discrete[1]. With the success of variational auto-encoders and other deep variational inference techniques, this parametrization is often actuated in terms of neural networks. However, now the question is what should be the optimization objective. Intuitively, we want our model $p_\theta(\cdot)$ to assign

[1]For the sake of brevity, we will always consider the data X to be continuous and consequently, talk about probability density. However, unless specified otherwise, these discussions would continue to hold by replacing the probability density with probability mass when the data X is discrete.

high probability density to the data points $\mathbf{x} \sim X$. Thus, we set the optimization objective as the maximization of the data likelihood.

Definition 3.1.1 (Variational Inference Optimization Objective). Given data X and a parametrization family Θ, the variational inference optimization objective is:

$$\textbf{maximize } p_\theta(\mathbf{x}) \text{ for } \mathbf{x} \sim X \textbf{ subject to } \theta \in \Theta.$$

Remark 3.1.2 (Log-Likelihood Maximization). Since $\log(\cdot)$ is a monotonically increasing function, we can equivalently try to maximize $\log p_\theta(\mathbf{x})$, which is the **log-likelihood of data**, instead of maximizing the **likelihood of data** $p_\theta(\mathbf{x})$ directly. The advantage of maximizing the log-likelihood of data will be evident from the subsequent discussion.

Now, note that exactly how to optimize the parameter θ of $p_\theta(\mathbf{x})$ is not evident from the definition of the optimization objective itself. Thus, we introduce the latent variable Z such that $\mathbf{z} = \mathbf{z}(\mathbf{x})$ is the latent representation of the data point $\mathbf{x}$. The introduction of the latent variable not only gives a trainable objective but also enables learning a generative model $p_\theta(\mathbf{x} \mid \mathbf{z})$ of the data. With the latent variables, we obtain a lower bound to the maximization objective $\log p_\theta(\mathbf{x})$, which can then be easily parametrized and optimized.

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \log \int_{\mathbf{z}\in\mathcal{Z}} p_\theta(\mathbf{x},\mathbf{z})\, \mathrm{d}\mathbf{z} = \log \int_{\mathbf{z}\in\mathcal{Z}} q_\phi(\mathbf{z} \mid \mathbf{x}) \cdot \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\, \mathrm{d}\mathbf{z} = \log \mathbb{E}_{q_\phi(\mathbf{z}\mid\mathbf{x})}\left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\right] \\ &\geq^* \mathbb{E}_{q_\phi(\mathbf{z}\mid\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\right] = \mathbb{E}_{q_\phi(\mathbf{z}\mid\mathbf{x})}\left[p_\theta(\mathbf{x} \mid \mathbf{z})\right] - \mathcal{D}_{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, p_\theta(\mathbf{z})\right) \quad (3.1.1)\end{aligned}$$

Here, the inequality $(*)$ follows from Jensen's inequality applied to the concave function $\log(\cdot)$. Note that the last expression in Equation 3.1.1 contains exactly the same terms as the ELBO derived earlier in Definition 2.3.11. Since $\log p_\theta(\mathbf{x})$ is a maximization objective, we aim to maximize this lower bound ELBO and expect that this will also improve the data log-likelihood $\log p_\theta(\mathbf{x})$ and consequently the data likelihood $p_\theta(\mathbf{x})$.

Now, for a data point $\mathbf{x}$, if we set the **estimator** $\hat{w} = \hat{w}(\mathbf{z}) = \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}\mid\mathbf{x})}$, then we get the following two of its properties from Equations 3.1.1. **1.** $\mathbb{E}[\hat{w}] = p_\theta(\mathbf{x})$ and **2.** $\log p_\theta(\mathbf{x}) \geq \mathbb{E}[\log \hat{w}]$. These observations demonstrate the core idea of variational inference and thus, we formalize these observations into the following definitions.

Definition 3.1.3 (Unbiased Estimator and Variational Lower Bound). Let $\mathcal{X}, \mathbf{x} \in \mathcal{X}$, and X denote the space of input data, a data point, and the random variable corresponding to the data point respectively. Let Z^i $(i \in \{1, \ldots, n\})$ be the n latent variables used to model the data distribution, $\mathcal{Z}^i$ be the space of the latent variable Z^i, and $\mathbf{z}^i$ represent a sample corresponding to Z^i. Consider a function $\hat{w}^n : \mathcal{X} \times \mathcal{Z}^1 \times \ldots \times \mathcal{Z}^n \longrightarrow \mathbb{R}_0^+$. Let $q\left(\mathbf{z}^1, \ldots, \mathbf{z}^n \mid \mathbf{x}\right)$ be the joint distribution over the latents. Then, **1.** $\hat{w}^n$ is called an **unbiased estimator** of $p\left(\mathbf{x}\right)$ iff $\mathbb{E}_{q(\mathbf{z}^1,\ldots,\mathbf{z}^n|\mathbf{x})}\left[\hat{w}^n\right] = p\left(\mathbf{x}\right)$ for any $\mathbf{x}$, and **2.** for an unbiased estimator $\hat{w}$ of $p\left(\mathbf{x}\right)$, we denote the corresponding **variational lower bound (VLB)** by $\hat{L}$ and define it as $\hat{L} = \mathbb{E}_{q(\mathbf{z}^1,\ldots,\mathbf{z}^n|\mathbf{x})}\left[\log \hat{w}\right]$.

The following Lemma forms the theoretical basis of any variational inference approach.

Lemma 3.1.4 (Variational Lower Bound). For any unbiased estimator $\hat{w}$ of $p\left(\mathbf{x}\right)$ and for the corresponding variational lower bound $\hat{L} = \mathbb{E}\left[\log \hat{w}\right]$, we have:

$$\log p\left(\mathbf{x}\right) \geq \hat{L} = \mathbb{E}\left[\log \hat{w}\right]$$

Proof . We have $\log p\left(\mathbf{x}\right) = \log \mathbb{E}\left[\hat{f}\right] \geq^{\dagger} \mathbb{E}\left[\log \hat{f}\right] = \hat{L}$. Here, the equality $(\dagger)$ follows from Jensen's Inequality for the concave function $\log\left(\cdot\right)$, which completes the proof. $\square$

Remark 3.1.5 (Variational Lower Bound as Estimator). Note that the variational lower bound $\hat{L} = \mathbb{E}\left[\log \hat{f}\right]$ can also be viewed as an estimator of $\log p\left(\mathbf{x}\right)$. This idea has been demonstrated previously in the works of Nowozin (2018). From Lemma 3.1.4, it is clear that this estimator is biased and always underestimates the true objective $\log p\left(\mathbf{x}\right)$.

3.2. Variational Gap and Estimator Variance

Lemma 3.1.4 and Remark 3.1.5 demonstrate that the variational lower bound always underestimates the true log-likelihood $p\left(\mathbf{x}\right)$. Thus, in order to train the model to optimize $p_\theta\left(\mathbf{x}\right)$, we are required to train the variational lower bound that is always biased. In this context, one of the ways in which we can improve the variational inference is to try to increase the variational lower bound and make it as close to the true objective as possible. Achieving this would enable optimization of a less biased variational lower bound and thus, we can improve variational inference to obtain a better model of the data. Towards this, we consider the difference between the terms $\log p(\mathbf{x})$ and the variational lower bound $\hat{L} = \mathbb{E}[\log \hat{w}]$ corresponding to the unbiased estimator $\hat{w}$ of $p(\mathbf{x})$.

Definition 3.2.1 (Variational Gap). Let $\hat{w}$ be an unbiased estimator of $p(\mathbf{x})$ and let $\hat{L} = \mathbb{E}[\log \hat{w}]$ be the corresponding variational lower bound. The **variational gap** of the estimator is denoted by $\mathcal{V}(\hat{w})$ and is defined as follows:

$$\mathcal{V}(\hat{w}) = \log p(\mathbf{x}) - \hat{L} = \log \mathbb{E}[\hat{w}] - \mathbb{E}[\log \hat{w}]$$

Corollary 3.2.2. For any unbiased estimator $\hat{w}$ of $p(\mathbf{x})$, we have: $\mathcal{V}(\hat{w}) \geq 0$.

Proof . It follows from Lemma 3.1.4: $\log p(\mathbf{x}) \geq \hat{L} \implies \mathcal{V}\left(\hat{f}\right) = \log p(\mathbf{x}) - \hat{L} \geq 0$. □

Remark 3.2.3 (Hölder Defect). This variational gap is also known by different names in different domains; it is referred to as the **Hölder defect** in some earlier literature in the areas of mathematics and economics (Becker (2012)).

From the above discussion, it is clear that we need to squeeze the variational gap to improve the variational inference. Since the variational gap is essentially the gap generated by Jensen's inequality, we aim to exploit its equality condition.

Theorem 3.2.4 (Jensen's Inequality). Let f be a concave function and X be a random variable. Then, $f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$. Further, if f is strictly concave, then equality holds iff $X = \mathbb{E}[X]$ with probability 1, i.e., X is a constant.

Proof . The proof can be found in Theorem 2.6.2 of Cover and Thomas (2006). □

Theorem 3.2.4 combined with the estimator view of variational inference points at a mechanism to reduce the variational gap. From Section 3.1, it is clear that the estimator-view of variational inference has the following benefits: **1.** each unbiased estimator $\hat{w}$ of $p(\mathbf{x})$ leads to its own variational lower bound $\mathbb{E}[\log \hat{w}]$, **2.** this variational lower bound can be computed in terms of powerful parametrizations such as neural nets and thus, it can be optimized to obtain the parameters of the variational inference model. In addition, this view can be combined with the equality condition of Jensen's inequality to connect the variational gap $\mathcal{V}\left(\hat{w}\right)$ corresponding to the estimator $\hat{w}$ with its variance $\text{Variance}\left(\hat{w}\right)$. From Theorem 3.2.4, we have that the variational gap $\mathcal{V}\left(\hat{w}\right)$ is 0 when $\hat{w}$ equals its own expectation value of $p\left(\mathbf{x}\right)$ with probability 1. Thus, if we reduce the *spread* of the distribution of the estimator $\hat{w}$, which can be measured using $\text{Variance}\left(\hat{w}\right)$, then the distribution of $\hat{w}$ will become *sharper*. A sharper distribution of $\hat{w}$ would be more centralized and have *larger* mass at its expectation value. The larger the mass near the expectation value, the more close to equality condition will be the corresponding estimator $\hat{w}$ and thus, the *smaller* will be the variational gap. Thus, the core idea can be summarized as follows: **if** $\text{Variance}\left(\hat{w}\right)\longrightarrow 0$**, then** $\mathcal{V}\left(\hat{w}\right)\longrightarrow 0$ **as well, which improves the variational inference**. This idea has been explored previously in the works of Huang et al. (2019), Huang and Courville (2019), and Klys et al. (2018), and our approach also builds on this idea.

3.3. Bounding the Variational Gap

3.3.1. A Simpler Upper Bound on the Variational Gap

In the previous Section 3.2, we outlined the intuition that decreasing the estimator variance can result in squeezing of the variational gap, which then results in improved variational inference. In this Section, we prove a simpler upper bound on the variational gap in terms of only the variance of the estimator and the number of samples involved under certain amicable conditions, which forms the first contribution of our work.

Theorem 3.3.1 (An Upper Bound on the Variational Gap). Let $\hat{w}$ be an unbiased estimator of $\log p\left(\mathbf{x}\right)$ for a data point $\mathbf{x} \sim X$ sampled from the data random variable X. Let **1**. $\exists\, c \in \mathbb{R}^{+}$ such that $\hat{w} > c$, and let **2**. $\text{Variance}\left(\hat{w}\right)$ be finite. Then,

$$\mathcal{V}\left(\hat{w}\right) < \frac{1}{2 \cdot c^{2}} \cdot \text{Variance}\left(\hat{w}\right)$$

Thus, the variational gap of the estimator $\hat{w}$ can be upper-bounded by an expression defined entirely in terms of the variance of the estimator itself *and nothing else.*

Proof . Consider the second-order Taylor series expansion about the point $w^\star$ for the estimator $\hat{w}$ and for a twice differentiable function f. Based on condition **1**, we know that $\hat{w} \in (c, +\infty)$. Thus, by Taylor's theorem, there exists a real number $w^\dagger > c$ such that:

$$f(\hat{w}) = f(w^\star) + f'(w^\star) \cdot (\hat{w} - w^\star) + f''\left(w^\dagger\right) \cdot \frac{(\hat{w} - w^\star)^2}{2!} \tag{3.3.1}$$

Now, we make the following settings: $f(\cdot) = \log(\cdot)$ and $w^\star = \mathbb{E}[\hat{w}]$. Clearly, these substitutions are legal, as f is twice differentiable over $(c, +\infty)$ and $w^\star \in (c, +\infty)$. Thus,

$$\begin{aligned}
& f(\hat{w}) = f(\mathbb{E}[\hat{w}]) + f'(\mathbb{E}[\hat{w}]) \cdot (\hat{w} - \mathbb{E}[\hat{w}]) + f''\left(w^\dagger\right) \cdot \frac{(\hat{w} - \mathbb{E}[\hat{w}])^2}{2!} \\
\overset{(1)}{\Longrightarrow}\ & \mathbb{E}[f(\hat{w})] = \mathbb{E}[f(\mathbb{E}[\hat{w}])] + \mathbb{E}[f'(\mathbb{E}[\hat{w}]) \cdot (\hat{w} - \mathbb{E}[\hat{w}])] + \mathbb{E}\left[f''\left(w^\dagger\right) \cdot \frac{(\hat{w} - \mathbb{E}[\hat{w}])^2}{2!}\right] \\
\Longrightarrow\ & \mathbb{E}[\log \hat{w}] = \mathbb{E}[\log \mathbb{E}[\hat{w}]] + \mathbb{E}\left[\log' \mathbb{E}[\hat{w}] \cdot (\hat{w} - \mathbb{E}[\hat{w}])\right] + \mathbb{E}\left[\log''\left(w^\dagger\right) \cdot \frac{(\hat{w} - \mathbb{E}[\hat{w}])^2}{2!}\right] \\
\overset{(2)}{\Longrightarrow}\ & \mathbb{E}[\log \hat{w}] = \log p(\mathbf{x}) + \log' p(\mathbf{x}) \cdot \underbrace{\mathbb{E}[\hat{w} - \mathbb{E}[\hat{w}]]}_{=0} + \frac{-1/\left(w^\dagger\right)^2}{2} \cdot \text{Variance}(\hat{w}) \\
\overset{(3)}{\Longrightarrow}\ & \mathcal{V}(\hat{w}) = p(\mathbf{x}) - \mathbb{E}[\log \hat{w}] = \frac{1}{2 \cdot \left(w^\dagger\right)^2} \cdot \text{Variance}(\hat{w}) < \frac{1}{2 \cdot c^2} \cdot \text{Variance}(\hat{w})
\end{aligned} \tag{3.3.2}$$

Here, the implication (1) follows from taking expectation on both the sides of the previous equality, the implication (2) follows from the substitutions of $f(\cdot) = \log(\cdot)$ and its derivatives, and the implication (3) follows from the assumption $w^\dagger > c$. $\square$

Remark 3.3.2 (**Lower Bounded Estimator Assumption (LBE)**). In Theorem 3.3.1, we made two important assumptions on the estimator $\hat{w}$. The later assumption of Variance $(\hat{w})$ being finite is a widely accepted and reasonable assumption which is used in many other works to prove the corresponding theoretical guarantees (Huang et al. (2019), Burda et al. (2015)). However, the former assumption of $\hat{w} > c$ seems to be a fairly strong assumption, and we call it the **lower bounded estimator assumption (LBE)**. In the subsequent discussions, we will consider the details of this assumption and argue that it is justifiable.

Remark 3.3.3 (**Connection to Hölder Defect**). Our motivation to pursue this approach comes from the works of Huang and Courville (2019), where the authors consider first-order Taylor-series expansion about the median of the estimator. Instead, we consider the second-order Taylor-expansion about the mean of the estimator. This second-order expansion allows us to obtain an upper bound in terms of the variance of the estimator, which is what we wanted to achieve. However, this technique of second-order Taylor-expansion is fairly common in mathematical literature. For instance, similar techniques are used to prove optimality tests in calculus and these techniques have also been used independently in the works of Becker (2012) on the Hölder defect. However, our approach uses this technique specifically for upper-bounding the variational gap in terms of the estimator variance, and nothing else, and thus, differs from these previous applications.

3.4. Variationally-Asymptotic Monte-Carlo Estimators

In this Section, we continue with the use of Theorem 3.3.1 to derive the properties of the rate of improvement of variational inference. Recall that Theorem 3.3.1 gives an upper bound on the variational gap $\mathcal{V}(\hat{w})$ corresponding to an unbiased estimator $\hat{w}$ of $p(\mathbf{x})$ in terms of its variance $\text{Variance}(\hat{w})$. This results in a powerful *reductionism*; **we can reduce the problem of improving variational inference to the problem of finding low-variance estimators of the data likelihood** $p(\mathbf{x})$. However, by itself, Theorem 3.3.1 does not provide a mechanism to decrease the estimator variance. Here, we take inspiration from the recent work on importance weighted autoencoders (IWAE) by Burda et al. (2015), which demonstrates a mechanism of provably improving the variational lower bound, thereby decreasing the variational gap.

Theorem 3.4.1 (IWAE (Burda et al. (2015))). Let $\mathcal{X}$ denote the space of input data. Consider the estimator $\hat{w}^n_{\text{IWAE}} = \frac{1}{n}\sum_{i=1}^{n}\frac{p(\mathbf{x},\mathbf{z}^i)}{q(\mathbf{z}^i|\mathbf{x})}$ defined for $\mathbf{x} \in \mathcal{X}$ using samples $\mathbf{z}^i$ $(i \in \{1,\ldots,n\})$ such that $\mathbf{z}^i \overset{\text{iid}}{\sim} q(\cdot \mid \mathbf{x})$. Let $\hat{L}^n_{\text{IWAE}} = \mathbb{E}[\log \hat{w}^n_{\text{IWAE}}]$. Then,

1. $\mathbb{E}[\hat{w}^n_{\text{IWAE}}] = p(\mathbf{x})$, i.e., $\hat{w}^n_{\text{IWAE}}$ is an unbiased estimator of $p(\mathbf{x})$,

2. $\log p(\mathbf{x}) \geq \hat{L}^n_{\text{IWAE}}$, i.e., $\hat{L}^n_{\text{IWAE}}$ is indeed a variational lower bound for $\log p(x)$,

3. $\hat{L}^n_{\text{IWAE}} \geq \hat{L}^m_{\text{IWAE}}\ \forall\ n > m$, i.e., the lower bound increases monotonically with n, and

4. $\lim_{n\longrightarrow\infty} \hat{L}^n_{\text{IWAE}} = \log p(\mathbf{x})$, i.e., the lower bound is asymptotically unbiased.

Proof . Please refer to the Appendix A of Burda et al. (2015) for the proof. □

From Theorem 3.4.1 and the definition of the IWAE estimator $\hat{w}^n_{\text{IWAE}}$, we observe two important components to design low-variance estimators.

- Instead of a single sample, use multiple samples from the proposal distribution in order to define a set of unbiased estimators of the data likelihood.
- Consider an average of these estimators to create a single desired estimator.

With these observations, we consider the following definitions.

Definition 3.4.2 (Monte-Carlo Estimator (MCE)). Let $\{\hat{w}_i\}_{i=1}^n$ be a set of n unbiased estimators of the data likelihood $p(\mathbf{x})$. Then, the corresponding Monte-Carlo estimator of the data likelihood $p(\mathbf{x})$ is denote by $\hat{w}^n$, and is defined as $\hat{w}^n = \frac{1}{n}\sum_{i=1}^n \hat{w}_i$.

Lemma 3.4.3. If $\hat{w}_i$ are unbiased estimators of $p(\mathbf{x})$, then the corresponding Monte-Carlo estimator is also unbiased, i.e., $\mathbb{E}[\hat{w}^n] = p(\mathbf{x})$.

PROOF . We have $\mathbb{E}[\hat{w}^n] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \hat{w}_i\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}[\hat{w}_i] = \frac{1}{n}\sum_{i=1}^n p(\mathbf{x}) = p(\mathbf{x})$. □

However, the most important aspect of using the Monte-Carlo estimators is their variance reduction properties. As we will demonstrate in Chapter 4, the n individual estimators can be easily designed so that the variance of the Monte-Carlo estimator reduces at the rate of $\frac{1}{n}$. Towards formalizing this idea, we first consider some definitions.

Definition 3.4.4 (Big-O Notation). Consider two functions $f, g : \mathbb{N} \longrightarrow \mathbb{R}$. We define f to be of **order** g, and denote it by $f(n) = \mathcal{O}(g(n))$ iff $\exists\ n_0 \in \mathbb{N}$ and $\exists\ k \in \mathbb{R}^+$ such that $\forall\ n \in \mathbb{N}$ with $n > n_0$, we have $f(n) \leq k \cdot g(n)$, i.e., eventually, the growth of the function f is upper bounded by a scalar k times that of the function g.

Definition 3.4.5 (Variationally-Asymptotic Monte-Carlo Estimator). Let $\{\hat{w}_i\}_{i=1}^n$ be a set of unbiased estimators of $p(\mathbf{x})$ and let $\hat{w}^n = \frac{1}{n}\sum_{i=1}^n \hat{w}_i$ be the corresponding Monte-Carlo estimator. Due to lack of a better terminology, we call the estimator $\hat{w}^n$ to be variationally-asymptotic Monte-Carlo estimator iff it satisfies

1. $\exists\ c > 0$ such that $\hat{w}_i > c\ \forall\ i \in \{1, \dots, n\}$, i.e., the individual estimators $\hat{w}_i$ are lower bounded estimators, and

2. $\exists\ b > 0$ such that $\text{Variance}(\hat{w}^n) \leq \frac{b}{n}$, i.e., the variance of the Monte-Carlo estimator decreases to 0 at a rate of $\frac{1}{n}$ or faster and thus, $\text{Variance}(\hat{w}^n) = \mathcal{O}\left(\frac{1}{n}\right)$.

With these definitions, we prove the master theorem that describes the rate of convergence of the variational gap to 0 for a general family of estimators. We prove that satisfying the two conditions of the Definition 3.4.5 results in decrease in the variational gap at a rate of $\mathcal{O}\left(\frac{1}{n}\right)$.

Theorem 3.4.6 (**Master Theorem on the Rate of Improvement of VI**).
Let $\{\hat{w}_i\}_{i=1}^{n}$ be a set of unbiased estimators of $p(\mathbf{x})$ such that the Monte-Carlo estimator $\hat{w}^n = \frac{1}{n}\sum_{i=1}^{n} \hat{w}_i$ is in fact a variationally-asymptotic Monte-Carlo estimator, i.e., **1.** $\exists\, c > 0$ such that $\hat{w}^i > c \,\forall\, i \;\; (1 \le i \le n)$, and **2.** $\exists\, b > 0$ such that $\text{Variance}\left(\hat{w}^n\right) \le \frac{b}{n}$. Then, $\mathcal{V}\left(\hat{w}^n\right) = \mathcal{O}\left(\frac{1}{n}\right)$, i.e., the variational gap for the estimator $\hat{w}^n$ goes to 0 at the rate of $\frac{1}{n}$ or faster. Thus, for the variational lower bound $\hat{L}^n = \mathbb{E}\left[\log \hat{w}^n\right]$, we have that $\hat{L}^n \longrightarrow \log p\left(\mathbf{x}\right)$ at the rate of $\frac{1}{n}$ or faster.

Proof . First, notice that the Monte-Carlo estimator $\hat{w}^n$ is also a lower bounded estimator.

$$\hat{w}^n = \frac{1}{n}\sum\nolimits_{i=1}^{n} \hat{w}_i >^{*} \frac{1}{n}\sum\nolimits_{i=1}^{n} c = c. \tag{3.4.1}$$

Here, the inequality $(*)$ follows from the given condition **1**. Thus, $\exists\; c > 0$ such that $\hat{w}^n > c$, which gives that $\hat{w}^n$ is a lower bounded estimator. Thus, we can use the upper bound on the variational gap from Theorem 3.3.1 to prove the desired result as follows.

$$\mathcal{V}\left(\hat{w}^n\right) \le \frac{1}{2\cdot c^2}\cdot \text{Variance}\left(\hat{w}^n\right) \le^{\dagger} \frac{1}{2\cdot c^2}\cdot\frac{b}{n} = \frac{b}{2\cdot c^2\cdot n} \tag{3.4.2}$$

Here, the inequality $(\dagger)$ follows from the given condition **2**. Thus, $\text{Variance}\left(\hat{w}^n\right) = \mathcal{O}\left(\frac{1}{n}\right)$.□

Remark 3.4.7 (**Emphasis on Variance Reduction in Estimators**). As discussed earlier, the lower bounded estimator assumption, which is the condition **1** from Definition 3.4.5, is assumed in Theorem 3.4.6. Towards the end of this Chapter, we will discuss this condition and justify it by giving arguments in the support of its feasibility. With this, we will be assuming this condition in all our subsequent discussions. Thus, Theorem 3.4.6 completes the reduction of the problem of the rate of improvement in the variational inference to the rate of decrease of the variance of the Monte-Carlo estimator under consideration.

3.4.1. Variance Reduction Techniques

In this Subsection, we focus on the techniques for reducing the variance of the Monte-Carlo estimators. These techniques revolve around the following idea: for a set of estimators $\{\hat{w}_i\}_{i=1}^n$ with an upper bound on their variance, if either all of them or sufficiently many of them are pairwise uncorrelated, then the variance of the Monte-Carlo estimator $\hat{w}^n = \frac{1}{n}\sum_{i=1}^n \hat{w}_i$ vanishes to 0 at the rate of $\mathcal{O}\left(\frac{1}{n}\right)$. One of the mechanisms, then, of making the individual estimators $\hat{w}_i$ uncorrelated is to use i.i.d. samples to construct these estimators. This is precisely the idea underlying the IWAE estimator in Burda et al. (2015).

Theorem 3.4.8. Consider the estimators $\hat{w}_{\text{IWAE},i} = \frac{p(\mathbf{x},\mathbf{z}^i)}{q(\mathbf{z}^i|\mathbf{x})}$ defined using samples $\mathbf{z}^i$ $(i \in \{1,\ldots,n\})$ such that $\mathbf{z}^i \overset{\text{iid}}{\sim} q(\cdot \mid \mathbf{x})$. Then, we have that $\forall\, i,j \in \{1,\ldots,n\}$ with $i \neq j$, the estimators $\hat{w}_{\text{IWAE},i}$ and $\hat{w}_{\text{IWAE},j}$ are independent and thus, uncorrelated.

Proof . Note that for a given data point $\mathbf{x}$, the estimator $\hat{w}_{\text{IWAE},i} = \frac{p(\mathbf{x},\mathbf{z}^i)}{q(\mathbf{z}^i|\mathbf{x})}$ is a function of $\mathbf{z}^i$. We know that $\forall\, i,j \in \{1,\ldots,n\}$ with $i \neq j$, the samples $\mathbf{z}^i$ and $\mathbf{z}^j$ are independent and we also know that functions of independent random variables are also independent. Thus, $\hat{w}_{\text{IWAE},i}$ and $\hat{w}_{\text{IWAE},j}$ are independent, which makes them uncorrelated as well. □

Now, it is easy to see that if the individual estimators are all pairwise uncorrelated, then the Monte-Carlo estimator has a variance that decreases to 0 at the rate of $\mathcal{O}\left(\frac{1}{n}\right)$.

Theorem 3.4.9. Consider the set of estimators $\{\hat{w}_i\}_{i=1}^n$ such that **1.** $\forall\, i,j \in \{1,\ldots,n\}$ with $i \neq j$, the estimators $\hat{w}_{\text{IWAE},i}$ and $\hat{w}_{\text{IWAE},j}$ are all pairwise uncorrelated, and **2.** $\exists\, B > 0$ such that $\text{Variance}\left(\hat{w}_i\right) < B\ \forall\, i \in \{1,\ldots,n\}$. Then, for the corresponding Monte-Carlo estimator $\hat{w}^n = \frac{1}{n}\sum_{i=1}^n \hat{w}_i$, we have that $\text{Variance}\left(\hat{w}^n\right) = \mathcal{O}\left(\frac{1}{n}\right)$.

Proof . Consider the following manipulations.

$$\begin{aligned}
\text{Variance}\left(\hat{w}^n\right) &= \text{Variance}\left(\frac{1}{n}\sum\nolimits_{i=1}^n \hat{w}_i\right) =^{*} \left(\frac{1}{n}\right)^2 \cdot \text{Variance}\left(\sum\nolimits_{i=1}^n \hat{w}_i\right) \\
&=^{\dagger} \frac{1}{n^2}\cdot\left(\sum\nolimits_{i=1}^n \text{Variance}\left(\hat{w}_i\right) + 2\cdot\sum\nolimits_{1\le i<j\le n}\text{Covariance}\left(\hat{w}_i,\hat{w}_j\right)\right) \\
&<^{\ddagger} \frac{1}{n^2}\cdot\left(\sum\nolimits_{i=1}^n B + 2\cdot\sum\nolimits_{1\le i<j\le n} 0\right) = \frac{n\cdot B}{n^2} = \frac{B}{n}
\end{aligned} \tag{3.4.3}$$

Here, the first two equalities follow from the standard variance properties and the last inequality follows from the given conditions **1** and **2**. □

Note that such IWAE-like approaches usually require a significantly large number of samples to improve the variational inference by a considerable amount. Thus, although there are theoretical guarantees of improvement, there is a need to improve upon the i.i.d. samples as done in IWAE by Burda et al. (2015). In this direction, another idea that is explored in the works of Huang et al. (2019) on hierarchical importance-weighted autoencoders is to learn estimators that can have a negative correlation. In particular, this approach utilizes the estimators $\hat{w}_{\text{HIWAE},i} = \pi_i\left(\mathbf{z}^0, \mathbf{z}^i\right) \cdot \frac{p\left(\mathbf{x}, \mathbf{z}^i\right) \cdot r\left(\mathbf{z}^0 \mid \mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^0\right) \cdot q\left(\mathbf{z}^0\right)}$, where $\mathbf{z}^i$ are sampled conditionally independently given $\mathbf{z}^0$. Although Huang et al. (2019) empirically demonstrate that these estimators end up learning negative correlation, there is no theoretical guarantee that this will always happen. Thus, there is no theoretical guarantee that HIWAE estimator $\hat{w}^n_{\text{HIWAE}} = \frac{1}{n}\sum_{i=1}^{n} \hat{w}_{\text{HIWAE},i}$ will always have decreasing variance. In this context, our variance reduction technique, described in Chapter 4, achieves the best of both the worlds.

Our Variance Reduction Approach: We sample the latent variables $\mathbf{z}^{1:n}$ sequentially by explicitly modeling their dependency using a conditional proposal $q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}, \mathbf{x}\right)$ or by using ideas from antithetic sampling (Wilson (1983)). We then create the estimators $\{\hat{w}_i\}_{i=1}^n$ such that either all or sufficiently many of them become pairwise uncorrelated and we obtain a Monte-Carlo estimator with $\mathcal{O}\left(\frac{1}{n}\right)$ variance.

We claim that our approach is better than the i.i.d. sampling in IWAE-like approaches. This is because our approach models the dependencies between latent variables by modeling their joint distribution. This enables modeling of posterior distributions that are more expressive as compared to that of the i.i.d. samples. Since each new sample is obtained conditioned on the previous ones, the variational lower bound can be made to increase with lesser number of samples. In fact, our approach of conditional sampling is a generalization iof the i.i.d. sampling; the proposal distribution $q\left(\mathbf{z}^i \mid \mathbf{x}\right)$ of i.i.d. samples is a special case of the conditional proposal $q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}, \mathbf{x}\right)$, where there is no dependence in $\mathbf{z}^i$ and $\mathbf{z}^{1:i-1}$. Further, in Section 4.7, we will demonstrate that our approaches indeed perform better than the baseline IWAE approach, which supports our claim.

3.5. Related Work

In this Section, we consider the previously demonstrated theoretical guarantees of improvement in variational inference with the number of samples along with their assumptions. We first consider the IWAE by Burda et al. (2015), which is the initial idea in this direction.

- **Results of IWAE by Burda et al. (2015)**

  Let $\hat{w}_{\text{IWAE},i} = \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i|\mathbf{x}\right)}$, $\hat{w}^n_{\text{IWAE}} = \frac{1}{n}\sum_{i=1}^{n}\hat{w}_{\text{IWAE},i}$, $\hat{L}^n_{\text{IWAE}} = \mathbb{E}\left[\log \hat{w}^n_{\text{IWAE}}\right]$. Then,

  (1) $\forall\; n, m \in \mathbb{N}$ with $n < m$, $\hat{L}^n_{\text{IWAE}} \leq \hat{L}^m_{\text{IWAE}}$.

  (2) $\lim_{n\longrightarrow\infty} \hat{L}^n_{\text{IWAE}} = p\left(\mathbf{x}\right)$.

- **Assumptions**

  (1) $\hat{w}_{\text{IWAE},i}$ is bounded.

This approach proves that with an increasing number of samples, the variational lower bound approaches the true marginal $\log p\left(\mathbf{x}\right)$, thereby decreasing the variational gap to 0. However, it does not provide an upper bound on the variational gap while doing so. An extension of the IWAE approach is provided by Cremer et al. (2017), where they interpret the IWAE as performing just the usual variational inference but with a more complex posterior induced by the use of multiple samples. However, their theoretical guarantees are analogous to the one in IWAE, demonstrating only that their variational lower bound converges to $\log p\left(\mathbf{x}\right)$. The next major step was given by Nowozin (2018), in which $\hat{L}^n_{\text{IWAE}}$ is viewed as a biased estimator of $\log p\left(\mathbf{x}\right)$ and thus, its bias and variance is estimated.

- **Results of Nowozin (2018)**

  Let $\hat{w}_i \overset{\text{iid}}{\sim} P$ with $\mathbb{E}\left[\hat{w}_i\right] = p\left(\mathbf{x}\right)$ $(1 \leq i \leq n)$, where P is the distribution of each of the n individual i.i.d. estimators. Let $\hat{w}^n = \frac{1}{n}\sum_{i=1}^{n}\hat{w}_i$, and $\hat{L}^n = \mathbb{E}\left[\log \hat{w}^n\right]$. Let $\mu_j = \mathbb{E}_P\left[\left(w - \mathbb{E}_P\left[w\right]\right)^j\right]$ $\forall\; j \in \mathbb{N}$ and $\mu = \mathbb{E}_P\left[w\right]$. Then,

  (1) $-\mathcal{V}\left(\hat{w}^n\right) = \hat{L}^n - \log p\left(\mathbf{x}\right) = -\frac{1}{n}\cdot\frac{\mu_2}{2\cdot\mu^2} + \frac{1}{n^2}\cdot\left(\frac{\mu_3}{3\cdot\mu^3} - \frac{3\cdot\mu_2^2}{4\cdot\mu^4}\right)$
  $-\frac{1}{n^3}\cdot\left(\frac{\mu_4}{4\cdot\mu^4} - \frac{3\cdot\mu_2^2}{4\cdot\mu^4} + \frac{10\cdot\mu_2\cdot\mu_3}{5\cdot\mu_5^5}\right) + o\left(\frac{1}{n^3}\right)$.

- **Assumptions**

  (1) $\hat{w}_i$ must be sampled independently from P.

  (2) the distribution P of the i.i.d. estimators has finite moments of all orders.

This work hints that the variational gap is dominated by the $\frac{1}{n}$ term. Further, the proportionality constant $\frac{\mu_2}{2\cdot\mu^2}$ is related to the coefficient of variation $v = \sqrt{\frac{\mu_2}{\mu^2}}$ and thus, the variational gap is demonstrated to be related to the spread of the distribution of $\hat{w}$. Both these insights are proved in our approach and further, the mathematical form of our upper bound of the variation gap (from Theorem 3.3.1) has the same form as that proved by Nowozin (2018). The next major step, by Maddison et al. (2017), demonstrates a bound on the variational gap of any Monte-Carlo estimator $\hat{w}^n$ using a variant of its variance Variance $(\hat{w}^n)$.

- **Results of Maddison et al. (2017)**

  Consider estimators $\hat{w}_i$ with $\mathbb{E}[\hat{w}_i] = p(\mathbf{x})$ $(1 \leq i \leq n)$. Let $\hat{w}^n = \frac{1}{n}\sum_{i=1}^{n}\hat{w}_i$, and $\hat{L}^n = \mathbb{E}[\log \hat{w}^n]$. Let $g(n) = \mathbb{E}\left[(\hat{w}^n - p(\mathbf{x}))^6\right]$. Then,

  (1) $\hat{L}^n \longrightarrow \log p(\mathbf{x})$ as $n \longrightarrow \infty$.

  (2) $\mathcal{V}(\hat{w}^n) \leq \frac{1}{2} \cdot \text{Variance}\left(\frac{\hat{w}^n}{p(\mathbf{x})}\right) + \mathcal{O}\left(\sqrt{g(n)}\right)$.

- **Assumptions**

  (1) $\hat{w}^n$ is uniformly and strongly consistent.

  (2) $\limsup_{n\longrightarrow\infty} \mathbb{E}\left[\frac{1}{\hat{w}^n}\right]$ is bounded above.

This bound connects $\mathcal{V}(\hat{w}^n)$ with Variance $\left(\frac{\hat{w}^n}{p(\mathbf{x})}\right)$, a variant of Variance $(\hat{w}^n)$. However, it also sheds light on a crucial idea; the expectation of the inverse estimator is required to be bounded above. In our bound from Theorem 3.3.1, the strong assumption lower bounded estimator is connected to this assumption. If the lower-bounded estimator condition holds, then there exists $c > 0$ such that $\hat{w}^n < c$. Using this, we get that $\mathbb{E}\left[\frac{1}{\hat{w}^n}\right] < \mathbb{E}\left[\frac{1}{c}\right] = \frac{1}{c}$. Thus, if lower bounded estimator assumption holds true, then assumption (2) of Maddison et al. (2017) holds true as well. This relation also suggests that the lower bounded estimator assumption is stronger than assumption (2) of Maddison et al. (2017) as the former implies the latter. However, this upper bound is complex and can not be applied directly to comment on the rate of decrease of the variational gap with the number of samples. Thus, our approach provides a simpler bound on the variational gap at the cost of a related but slightly stronger assumption. Another approach that attempts to derive a simpler upper bound on the variational gap is the work by Klys et al. (2018).

- **Results of Klys et al. (2018)**

  Let $\hat{w}$ be an unbiased estimator of $p(\mathbf{x})$. Then,

  (1) $\exists\, C > 0$ independent of $\hat{w}$ such that for any sufficiently small $\epsilon > 0$, $\mathcal{V}(\hat{w}) < C \cdot \left(\epsilon + \frac{\text{Variance}(\hat{w})}{\epsilon}\right)$. Thus, if Variance $(\hat{w}) \longrightarrow 0$, then $\mathcal{V}(\hat{w}) \longrightarrow 0$.

- **Assumptions**

  (1) The assumption $\hat{w}^n$ is uniformly integrable is needed to complete the proof, which was suggested by Christian Naesseth but missing from the original works of Klys et al. (2018).

The utility of this approach is that the variational gap and the estimator variance are related through a simple bound. Our work removes the need for the extra constant C from the upper bound and gives a simpler bound using only the estimator variance. Another approach is demonstrated in the works of Huang et al. (2019) and Huang and Courville (2019).

- **Results of Huang and Courville (2019)**

  For an unbiased estimator $\hat{w}$ of $p(\mathbf{x})$, let $\mu_X = p(\mathbf{x}), \nu_X$ be its mean and median. Let $\mu_Y = \mathbb{E}[\log \hat{w}], \nu_Y$ be the mean and median of the distribution of $\log \hat{w}$. Let $\exists\, C_X, C_Y > 0$ such that $|\mu_X - \nu_X| \leq C_X,\ |\mu_Y - \nu_Y| \leq C_Y$. Then,

  (1) $\mathcal{V}(\hat{w}^n) < \frac{C_X}{\mu_X - C_X} + C_Y$.

  (2) Further, let σ_X, σ_Y be the standard deviations of $\hat{w}$ and $\log \hat{w}$. Then, $\mathcal{V}(\hat{w}^n) < \frac{\sigma_X}{p(\mathbf{x}) - \sigma_X} + \sigma_Y$.

- **Assumptions**

  (1) The distribution of $\hat{w}$ has a single median.

  (2) $p(\mathbf{x}) = \mu_X > C_X$.

  (3) Constant $C_X > 0$ exists such that it satisfies assumption (2) as well as $|\mu_X - \nu_X| \leq C_X$ at the same time.

- **Results of HIWAE by Huang et al. (2019)**

  Let $\hat{w}^n$ be an unbiased Monte-Carlo estimator of $p(\mathbf{x})$. Then,

  (1) $\lim_{n \longrightarrow \infty} \text{Variance}(\hat{w}^n) = 0 \implies \log \hat{w}^n \longrightarrow \log p(\mathbf{x})$ in probability.

  (2) Further, if $\log \hat{w}^n \to \log p(\mathbf{x})$ in $\mathcal{L}^2$ norm, then $\lim_{n \to \infty} \text{Variance}(\hat{w}^n) = 0$.

- **Assumptions**

  (1) $\mathbb{E}[|\hat{w}^n|] < \infty$, $\{\hat{w}^n\}$ is uniformly integrable.

  (2) $\mathbb{E}\left[|\log \hat{w}^n|^2\right] < \infty$, $\{\log \hat{w}^n\}$ is bounded in $\mathcal{L}^1$ norm.

  (3) $\{\log \hat{w}^n\}$ is uniformly integrable.

These results relax the strict conditions from the work of Burda et al. (2015) and provide a simple upper bound to the variational gap in terms of the variance of the estimator. However, the problem with this approach is that it requires too many conditions on the distribution of the estimator. In comparison, we argue that we provide a fairly elegant upper bound on the variational gap using only two simple assumptions. This concludes the comparison of our results with the previously published ones and we turn our attention to the analysis of the crucial lower-bounded estimator assumption.

3.6. On the Lower Bounded Estimator Assumption

In this section, we discuss in detail the LBE assumption and provide justifications for it. Although this assumption is fairly strong, we aim to argue that this assumption is reasonable and justifiable. In this context, we provide the following arguments:

(1) In maximizing the variational lower bound/ELBO, we are effectively attempting to maximize the values of the estimators. Further, the estimator value indeed increases as the training progresses, even with sampled latent variables and even for validation and test dataset, and this can be easily verified empirically.

(2) LBE assumption ensures that all data points get a probability density of at least c, which is a desirable property of a parametrized model of the data.

3.6.1. Estimating the Variational Lower Bound

Note that for any unbiased estimator $\hat{w}$ of $p(\mathbf{x})$, we have the corresponding trainable variational lower bound $\hat{L} = \mathbb{E}[\log \hat{w}]$. This expectation is taken with respect to the proposal distribution $q(\mathbf{z} \mid \mathbf{x})$, where $\mathbf{z}$ represents all the latent variables involved in computing the estimator $\hat{w}$. However, note that computing the expectation value exactly is difficult and thus, we effectively compute its Monte-Carlo expectation value. Thus, we get the estimation $\widetilde{L}$ of the true variational lower bound $\hat{L}$ as $\hat{L} \approx \widetilde{L} = \frac{1}{m}\sum_{\mathbf{z}^i \sim q(\cdot|\mathbf{x}),\ 1 \leq i \leq m} \log \hat{w}^i$, where $\hat{w}^i$ is the estimator value computed using the sample $\mathbf{z}^i$. Thus, during training, we are effectively maximizing $\log \hat{w}$, the log of the estimator value, which in term implies that we are effectively increasing the estimator value itself.

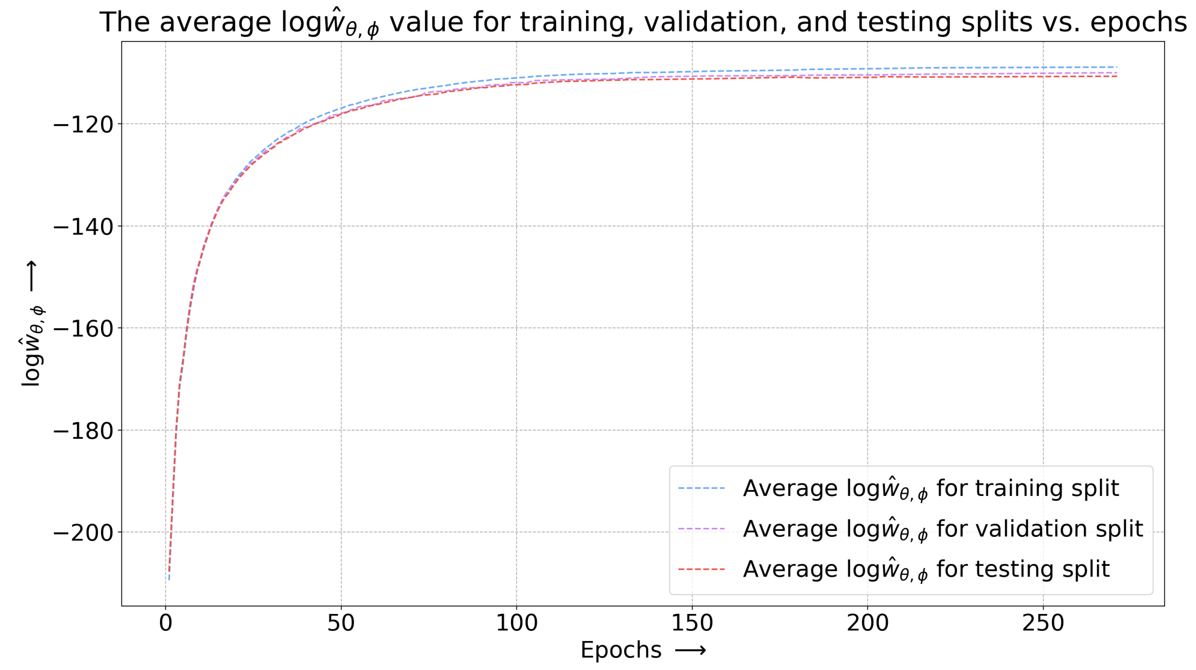


Fig. 3.1. The graph of the values of $\log \hat{w}_{CIWAE}$ for the training, validation and testing splits of the OMNIGLOT dataset (Lake et al. (2015)). Our CIWAE approach is described in Section 4.2. This graph emphasizes that our training process does not overfit and thus, the estimator values increase on all the splits of the dataset. This graph is an example; similar training curves are obtained for other datasets with all approaches.

Now, the estimator $\hat{w}$ is usually parametrized as $\hat{w}_{\theta,\phi}$, where ϕ represents the encoder parameters and θ represents the decoder and prior parameters. Thus, the estimator values $\hat{w}_{\theta,\phi}$ for the training dataset increase during the training process. However, even though we

train our model on the training dataset, we require the estimator values for the validation and testing datasets to increase as well. This is because, in principle, it is possible that during the training process, the estimator values increase only for the training dataset but decreases for the validation and testing datasets. However, this happens if and only if we overfit the training dataset. Since we stop training our model parameters ϕ, θ as soon as they start to overfit, our model generalizes better and thus, estimator values increase even on the validation and test datasets during the training process.
Thus, with Figure 3.1, it is clear that after the training is complete, the value of the estimator $\hat{w}_{\theta,\phi}$ is much higher than their initial value. Thus, if the initial values of estimators are positive, which is usually true based on the design, it is not difficult to imaging that the value of the trained estimators $\hat{w}_{\theta,\phi}$ will be lower-bounded by a positive constant.

3.6.2. Consequences of LBE

Next, we can see that if the estimator $\hat{w}$ is lower-bounded, then so is the data marginal. If $\exists\ c > 0$ such that $\hat{w} > c$, then $p(\mathbf{x}) = \mathbb{E}[\hat{w}] > \mathbb{E}[c] = c$, which gives that $p(\mathbf{x}) > c$. Now, we argue that this is indeed a desirable property. From Definition 3.1.1, we know that we want to learn parametrized models $p_\theta(\cdot)$ of data which assign high probability density to data points $\mathbf{x}$. This is what is ensured in the above inequality; each data point $\mathbf{x}$ gets a probability density of at least c. On the contrary, those parametrized models that do not satisfy this property would not be desirable as there will be data points with arbitrarily low densities associated with them. However, at the same time, there are distributions $p(\mathbf{x})$ that we would like to use for modeling data but that do not satisfy this property. For instance, the Gaussian distribution $p(\mathbf{x}) = \mathcal{N}(\mu, \sigma^2)$ with mean μ and standard deviation σ is a simple distribution for modeling the data that does not satisfy the above property $p(\mathbf{x}) > c$ for some $c > 0$. This example demonstrates a limitation of our approach, which is the result of the strong lower bounded estimator assumption. This points towards a possible future research direction of studying the extent to which a data distribution $p_{\text{data}}(\mathbf{x})$ can be modeled with parametrized distribution $p_\theta(\mathbf{x})$; e.g., modeling a Gaussian distribution *sufficiently well* with an appropriately truncated Gaussian distribution.

3.7. Conclusions and the Significance of the Results

In this Section, we conclude our theoretical results and emphasize their significance.

- Firstly, in Section 3.3, we demonstrate a simple and elegant bound on the variational gap in terms of only the estimator variance. As discussed in Section 3.6, the connection in the variational gap and certain statistical properties of the estimator has previously been observed and exploited for improving variational inference. In the same spirit, our result from Theorem 3.3.1 establishes a direct connection between the variational gap and the estimator variance.
- One of the strengths of our approach is that it reduces the problem of improving variational inference to an easier problem of designing low-variance estimators of the data likelihood $p(\mathbf{x})$. Many previous approaches use multi-sample estimators to improve variational inference. Our result provides a perspective on these approaches; we demonstrate that using multiple samples can result in low-variance estimators, which then lead to a low variational gap and thus, better variational inference.
- Another strength of our approach is that it allows us to address the rate at which the variational inference improves with the number of samples. Since Theorem 3.3.1 directly connects the improvement in variational inference to the estimator variance, we can study the properties of the variances of multi-sample estimators and understand the rate at which using more samples would improve the variational inference.
- Finally, our approach can be combined with well-established variance reduction techniques to design low-variance estimators for improving variational inference. In fact, in Chapter 4, we will use the ideas developed in this section to construct two approaches to improving variational inference and demonstrate their efficacy.

This completes our discussion of the theoretical properties of multi-sample variational inference and their variational gaps. In the next Chapter, we deal with the question of sample efficiency in multi-sample variational inference.

Chapter 4

Variational Inference with Conditional Sampling

4.1. Conditional Sampling in Variational Inference

In this Chapter, we consider the problem of the sample efficiency in multi-sample variational inference. In the previous Chapter, we demonstrated a simple upper bound on the variational gap in terms of the estimator variance and then demonstrated that with multi-sample estimators, the variational gap can be made to vanish at the rate of $\mathcal{O}\left(\frac{1}{n}\right)$. These results provided us with a powerful reduction; we reduced the problem of improving variational inference to the problem of designing low-variance multi-sample estimators. Thus, we next consider the question of sample efficiency in low-variance estimators. The IWAE approach by Burda et al. (2015) utilizes n i.i.d. samples $\mathbf{z}^i$ drawn from the proposal distribution $q\left(\cdot \mid \mathbf{x}\right)$ to create n i.i.d. estimators $\hat{w}_{\text{IWAE},i} = \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i|\mathbf{x}\right)}$. The corresponding IWAE estimator $\hat{w}^n_{\text{IWAE}} = \frac{1}{n}\sum_{i=1}^{n}\hat{w}_{\text{IWAE},i}$ is the desired low-variance Monte-Carlo estimator for $p\left(\mathbf{x}\right)$. However, empirically, the number of samples required for achieving significant improvements is significantly large; for instance, Burda et al. (2015) demonstrate experiments with upto $n = 50$ samples.

One of the natural extension of this idea is to use conditional sampling of latent variables and select each next sample conditioned on the previous ones: $\mathbf{z}^i \sim q\left(\cdot \mid \mathbf{z}^{1:i-1}\right)$. However, with this aforementioned conditional sampling, the usual IWAE estimators $\hat{w}_{\text{IWAE},i}$ are rendered correlated, thereby leading to no guarantees about the $\mathcal{O}\left(\frac{1}{n}\right)$ nature of the variance of their Monte-Carlo estimator. Towards this, we introduce a different set of estimators which, despite the conditional sampling of latent variables, results in pairwise uncorrelated estimators. This approach, which we call **Conditional-IWAE (CIWAE)**, uses the corrected estimators $\hat{w}_{\text{CIWAE},i} = \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i|\mathbf{z}^{1:i-1},\mathbf{x}\right)}$. We prove that these estimators are always pairwise uncorrelated, despite the conditional sampling of latent variables. However, we can see that it is not necessary to make *all* the estimators uncorrelated. We know that for a Monte-Carlo estimator $\hat{w}^n$, its variance is given by: $\text{Variance}\left(\hat{w}^n\right) = \frac{1}{n^2} \cdot \left(\sum_{i=1}^{n} \text{Variance}\left(\hat{w}_i\right) + 2 \cdot \sum_{1\leq i<j\leq n} \text{Covariance}\left(\hat{w}_i, \hat{w}_j\right)\right)$. Thus, the idea here is that we can sample the latent variables so that *significantly many* of the n IWAE estimators $\hat{w}_{\text{IWAE},i} = \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i|\mathbf{x}\right)}$ become uncorrelated to give a Monte-Carlo estimator with $\mathcal{O}\left(\frac{1}{n}\right)$ variance guarantee. There can be many mechanisms of sampling variables to achieve this effect but we utilize the idea of antithetic sampling (Wilson (1983)). In order to generate n samples, we first generate $\left\lfloor\frac{n}{2}\right\rfloor$ samples from the proposal distribution in an i.i.d. manner and use their reflections in the mean of the distribution. These negatively correlated samples should only induce a limited amount of negative correlation in the corresponding estimators. In fact, we prove that only $\mathcal{O}\left(n\right)$ of the $\mathcal{O}\left(n^2\right)$ covariance terms are non-zero. This helps us achieve a Monte-Carlo estimator that retains the $\mathcal{O}\left(\frac{1}{n}\right)$ variance properties. Due to the antithetic nature of sampling involved, we term this approach **Antithetic-IWAE**. This approach has been considered before in the work of Klys et al. (2018). However, we are the first to give the estimator-variance based perspective on this approach and demonstrate its theoretical and empirical utility. In conclusion, our take on sample efficiency can be summarized as follows.

Our Sample Efficiency Approach: Sample each new latent variable conditioned on all or some of the previous ones, and then design appropriate estimators so that the corresponding Monte-Carlo estimator is low-variance.

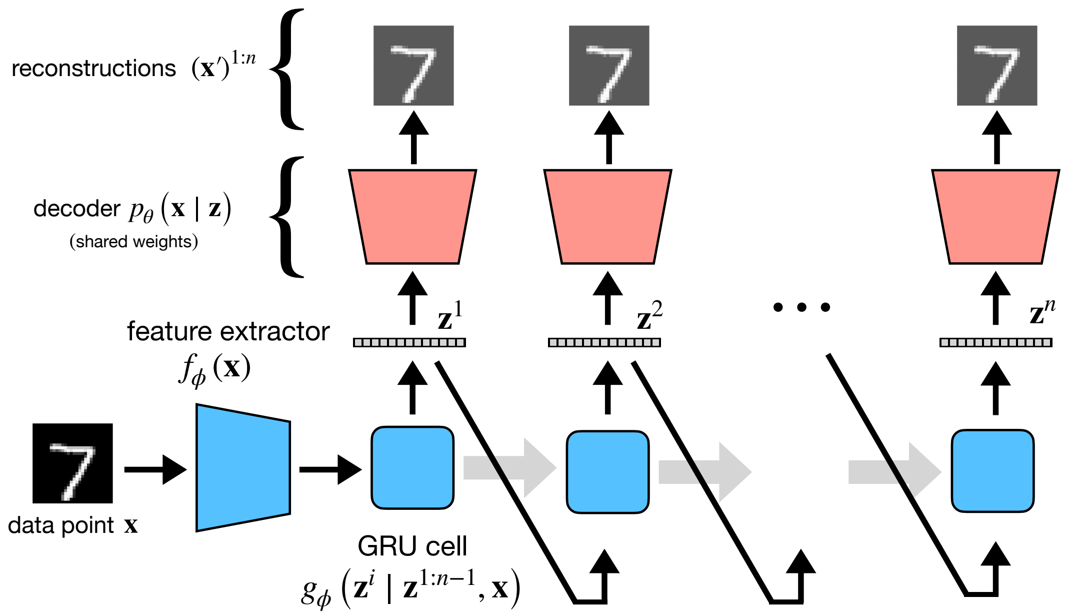
4.2. Conditional-IWAE

As seen previously, the CIWAE approach involves taking n conditional samples $\mathbf{z}^{1:n}$ such that each next sample $\mathbf{z}^i$ is sampled conditioned on all previous ones, i.e., $\mathbf{z}^i \sim q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}, \mathbf{x}\right)$. This approach and the involved sampling scheme is detailed in Algorithm 1.

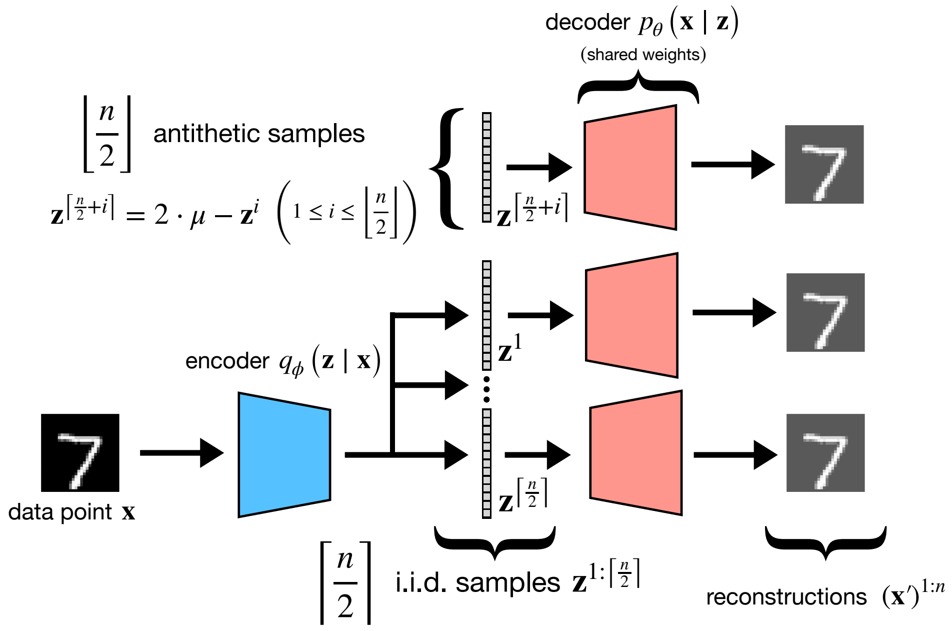
Algorithm 1: Conditional sampling algorithm and the CIWAE approach.

```
function: CONDITIONAL-SAMPLING-ALGORITHM
inputs  : $\mathbf{x} \sim p(\cdot)$: A data point, where $p(\cdot)$ represents the data distribution.
          $q_\phi(\mathbf{z} \mid \mathbf{z}^{i-1}, \mathbf{x})$: The parametrized encoder (neural network).
          $n$: The number of conditional samples to be used for inference.
outputs : $\mathbf{z}^{1:n}$: $n$ conditionally sampled latent variables.
for $i = 1, \ldots, n$ do
    Sample $\mathbf{z}^i \sim q_\phi(\cdot \mid \mathbf{z}^{1:i-1}, \mathbf{x})$;
end
return $\mathbf{z}^{1:n}$;

function: CIWAE
inputs  : $\mathbf{x} \sim p(\cdot)$: A data point, where $p(\cdot)$ represents the data distribution.
          $q_\phi(\mathbf{z} \mid \mathbf{z}^{i-1}, \mathbf{x})$: The parametrized encoder (neural network).
          $p_\theta(\mathbf{x} \mid \mathbf{z})$: The parametrized decoder (neural network).
          $p_\theta(\mathbf{z})$: The prior on all latent variables.
          $n$: The number of conditional samples to be used for inference.
outputs : $\hat{w}^n_{\text{CIWAE}}$: The CIWAE estimator.
          $\tilde{L}^n_{\text{CIWAE}}$: The trainable estimator for the lower bound of CIWAE.
$\mathbf{z}^{1:n} \longleftarrow$ CONDITIONAL-SAMPLING-ALGORITHM($\mathbf{x}, q_\phi, n$);
for $i = 1, \ldots, n$ do
    $\hat{w}_{\text{CIWAE},i} \longleftarrow \frac{p_\theta(\mathbf{x}|\mathbf{z}^i) \cdot p_\theta(\mathbf{z}^i)}{q_\phi(\mathbf{z}^i|\mathbf{z}^{1:i-1}, \mathbf{x})}$;
end
$\hat{w}^n_{\text{CIWAE}} \longleftarrow \frac{1}{n}\sum_{i=1}^n \hat{w}_{\text{CIWAE},i}$;
$\tilde{L}^n_{\text{CIWAE}} \longleftarrow \log \frac{1}{n}\sum_{i=1}^n \hat{w}_{\text{CIWAE},i}$;
return $\hat{w}^n_{CIWAE}, \tilde{L}^n_{CIWAE}$
```

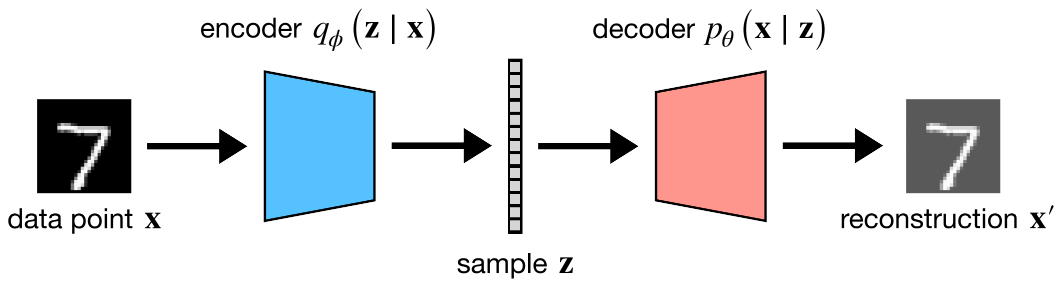

reconstructions $(\mathbf{x}')^{1:n}$
decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$
(shared weights)
feature extractor
$f_\phi(\mathbf{x})$
$\mathbf{z}^1$
$\mathbf{z}^2$
$\mathbf{z}^n$
data point $\mathbf{x}$
GRU cell
$g_\phi(\mathbf{z}^i \mid \mathbf{z}^{1:n-1}, \mathbf{x})$

(a) A schematic representation of our CIWAE approach implemented with neural networks modeling the involved distributions.

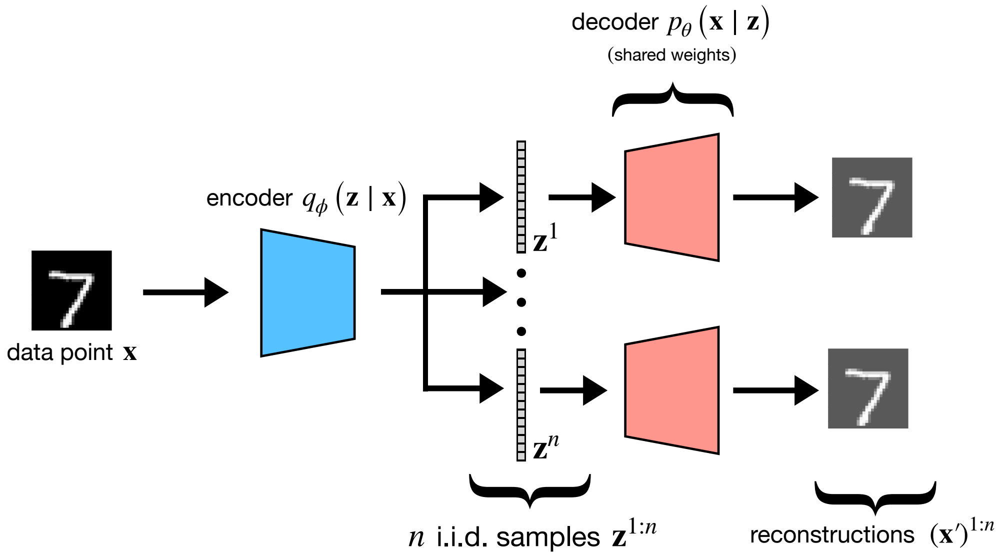

decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$
(shared weights)
$\left\lfloor \frac{n}{2} \right\rfloor$ antithetic samples
$\mathbf{z}^{\lceil \frac{n}{2}+i \rceil} = 2 \cdot \mu - \mathbf{z}^i \quad \left(1 \leq i \leq \left\lfloor \frac{n}{2} \right\rfloor\right)$
$\mathbf{z}^{\lceil \frac{n}{2}+i \rceil}$
encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$
$\mathbf{z}^1$
$\mathbf{z}^{\lceil \frac{n}{2} \rceil}$
data point $\mathbf{x}$
$\left\lceil \frac{n}{2} \right\rceil$ i.i.d. samples $\mathbf{z}^{1:\lceil \frac{n}{2} \rceil}$
reconstructions $(\mathbf{x}')^{1:n}$

(b) A schematic representation of the AIWAE approach (Klys et al. (2018)) implemented with neural networks modeling the involved distributions.

Fig. 4.1. The schematic representations of the CIWAE and AIWAE approaches. For comparison and contrast, the schematic representations of the VAE and IWAE approaches are given in Figure 4.2.


encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$
decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$
data point $\mathbf{x}$
sample $\mathbf{z}$
reconstruction $\mathbf{x}'$

(a) A schematic representation of the VAE approach (Kingma and Welling (2013)) implemented with neural networks modeling the involved distributions.


decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$
(shared weights)
encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$
data point $\mathbf{x}$
$\mathbf{z}^1$
$\mathbf{z}^n$
n i.i.d. samples $\mathbf{z}^{1:n}$
reconstructions $(\mathbf{x}')^{1:n}$

(b) A schematic representation of the IWAE approach (Burda et al. (2015)) implemented with neural networks modeling the involved distributions.

Fig. 4.2. The schematic representations of the VAE (Kingma and Welling (2013)) and IWAE (Burda et al. (2015)) approaches.

Figure 4.1 (a) shows the schematic representation of the neural networks involved in implementing the CIWAE approach. As discussed previously, the CIWAE approach performs conditional sampling where each next sample $\mathbf{z}^i$ is sampled conditioned on all the previous samples: $\mathbf{z}^i \sim q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}, \mathbf{x}\right)$, which requires parametrizing an encoder that can model dependencies on the variable number of samples $\mathbf{z}^{1:i-1}$. We achieve this by modeling the encoder in terms of **1.** a feature extractor $f_\phi(\mathbf{x})$ that outputs an encoding of the input data point $\mathbf{x}$, and **2.** a gated recurrent unit (GRU) cell (Cho et al. (2014)) that can model the conditioning of $\mathbf{z}^{1:i-1}$ and $\mathbf{x}$ required to sample $\mathbf{z}^i$. The hidden state h^0 of the GRU cell is initialized with the encoding of the input $\mathbf{x}$. For obtaining the first sample, the output of the GRU cell is used and the updated hidden state $\mathbf{h}^1$ contains an encoding of $\mathbf{x}$. In general, the previous sample $\mathbf{z}^{i-1}$ is set as the input to the cell and the hidden state $\mathbf{h}^{i-1}$ containing the encoding of $\mathbf{x}, \mathbf{z}^{1:i-2}$ is used to obtain the updated state $\mathbf{h}^i$ containing an encoding of $\mathbf{x}, \mathbf{z}^{1:i-2}$ and the output is used to sample the next latent $\mathbf{z}^i$.

4.3. Properties of Conditional-IWAE

Now, having seen the implementation details of the CIWAE approach, we prove through the following theorems the properties of the rate of decrease of its variational gap.

Theorem 4.3.1. The estimator $\hat{w}_{\text{CIWAE},i} = \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1},\mathbf{x}\right)}$ is an unbiased estimator of $p(\mathbf{x})$, i.e., $\mathbb{E}\left[\hat{w}_{\text{CIWAE},i}\right] = \mathbb{E}_{q(\mathbf{z}^{1:n}\mid\mathbf{x})}\left[\frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1},\mathbf{x}\right)}\right] = p(\mathbf{x}) \ \forall\, i \in \{1,\dots,n\}$.

Proof . Consider the following manipulations.

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{z}^{1:n}\mid\mathbf{x})}\left[\frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1},\mathbf{x}\right)}\right] = \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i \int_{\mathbf{z}^{i+1:n}} \mathrm{d}\mathbf{z}^{i+1:n} q\left(\mathbf{z}^{1:n} \mid \mathbf{x}\right) \cdot \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1},\mathbf{x}\right)} \\
&=^* \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1},\mathbf{x}\right)} \int_{\mathbf{z}^{i+1:n}} \mathrm{d}\mathbf{z}^{i+1:n} q\left(\mathbf{z}^{1:n} \mid \mathbf{x}\right) \\
&=^\dagger \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1},\mathbf{x}\right)} \cdot q\left(\mathbf{z}^{1:i} \mid \mathbf{x}\right) \\
&=^\ddagger \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i p\left(\mathbf{x},\mathbf{z}^i\right) \cdot q\left(\mathbf{z}^{1:i-1} \mid \mathbf{x}\right) = \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} q\left(\mathbf{z}^{1:i-1} \mid \mathbf{x}\right) \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i p\left(\mathbf{x},\mathbf{z}^i\right) \\
&= \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} q\left(\mathbf{z}^{1:i-1} \mid \mathbf{x}\right) \cdot p(\mathbf{x}) = p(\mathbf{x}) \cdot \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} q\left(\mathbf{z}^{1:i-1} \mid \mathbf{x}\right) = p(\mathbf{x}) \cdot 1 = p(\mathbf{x})
\end{aligned} \tag{4.3.1}$$

Here, the equalities $(*), (\dagger), (\ddagger)$ follow from marginalization of appropriate variables. The order of integration for the positive integrand can be shuffled due to Tonelli's theorem. $\square$

Theorem 4.3.2. The estimators $\hat{w}_{\text{CIWAE},i} = \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q(\mathbf{z}^i|\mathbf{z}^{1:i-1},\mathbf{x})}$, $\hat{w}_{\text{CIWAE},j} = \frac{p\left(\mathbf{x},\mathbf{z}^j\right)}{q(\mathbf{z}^j|\mathbf{z}^{1:j-1},\mathbf{x})}$ are pairwise un-correlated for all i, j $(1 \le j < i \le n)$, i.e. $\mathbb{E}\left[\hat{w}_{\text{CIWAE},i} \cdot \hat{w}_{\text{CIWAE},j}\right] = \mathbb{E}_{q(\mathbf{z}^{1:n}|\mathbf{x})}\left[\frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q(\mathbf{z}^i|\mathbf{z}^{1:i-1},\mathbf{x})} \cdot \frac{p\left(\mathbf{x},\mathbf{z}^j\right)}{q(\mathbf{z}^j|\mathbf{z}^{1:j-1},\mathbf{x})}\right] = p\left(\mathbf{x}\right)^2 = \mathbb{E}\left[\hat{w}_{\text{CIWAE},i}\right] \cdot \mathbb{E}\left[\hat{w}_{\text{CIWAE},j}\right]$.

PROOF . Consider the following manipulations for any i, j $(1 \le i < j \le n)$.

$$\begin{aligned}
&\mathbb{E}\left[\hat{w}_{\text{CIWAE},i} \cdot \hat{w}_{\text{CIWAE},j}\right] = \mathbb{E}_{q(\mathbf{z}^{1:n}|\mathbf{x})}\left[\frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}\right)} \cdot \frac{p\left(\mathbf{x},\mathbf{z}^j\right)}{q\left(\mathbf{z}^j \mid \mathbf{z}^{1:j-1}\right)}\right] \\
&=^{(1)} \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i \int_{\mathbf{z}^{i+1:j-1}} \mathrm{d}\mathbf{z}^{i+1:j-1} \int_{\mathbf{z}^j} \mathrm{d}\mathbf{z}^j \int_{\mathbf{z}^{j+1:n}} \mathrm{d}\mathbf{z}^{j+1:n} \\
&\qquad q\left(\mathbf{z}^{1:n} \mid \mathbf{x}\right) \cdot \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}\right)} \cdot \frac{p\left(\mathbf{x},\mathbf{z}^j\right)}{q\left(\mathbf{z}^j \mid \mathbf{z}^{1:j-1}\right)} \\
&=^{(2)} \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i \int_{\mathbf{z}^{i+1:j-1}} \mathrm{d}\mathbf{z}^{i+1:j-1} \int_{\mathbf{z}^j} \mathrm{d}\mathbf{z}^j \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}\right)} \cdot \frac{p\left(\mathbf{x},\mathbf{z}^j\right)}{q\left(\mathbf{z}^j \mid \mathbf{z}^{1:j-1}\right)} \\
&\qquad \int_{\mathbf{z}^{j+1:n}} \mathrm{d}\mathbf{z}^{j+1:n}\, q\left(\mathbf{z}^{1:n} \mid \mathbf{x}\right) \\
&=^{(3)} \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i \int_{\mathbf{z}^{i+1:j-1}} \mathrm{d}\mathbf{z}^{i+1:j-1} \int_{\mathbf{z}^j} \mathrm{d}\mathbf{z}^j\, q\left(\mathbf{z}^{1:j} \mid \mathbf{x}\right) \cdot \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}\right)} \cdot \frac{p\left(\mathbf{x},\mathbf{z}^j\right)}{q\left(\mathbf{z}^j \mid \mathbf{z}^{1:j-1}\right)} \\
&=^{(4)} \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i \int_{\mathbf{z}^{i+1:j-1}} \mathrm{d}\mathbf{z}^{i+1:j-1} \int_{\mathbf{z}^j} \mathrm{d}\mathbf{z}^j\, q\left(\mathbf{z}^{1:j-1} \mid \mathbf{x}\right) \cdot \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}\right)} \cdot p\left(\mathbf{x},\mathbf{z}^j\right) \\
&=^{(5)} \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i \int_{\mathbf{z}^{i+1:j-1}} \mathrm{d}\mathbf{z}^{i+1:j-1}\, q\left(\mathbf{z}^{1:j-1} \mid \mathbf{x}\right) \cdot \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}\right)} \int_{\mathbf{z}^j} \mathrm{d}\mathbf{z}^j\, p\left(\mathbf{x},\mathbf{z}^j\right) \\
&=^{(6)} p\left(\mathbf{x}\right) \cdot \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i \int_{\mathbf{z}^{i+1:j-1}} \mathrm{d}\mathbf{z}^{i+1:j-1}\, q\left(\mathbf{z}^{1:j-1} \mid \mathbf{x}\right) \cdot \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}\right)} \\
&=^{(7)} p\left(\mathbf{x}\right) \cdot \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}\right)} \int_{\mathbf{z}^{i+1:j-1}} \mathrm{d}\mathbf{z}^{i+1:j-1}\, q\left(\mathbf{z}^{1:j-1} \mid \mathbf{x}\right) \\
&=^{(8)} p\left(\mathbf{x}\right) \cdot \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i\, q\left(\mathbf{z}^{1:i} \mid \mathbf{x}\right) \cdot \frac{p\left(\mathbf{x},\mathbf{z}^i\right)}{q\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}\right)} \\
&=^{(9)} p\left(\mathbf{x}\right) \cdot \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1} \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i\, q\left(\mathbf{z}^{1:i-1} \mid \mathbf{x}\right) \cdot p\left(\mathbf{x},\mathbf{z}^i\right) \\
&=^{(10)} p\left(\mathbf{x}\right) \cdot \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1}\, q\left(\mathbf{z}^{1:i-1} \mid \mathbf{x}\right) \int_{\mathbf{z}^i} \mathrm{d}\mathbf{z}^i\, p\left(\mathbf{x},\mathbf{z}^i\right) \\
&=^{(11)} p\left(\mathbf{x}\right) \cdot p\left(\mathbf{x}\right) \cdot \int_{\mathbf{z}^{1:i-1}} \mathrm{d}\mathbf{z}^{1:i-1}\, q\left(\mathbf{z}^{1:i-1} \mid \mathbf{x}\right) = p\left(\mathbf{x}\right)^2 = \mathbb{E}\left[\hat{w}_{\text{CIWAE},i}\right] \cdot \mathbb{E}\left[\hat{w}_{\text{CIWAE},j}\right]
\end{aligned} \tag{4.3.2}$$

Here, the equality (1) follows form the definition of the correlation, the equalities $(3), (6), (8), (10), (11)$ follow by marginalizing over the corresponding latent variables, and the equalities $(4), (9)$ follow by chain rule of probabilities.

Theorem 4.3.3. Consider the CIWAE estimator $\hat{w}^n_{\text{CIWAE}} = \frac{1}{n}\sum_{i=1}^n \hat{w}_{\text{CIWAE},i} = \frac{1}{n}\sum_{i=1}^n \frac{p(\mathbf{x},\mathbf{z}^i)}{q(\mathbf{z}^i|\mathbf{z}^{1:i-1},\mathbf{x})}$ defined in terms of n latent variables $\mathbf{z}^i$ $(i \in \{1,\ldots,n\})$, where sample $\mathbf{z}^i$ is sampled conditioned on all the previous samples $\mathbf{z}^{1:i-1}$. Let $\text{Variance}\left(\hat{w}_{\text{CIWAE},i}\right)$ be upper bounded by a finite constant, i.e., let $\exists\ B > 0$ such that $\text{Variance}\left(\hat{w}_{\text{CIWAE},i}\right) < B\ \forall\ i \in \{1,\ldots,n\}$. Then, $\text{Variance}\left(\hat{w}^n_{\text{CIWAE}}\right) = \mathcal{O}\left(\frac{1}{n}\right)$.

Proof . The proof follows directly from Theorem 4.3.2 as follows.

$$\begin{aligned}\text{Variance}\left(\hat{w}^n_{\text{CIWAE}}\right) &= \text{Variance}\left(\frac{1}{n}\sum_{i=1}^n \hat{w}_{\text{CIWAE},i}\right) =^{*} \left(\frac{1}{n}\right)^2 \text{Variance}\left(\sum_{i=1}^n \hat{w}_{\text{CIWAE},i}\right)\\ &=^{\dagger} \frac{1}{n^2}\cdot\left(\sum_{i=1}^n \text{Variance}\left(\hat{w}_{\text{CIWAE},i}\right) + 2\cdot\sum_{1\le i<j\le n}\text{Covariance}\left(\hat{w}_{\text{CIWAE},i}\cdot\hat{w}_{\text{CIWAE},j}\right)\right)\\ &=^{\ddagger} \frac{1}{n^2}\cdot\left(\sum_{i=1}^n \text{Variance}\left(\hat{w}_{\text{CIWAE},i}\right) + 0\right) < \frac{1}{n^2}\cdot\sum_{i=1}^n B = \frac{B\cdot n}{n^2} = \frac{B}{n}\end{aligned} \tag{4.3.3}$$

Here, the equalities $(*), (\dagger)$ follow from standard properties of variance, and the equality $(\ddagger)$ follows from Theorem 4.3.2. □

Theorem 4.3.4. Consider the CIWAE estimator $\hat{w}^n_{\text{CIWAE}} = \frac{1}{n}\sum_{i=1}^n \hat{w}_{\text{CIWAE},i} = \frac{1}{n}\sum_{i=1}^n \frac{p(\mathbf{x},\mathbf{z}^i)}{q(\mathbf{z}^i|\mathbf{z}^{1:i-1},\mathbf{x})}$ defined in terms of n latent variables $\mathbf{z}^i$ $(i \in \{1,\ldots,n\})$, where sample $\mathbf{z}^i$ is sampled conditioned on all the previous samples $\mathbf{z}^{1:i-1}$. Consider the corresponding variational lower bound $\hat{L}^n_{\text{CIWAE}} = \mathbb{E}\left[\log \hat{w}^n_{\text{CIWAE}}\right]$. Let **1.** $\text{Variance}\left(\hat{w}_{\text{CIWAE},i}\right)$ be upper bounded by a finite constant, i.e., let $\exists\ B > 0$ such that $\text{Variance}\left(\hat{w}_{\text{CIWAE},i}\right) < B\ \forall\ i \in \{1,\ldots,n\}$, and let **2.** $\hat{w}_{\text{CIWAE},i}$ be lower bounded estimators, i.e, let $\exists\ c > 0$ such that $\hat{w}_{\text{CIWAE},i} > c\ \forall\ i \in \{1,\ldots,n\}$. Then, $\hat{w}^n_{\text{CIWAE}}$ is a variationally asymptotic Monte-Carlo estimator of $p(\mathbf{x})$, i.e., $\hat{w}^n_{\text{CIWAE}}$ satisfies: **1.** $\mathbb{E}\left[\hat{w}^n_{\text{CIWAE}}\right] = p(\mathbf{x})$, **2.** $\log p(\mathbf{x}) \ge \hat{L}^n_{\text{CIWAE}}$, **3.** $\mathcal{V}\left(\hat{w}^n_{\text{CIWAE}}\right) = \mathcal{O}\left(\frac{1}{n}\right)$, i.e., $\mathcal{V}\left(\hat{w}^n_{\text{CIWAE}}\right) \longrightarrow 0$ and $\hat{L}^n_{\text{CIWAE}} \longrightarrow \log p(\mathbf{x})$ at the rate $\mathcal{O}\left(\frac{1}{n}\right)$.

PROOF . **1.** Consider the following manipulations.

$$\mathbb{E}\left[\hat{w}^n_{\text{CIWAE}}\right] = \mathbb{E}\left[\frac{1}{n}\sum\nolimits_{i=1}^{n}\hat{w}_{\text{CIWAE},i}\right] = \frac{1}{n}\sum\nolimits_{i=1}^{n}\mathbb{E}\left[\hat{w}_{\text{CIWAE},i}\right] =^{*} \frac{1}{n}\sum\nolimits_{i=1}^{n} p\left(\mathbf{x}\right) = p\left(\mathbf{x}\right)$$

Here, the equality $(*)$ follows from Theorem 4.3.1.

2. Since part **1** proves that $\hat{w}^n_{\text{CIWAE}}$ is an unbiased estimator of $p\left(\mathbf{x}\right)$, Lemma 3.1.4 implies that $\log p\left(\mathbf{x}\right) \geq \mathbb{E}\left[\log \hat{w}^n_{\text{CIWAE}}\right] = \hat{L}^n_{\text{CIWAE}}$.

3. The given condition **1** along with Theorem 4.3.3 gives Variance $\left(\log \hat{w}^n_{\text{CIWAE}}\right) < \frac{B}{n}$. Further, the given condition **2** along with Theorem 3.3.1 gives that $\mathcal{V}\left(\hat{w}^n_{\text{CIWAE}}\right) \leq \frac{1}{2\cdot c^2} \cdot$ Variance $\left(\hat{w}^n_{\text{CIWAE}}\right)$. Combining these two conditions, we have $\mathcal{V}\left(\hat{w}^n_{\text{CIWAE}}\right) < \frac{B}{2\cdot c^2\cdot n}$. Thus, $\mathcal{V}\left(\hat{w}^n_{\text{CIWAE}}\right) = \mathcal{O}\left(\frac{1}{n}\right)$, which proves the desired result. □

4.4. Antithetic-IWAE

As seen in Section 4.2, the CIWAE approach utilizes conditionally sampled latent variables and a corrected estimator. The conditionally sampled latent variables are targeted at increasing the sample efficiency of the approach and the corrected estimators $\hat{w}_{\text{CIWAE},i} = \frac{p(\mathbf{x}),\mathbf{z}^i}{q(\mathbf{z}^i|\mathbf{z}^{1:i-1}|\mathbf{x})}$. ensure that they are always pairwise uncorrelated, which results in $\mathcal{O}\left(\frac{1}{n}\right)$ variance properties of the corresponding CIWAE estimator $\hat{w}^n_{\text{CIWAE}} = \frac{1}{n}\sum_{i=1}^{n}\hat{w}_{\text{CIWAE},i}$. However, our other technique does not necessitate *all* the estimators uncorrelated; we sample the latent variables in an antithetic manner and use the regular IWAE estimators as shown in Algorithm 2. The antithetic sampling leads to sufficiently many estimators to become uncorrelated so that the $\mathcal{O}\left(\frac{1}{n}\right)$ variance properties hold for the corresponding AIWAE estimator $\hat{w}^n_{\text{AIWAE}} = \frac{1}{n}\sum_{i=1}^{n}\hat{w}_{\text{AIWAE},i}$. Note that in AIWAE, we effectively generate only half the number $\left(\left\lceil\frac{n}{2}\right\rceil\right)$ of samples from the proposal $q\left(\cdot \mid \mathbf{x}\right)$. These samples are then used to create the remaining $n - \left\lceil\frac{n}{2}\right\rceil = \left\lfloor\frac{n}{2}\right\rfloor$ samples by reflecting these samples in the mean μ of the distribution modeled by $q\left(\cdot \mid \mathbf{x}\right)$. Thus, from Algorithm 2, we can see that $\mathbf{z}^i$ and $\mathbf{z}^{\left\lceil\frac{n}{2}\right\rceil+i}$ are the only pairs of correlated samples $\forall\ i \in \left\lfloor\frac{n}{2}\right\rfloor$; any other pair of samples is independent of each other. Thus, only the estimators $\hat{w}_{\text{AIWAE},i}$ and $\hat{w}_{\text{AIWAE},\left\lceil\frac{n}{2}\right\rceil+i}$ can be correlated and there are only $\mathcal{O}\left(n\right)$ number of these contributions to the variance of the $\hat{w}^n_{\text{AIWAE}}$, making it have the $\mathcal{O}\left(\frac{1}{n}\right)$ variance properties. We will prove these results in the next Section. Note that the schematic diagram of the implementation of the AIWAE approach is given in Figure 4.1 (b).

Algorithm 2: Antithetic sampling algorithm and the AIWAE approach.

```
function: ANTITHETIC-SAMPLING-ALGORITHM
inputs   : $\mathbf{x} \sim p(\cdot)$: A data point, where $p(\cdot)$ represents the data distribution.
           $q_\phi(\mathbf{z} \mid \mathbf{x})$: The parametrized encoder (neural network).
           $n$: The number of conditional samples to be used for inference.
outputs : $\mathbf{z}^{1:n}$: $n$ antithetically sampled latent variables.
for $i = 1, \dots, \left\lceil \frac{n}{2} \right\rceil$ do
    Sample $\mathbf{z}^i \overset{\text{iid}}{\sim} q_\phi(\cdot \mid \mathbf{x})$;
end
$\mu \longleftarrow$ Mean of the distribution modeled by $q(\cdot \mid \mathbf{x})$;
for $i = 1, \dots, \left\lfloor \frac{n}{2} \right\rfloor$ do
    Set $\mathbf{z}^{\left\lceil \frac{n}{2} \right\rceil + i} = 2 \cdot \mu - \mathbf{z}^i$ ;
end
return $\mathbf{z}^{1:n}$;
```

```
function: AIWAE
inputs   : $\mathbf{x} \sim p(\cdot)$: A data point, where $p(\cdot)$ represents the data distribution.
           $q_\phi(\mathbf{z} \mid \mathbf{x})$: The parametrized encoder (neural network).
           $p_\theta(\mathbf{x} \mid \mathbf{z})$: The parametrized decoder (neural network).
           $p_\theta(\mathbf{z})$: The prior on all latent variables.
           $n$: The number of conditional samples to be used for inference.
outputs : $\hat{w}^n_{\text{AIWAE}}$: The AIWAE estimator.
           $\tilde{L}^n_{\text{AIWAE}}$: The trainable estimator for the lower bound of AIWAE.
$\mathbf{z}^{1:n} \longleftarrow$ ANTITHETIC-SAMPLING-ALGORITHM($\mathbf{x}, q_\phi, n$);
for $i = 1, \dots, n$ do
    $\hat{w}_{\text{AIWAE},i} \longleftarrow \frac{p_\theta(\mathbf{x}|\mathbf{z}^i) \cdot p_\theta(\mathbf{z}^i)}{q_\phi(\mathbf{z}^i|\mathbf{x})}$;
end
$\hat{w}^n_{\text{AIWAE}} \longleftarrow \frac{1}{n} \sum_{i=1}^n \hat{w}_{\text{AIWAE},i}$;
$\tilde{L}^n_{\text{AIWAE}} \longleftarrow \log \frac{1}{n} \sum_{i=1}^n \hat{w}_{\text{AIWAE},i}$;
return $\hat{w}^n_{AIWAE}, \tilde{L}^n_{AIWAE}$
```

4.5. Properties of Antithetic-IWAE

Now, we prove through the following theorems the properties of the rate of decrease of the variational gap of the AIWAE approach.

Theorem 4.5.1. For *any* latent variables $\mathbf{v}^{1:n}$ with joint distribution $q\left(\mathbf{v}^{1:n} \mid \mathbf{x}\right)$, we have $\mathbb{E}_{q(\mathbf{v}^{1:n}|\mathbf{x})}\left[\frac{p\left(\mathbf{x},\mathbf{v}^{i}\right)}{q(\mathbf{v}^{i}|\mathbf{x})}\right] = p\left(\mathbf{x}\right)\ \forall\ i \in \{1,\ldots,n\}$. In particular, let $\mathbf{z}^{1:n}$ be the n latent variables sampled using the Antithetic Sampling Algorithm 2. Then, as a special case, the estimator $\hat{w}_{\text{AIWAE},i} = \frac{p\left(\mathbf{x},\mathbf{z}^{i}\right)}{q(\mathbf{z}^{i}|\mathbf{x})}$ is an unbiased estimator of $p\left(\mathbf{x}\right)$, i.e., $\mathbb{E}\left[\hat{w}_{\text{AIWAE},i}\right] = \mathbb{E}_{q(\mathbf{z}^{1:n}|\mathbf{x})}\left[\frac{p\left(\mathbf{x},\mathbf{z}^{i}\right)}{q(\mathbf{z}^{i}|\mathbf{x})}\right] = p\left(\mathbf{x}\right)\ \forall\ i \in \{1,\ldots,n\}$.

Proof . Consider the following manipulations.

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{v}^{1:n}|\mathbf{x})}\left[\frac{p\left(\mathbf{x},\mathbf{v}^{i}\right)}{q\left(\mathbf{v}^{i}\mid\mathbf{x}\right)}\right] &= \int_{\mathbf{v}^{i}} d\mathbf{v}^{i} \int_{\mathbf{v}^{1:i-1}} d\mathbf{v}^{1:i-1} \int_{\mathbf{v}^{i+1:n}} d\mathbf{v}^{i+1:n}\, q\left(\mathbf{v}^{1:n}\mid\mathbf{x}\right)\cdot\frac{p\left(\mathbf{x},\mathbf{v}^{i}\right)}{q\left(\mathbf{v}^{i}\mid\mathbf{x}\right)} \\
&= \int_{\mathbf{v}^{i}} d\mathbf{v}^{i}\, \frac{p\left(\mathbf{x},\mathbf{v}^{i}\right)}{q\left(\mathbf{v}^{i}\mid\mathbf{x}\right)} \int_{\mathbf{v}^{1:i-1}} d\mathbf{v}^{1:i-1} \int_{\mathbf{v}^{i+1:n}} d\mathbf{v}^{i+1:n}\, q\left(\mathbf{v}^{1:n}\mid\mathbf{x}\right) \\
&=^{*} \int_{\mathbf{v}^{i}} d\mathbf{v}^{i}\, \frac{p\left(\mathbf{x},\mathbf{v}^{i}\right)}{q\left(\mathbf{v}^{i}\mid\mathbf{x}\right)} \int_{\mathbf{v}^{1:i-1}} d\mathbf{v}^{1:i-1}\, q\left(\mathbf{v}^{1:i}\mid\mathbf{x}\right) \\
&=^{\dagger} \int_{\mathbf{v}^{i}} d\mathbf{v}^{i}\, q\left(\mathbf{v}^{i}\mid\mathbf{x}\right)\cdot\frac{p\left(\mathbf{x},\mathbf{v}^{i}\right)}{q\left(\mathbf{v}^{i}\mid\mathbf{x}\right)} = \int_{\mathbf{v}^{i}} d\mathbf{v}^{i}\, p\left(\mathbf{x},\mathbf{v}^{i}\right) =^{\ddagger} p\left(\mathbf{x}\right)
\end{aligned} \tag{4.5.1}$$

Here, the equalities $(*), (\dagger), (\ddagger)$ follow from the marginalization of the corresponding latent variables and this completes the proof of the general case. □

Theorem 4.5.2. Let $\mathbf{z}^{1:n}$ be the n latent variables sampled from the proposal $q\left(\cdot\mid\mathbf{x}\right)$ using the Antithetic Sampling Algorithm 2. Consider the estimators $\hat{w}_{\text{AIWAE},i} = \frac{p\left(\mathbf{x},\mathbf{z}^{i}\right)}{q(\mathbf{z}^{i}|\mathbf{x})}\ \forall\ i \in \{1,\ldots,n\}$. Let Variance $\left(\hat{w}_{\text{AIWAE},i}\right)$ be upper bounded by a finite constant, i.e., let $\exists\ B > 0$ such that Variance $\left(\hat{w}_{\text{AIWAE},i}\right) < B\ \forall\ i \in \{1,\ldots,n\}$. Then, for the AIWAE estimator $\hat{w}^{n}_{\text{AIWAE}} = \frac{1}{n}\sum_{i=1}^{n}\hat{w}_{\text{AIWAE},i}$, we have Variance $\left(\hat{w}^{n}_{\text{AIWAE}}\right) = \mathcal{O}\left(\frac{1}{n}\right)$.

Proof . For latent variables $\mathbf{z}^{1:n}$ sampled from the proposal $q\left(\cdot\mid\mathbf{x}\right)$ using the Antithetic Sampling Algorithm 2, we have the following relations.

$$\mathbf{z}^{i}\overset{\text{iid}}{\sim}q\left(\cdot\mid\mathbf{x}\right)\ \forall\ i \in \left\{1,\ldots,\left\lceil\frac{n}{2}\right\rceil\right\}, \text{ and } \mathbf{z}^{\lceil\frac{n}{2}\rceil+j} = 2\cdot\mu\left(\mathbf{z}^{j}\right) - \mathbf{z}^{j}\ \forall\ j \in \left\{1,\ldots,\left\lfloor\frac{n}{2}\right\rfloor\right\} \tag{4.5.2}$$

Now, note that $\forall\ j \in \left\{1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor\right\}$, $\mathbf{z}^{\left\lceil \frac{n}{2} \right\rceil + j}$ is a function of $\mathbf{z}^j$, which in turn are independent. We use the fact that the functions of independent random variables are themselves independent. This and Expressions 4.5.2 give the following relations in the samples.

$$\begin{aligned}
&\mathbf{z}^s \text{ is independent of } \mathbf{z}^t\ \forall\ s,t \in \left\{1, \ldots, \left\lceil \frac{n}{2} \right\rceil\right\} \text{ with } s \neq t \\
&\mathbf{z}^s \text{ is independent of } \mathbf{z}^t\ \forall\ s,t \in \left\{\left\lceil \frac{n}{2} \right\rceil + 1, \ldots, n\right\} \text{ with } s \neq t \\
&\mathbf{z}^s \text{ is independent of } \mathbf{z}^t\ \forall\ s \in \left\{1, \ldots, \left\lceil \frac{n}{2} \right\rceil\right\} \text{ and } \forall\ t \in \left\{\left\lceil \frac{n}{2} \right\rceil + 1, \ldots, n\right\}
\end{aligned} \tag{4.5.3}$$

Now, for a given data point $\mathbf{x}$, the estimators $\hat{w}_{\text{AIWAE},i} = \frac{p(\mathbf{x},\mathbf{z}^i)}{q(\mathbf{z}^i|\mathbf{x})}$ can be viewed as functions of the corresponding latent variable $\mathbf{z}^i$. This and Expressions 4.5.2 give the following relation in the estimators $\hat{w}_{\text{AIWAE},i}$ corresponding to the samples $\mathbf{z}^i$.

$$\begin{aligned}
&\hat{w}_{\text{AIWAE},s} \text{ is independent of } \hat{w}_{\text{AIWAE},t}\ \forall\ s,t \in \left\{1, \ldots, \left\lceil \frac{n}{2} \right\rceil\right\} \text{ with } s \neq t \\
&\hat{w}_{\text{AIWAE},s} \text{ is independent of } \hat{w}_{\text{AIWAE},t}\ \forall\ s,t \in \left\{\left\lceil \frac{n}{2} \right\rceil + 1, \ldots, n\right\} \text{ with } s \neq t \\
&\hat{w}_{\text{AIWAE},s} \text{ is independent of } \hat{w}_{\text{AIWAE},t}\ \forall\ s \in \left\{1, \ldots, \left\lceil \frac{n}{2} \right\rceil\right\} \text{ and } \forall\ t \in \left\{\left\lceil \frac{n}{2} \right\rceil + 1, \ldots, n\right\}
\end{aligned} \tag{4.5.4}$$

With these relations, we consider the variance of the AIWAE estimator.

$$\begin{aligned}
&\text{Variance}\left(\hat{w}^n_{\text{AIWAE}}\right) = \text{Variance}\left(\frac{1}{n}\sum_{i=1}^n \hat{w}_{\text{AIWAE},i}\right) =^* \left(\frac{1}{n}\right)^2 \text{Variance}\left(\sum_{i=1}^n \hat{w}_{\text{AIWAE},i}\right) \\
&=^\dagger \frac{1}{n^2}\left(\sum_{i=1}^n \text{Variance}\left(\hat{w}_{\text{AIWAE},i}\right) + 2 \cdot \sum_{1 \le i < j \le n} \text{Covariance}\left(\hat{w}_{\text{AIWAE},i}, \hat{w}_{\text{AIWAE},j}\right)\right) \\
&=^\ddagger \frac{1}{n^2}\left(\sum_{i=1}^n \text{Variance}\left(\hat{w}_{\text{AIWAE},i}\right) + 2 \cdot \sum_{i=1}^{\left\lfloor \frac{n}{2} \right\rfloor} \text{Covariance}\left(\hat{w}_{\text{AIWAE},i}, \hat{w}_{\text{AIWAE},\left\lceil \frac{n}{2} \right\rceil + i}\right)\right) \\
&\le \frac{1}{n^2}\left(\sum_{i=1}^n \text{Variance}\left(\hat{w}_{\text{AIWAE},i}\right) + 2 \cdot \sum_{i=1}^{\left\lfloor \frac{n}{2} \right\rfloor} \left|\text{Covariance}\left(\hat{w}_{\text{AIWAE},i}, \hat{w}_{\text{AIWAE},\left\lceil \frac{n}{2} \right\rceil + i}\right)\right|\right) \\
&\le^\star \frac{1}{n^2}\left(\sum_{i=1}^n \text{Variance}\left(\hat{w}_{\text{AIWAE},i}\right) + 2 \cdot \sum_{i=1}^{\left\lfloor \frac{n}{2} \right\rfloor} \sqrt{\text{Variance}\left(\hat{w}_{\text{AIWAE},i}\right) \cdot \text{Variance}\left(\hat{w}_{\text{AIWAE},\left\lceil \frac{n}{2} \right\rceil + i}\right)}\right) \\
&<^\S \frac{1}{n^2}\left(\sum_{i=1}^n B + 2 \cdot \sum_{i=1}^{\left\lfloor \frac{n}{2} \right\rfloor} \sqrt{B \cdot B}\right) = \frac{1}{n^2}\left(n \cdot B + 2 \cdot \left\lfloor \frac{n}{2} \right\rfloor \cdot B\right) \\
&\le \frac{1}{n^2}\left(n \cdot B + 2 \cdot \frac{n}{2} \cdot B\right) = \frac{2 \cdot B \cdot n}{n^2} = \frac{2 \cdot B}{n}
\end{aligned} \tag{4.5.5}$$

Here, the equalities $(*), (\dagger)$ follow from standard properties of variance of linear combination of random variables. The equality $(\ddagger)$ follows from relations from 4.5.5. The inequality $(\star)$ follows from **Cauchy-Schwarz ineqaulity**: for random variables A, B, we have

$|\text{Covariance}(A, B)| \leq \sqrt{\text{Variance}(A) \cdot \text{Variance}(B)}$. The inequality (§) follows from the assumption of an upper bound B on the variance of all the estimators $\hat{w}_{\text{AIWAE},i}$. Thus, $\text{Variance}(\hat{w}^n_{\text{AIWAE}}) = \mathcal{O}\left(\frac{1}{n}\right)$, proving the desired result. □

Theorem 4.5.3. Consider the AIWAE estimator $\hat{w}^n_{\text{AIWAE}} = \frac{1}{n}\sum_{i=1}^n \hat{w}_{\text{AIWAE},i} = \frac{1}{n}\sum_{i=1}^n \frac{p(\mathbf{x},\mathbf{z}^i)}{q(\mathbf{z}^i|\mathbf{x})}$, where $\mathbf{z}^{1:n}$ are n latent variables sampled from the proposal $q(\cdot \mid \mathbf{x})$ using the Antithetic Sampling Algorithm 2. Consider the corresponding variational lower bound $\hat{L}^n_{\text{AIWAE}} = \mathbb{E}[\log \hat{w}^n_{\text{AIWAE}}]$. Let **1.** $\text{Variance}\left(\hat{w}_{\text{AIWAE},i}\right)$ be upper bounded by a finite constant, i.e., let $\exists\ B > 0$ such that $\text{Variance}\left(\hat{w}_{\text{AIWAE},i}\right) < B\ \forall\ i \in \{1, \ldots, n\}$, and let **2.** $\hat{w}_{\text{AIWAE},i}$ be lower bounded estimators, i.e, let $\exists\ c > 0$ such that $\hat{w}_{\text{AIWAE},i} > c\ \forall\ i \in \{1, \ldots, n\}$. Then, $\hat{w}^n_{\text{AIWAE}}$ is a variationally asymptotic Monte-Carlo estimator of $p(\mathbf{x})$, i.e., $\hat{w}^n_{\text{AIWAE}}$ satisfies: **1.** $\mathbb{E}[\hat{w}^n_{\text{AIWAE}}] = p(\mathbf{x})$, **2.** $\log p(\mathbf{x}) \geq \hat{L}^n_{\text{AIWAE}}$, **3.** $\mathcal{V}(\hat{w}^n_{\text{AIWAE}}) = \mathcal{O}\left(\frac{1}{n}\right)$, i.e., $\mathcal{V}(\hat{w}^n_{\text{AIWAE}}) \longrightarrow 0$ and $\hat{L}^n_{\text{AIWAE}} \longrightarrow \log p(\mathbf{x})$ at the rate $\mathcal{O}\left(\frac{1}{n}\right)$.

Proof . **1.** Consider the following manipulations.

$$\mathbb{E}[\hat{w}^n_{\text{AIWAE}}] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \hat{w}_{\text{AIWAE},i}\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\hat{w}_{\text{AIWAE},i}\right] =^* \frac{1}{n}\sum_{i=1}^n p(\mathbf{x}) = p(\mathbf{x})$$

Here, the equality $(*)$ follows from Theorem 4.5.1.

2. Since part **1** proves that $\hat{w}^n_{\text{AIWAE}}$ is an unbiased estimator of $p(\mathbf{x})$, Lemma 3.1.4 implies that $\log p(\mathbf{x}) \geq \mathbb{E}[\log \hat{w}^n_{\text{AIWAE}}] = \hat{L}^n_{\text{AIWAE}}$.

The given condition **1** along with Theorem 4.5.2 gives $\text{Variance}(\log \hat{w}^n_{\text{AIWAE}}) < \frac{2 \cdot B}{n}$. Further, the given condition **2** along with Theorem 3.3.1 gives that $\mathcal{V}(\hat{w}^n_{\text{AIWAE}}) \leq \frac{1}{2 \cdot c^2} \cdot \text{Variance}(\hat{w}^n_{\text{AIWAE}})$. Combining these two conditions, we have $\mathcal{V}(\hat{w}^n_{\text{AIWAE}}) < \frac{2 \cdot B}{2 \cdot c^2 \cdot n} = \frac{B}{c^2 \cdot n}$. Thus, $\mathcal{V}(\hat{w}^n_{\text{AIWAE}}) = \mathcal{O}\left(\frac{1}{n}\right)$, which proves the desired result. □

4.6. Related Work

This section provides a brief survey of approaches related to CIWAE and AIWAE.

The CIWAE and AIWAE approaches generate multiple samples, each conditioned on either all or some of the previously generated samples, in order to achieve sample efficiency in multi-sample variational inference. The multiple samples effectively allow representing the proposal distribution to a greater *resolution.* This idea has been explored in the works of Huang et al. (2019). The works of Naesseth et al. (2018) also use a similar idea but they generate their multiple samples by combining the variational inference with sequential Monte-Carlo sampling. Cremer et al. (2017), Nowozin (2018) and Domke and Sheldon (2018) have demonstrated many interpretations of importance weighted VI that are in the same spirit as our work. Cremer et al. (2017) reinterpret IWAE as performing regular variational inference with a more complex proposal distribution, which matches our idea of modeling a joint distribution over latent representations in CIWAE. The interpretation by Nowozin (2018) views at the variational lower bound of IWAE as an estimator of the log-likelihood and provides a family of approaches that reduce the involved bias, which matches exactly with the idea of our work. Domke and Sheldon (2018) connect the idea of IWAE and defensive sampling, where the idea is to utilize a proposal q with wider spread to avoid the involved estimator from exploding due to q placing no density in regions where the true distribution has significant value. This technique helps in variance reduction, which is the core idea of our CIWAE and AIWAE approaches.

The AIWAE approach, originally from the works of Klys et al. (2018), is related to the idea of antithetic sampling and its role in variance reduction. The works of Owen (2013) provide the notion of antithetic variates, which build on negatively correlated estimators for variance reduction and are related to the idea of AIWAE. Similarly, Wu et al. (2019) demonstrate the approach of differentiable antithetic sampler along with the idea of using more representative samples for variance reduction, which matches our idea.

Many approaches that look at the problem of improving the variational inference problem from the point of view of constructing more powerful and expressive proposal distributions. Ranganath et al. (2016) and Huang et al. (2019) build on this idea and our CIWAE approach attempts to generalize these works. This is because we assume the most generic dependency structure among the involved latent variables; recall that CIWAE samples each next latent variable conditioned on all the previous approaches. Along the same line, many recent approaches consider multi-sample variational inference and learn a hierarchical proposal distribution. Some examples of such approaches are semi-implicit VI by Yin and Zhou (2018), doubly semi-implicit VI Molchanov et al. (2019), and their generalization importance-weighted hierarchical VI by Sobolev and Vetrov (2019).

Note that the recent work by Rainforth et al. (2018) demonstrates possible problems associated with a large number of samples. This work associates the number of samples with the signal-to-noise ratio involved in the training of the proposal distribution and thus, warns that tighter lower bounds may not always be beneficial. The work of Tucker et al. (2018) on doubly reparametrized gradients mitigates this issue.

Finally, while dealing with image data, we note that variational approaches are known to result in *blurry* generative models and their counterpart generative adversarial networks by Goodfellow et al. (2014) perform much better. However, with powerful neural architectures and using conditional sampling of latent variables, Vahdat and Kautz (2020) demonstrate high-quality image data generated by a VAE approach. Thus, the notion of conditional sampling of latent variables is not restricted only to theoretical considerations of VI but can lead to improvements in its practical aspects like producing better generative models!

4.7. Experimentation and Results

In this section, we show our experimentation with the CIWAE, AIWAE approaches and discuss the results. We carry out experiments on the two standard benchmarking datasets, MNIST (LeCun et al. (2010)) and OMNIGLOT (Lake et al. (2015)), which are also used by the previous works. Our baselines are the IWAE approach by Burda et al. (2015), which is one the key work that led to the idea of improving the VI with multiple samples, and the HIWAE approach by Huang et al. (2019), which, similar to our work, builds on the idea of modeling a hierarchical and more expressive proposal for improving over IWAE.

4.7.1. Dataset Details

Unfortunately, the experimentation in the field of VI is performed on many variants of the benchmarking dataset. For instance, the MNIST and OMNIGLOT datasets can be used as-is, with static binarization, or with dynamic binarization. These changes significantly change the performances of the same approach, thereby making a comparison of the effectiveness of different approaches difficult. Thus, we fix the following variants of the benchmarking datasets for which the experimental results of the baseline approaches are available.

- MNIST dataset with static binarization as provided by Larochelle and Murray (2011).
- OMNIGLOT dataset by Lake et al. (2015) with dynamic binarization.

Here, static binarization refers to a fixed quantization of all the pixels of the image data once and for all, and creating the dataset with these binarized images. This dataset is to be used as it is for training the model, without introducing any other forms of data augmentation techniques. For this reason, training on the statically binarized variants of any dataset is extremely difficult; it is not easy to harvest the benefits of regularization techniques like data augmentation. On the other hand, dynamic binarization does allow for image augmentation. In dynamic binarization, each pixel value $p \in [0, 1]$ of the image is treated as the probability that the pixel should be binarized to 1, i.e., we consider a Bernoulli (p) distribution at each pixel and take a sample out of it. The value of the sample, either 0 or 1, is set as the pixel value in the image. Thus, with dynamic binarization, each training image is essentially unique and different, which effectively regularizes the model better than any static binarization.

4.7.2. Architecture Details

We perform our experimentation with two major architecture choices, which we call **1.** CNN and **2.** MLP. The CNN architecture, as described in the works of Huang et al. (2018) and Huang et al. (2019), uses residual convolutional neural networks for modeling both the encoder $q_\phi(\cdot \mid \mathbf{x})$ and the decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$. In our AIWAE approach, we use the exact neural network architecture as used in the HIWAE baseline (Huang et al. (2019)). However, our CIWAE architecture requires an additional component, a GRU cell. Thus, for the CIWAE approach, we use a slight variation of encoder CNN architecture for the feature extractor neural network $f_\phi(\mathbf{x})$ and add a GRU cell $g_\phi(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}, \mathbf{x})$ to get the CNN-GRU architecture. The slight variation is done so that the number of trainable parameters in CIWAE and HIWAE are comparable, which eliminates the possibility that additional capacity is the reason for the performance improvement of the CIWAE approach. This gives us a set of approach-architecture pairs: **1.** (IWAE, CNN), **2.** (CIWAE, CNN-GRU), and **3.** (AIWAE, CNN). However, note that we can use the CIWAE encoder composed of the f_ϕ and g_ϕ components for carrying out the IWAE and the AIWAE approaches as well. Thus, we get two additional approach-architecture pairs: **4.** (IWAE, CNN-GRU) and **5.** (AIWAE, CNN-GRU). Thus, we have 5 different approach-architecture pairs in our experiments on the comparison with the HIWAE baseline by Huang et al. (2019) in Experiment 1.

However, the original IWAE experimentation uses a simple feed-forward neural network architecture, which we denote by MLP, for both the encoder and the decoder. Also, we observe in our experiments that the choice of architecture results in different performances for the same approach, and thus, we also perform experiments with IWAE, CIWAE and AIWAE approaches by replacing the residual convolutional neural networks with the feed-forward neural networks of the original IWAE experimentation by Burda et al. (2015). This gives us 5 more approach-architecture pairs: **1.** (IWAE, MLP), **2.** (CIWAE, MLP-GRU), and **3.** (AIWAE, MLP), **4.** (IWAE, MLP-GRU) and **5.** (AIWAE, MLP-GRU). Again, we slightly tweak the MLP-GRU architecture so that MLP and MLP-GRU architectures have a comparable number of trainable parameters. Experiment 2 demonstrates the comparison of the performances of these approach-architecture pairs and emphasizes the effect of the architecture choice on the proposed approaches.

The hyper-parameter settings for these two experiments are kept as identical to those in the experimentation of Huang et al. (2019) as possible. However, the hyper-parameter settings for the original IWAE experimentation of Burda et al. (2015) vary significantly, especially in the number of epochs (actually, the number of updates). Besides, Burda et al. (2015) demonstrate only 1 value per approach, whereas experiments of Huang et al. (2019) and our work demonstrate performance with multiple seeds. Thus, in Experiment 3, we use the best hyper-parameter settings possible and generate a single performance value for our approaches to show that our approaches perform significantly better as the number of samples increases.

4.7.3. Experiment 1: Comparison with IWAE-HIWAE Baselines

In this experiment, we compare our CIWAE and AIWAE approaches with our IWAE baselines, and the IWAE-HIWAE baselines of Huang et al. (2019). The results of the experiments on the MNIST dataset are documented in Table 4.1 and those on the OMNIGLOT dataset are documented in Table 4.2. In both these tables, we highlight certain results for the negative log-likelihood values NLL_{val} and NLL_{te} for the validation and the test split respectively. The $\hat{L}_{tr}$, $\hat{L}_{val}$, and $\hat{L}_{te}$ values are for reference and for reporting results in a manner consistent with Huang et al. (2019). We highlight the best approach with the CNN architecture and the best approach with the CNN-GRU architecture. If the IWAE baseline or the HIWAE approach of Huang et al. (2019) performs better than the two aforesaid best approaches, we highlight them. With this, we make the following observations.

(1) As the number of samples increases, the AIWAE and CIWAE approaches start to become competitive and even outperform the HIWAE baseline.
(2) The CNN architecture approaches usually perform better than the CNN-GRU approaches, which highlights the role of architecture in all the involved approaches.
(3) As the number of samples increases, AIWAE is observed to perform significantly better than other approaches in general.
(4) We also see that our baselines are stronger and can sometimes be better than the HIWAE approach and baselines of Huang et al. (2019).
(5) Our CIWAE approach performs competitively with our IWAE baseline but is not observed to be significantly better with this architecture setting.

4.7.4. Experiment 2: Effect of the Choice of Architecture

In this experiment, we compare our CIWAE and AIWAE approaches only with our IWAE baselines, where all the involved approaches are implemented with the MLP-based architectures. The results of the experiments on the MNIST dataset are documented in Table 4.3 and those on the OMNIGLOT dataset are documented in Table 4.4. In both these tables, we highlight certain results for the negative log-likelihood values NLL_{val} and NLL_{te} for the validation and the test split respectively. The $\hat{L}_{tr}$, $\hat{L}_{val}$, and $\hat{L}_{te}$ values are for reference and for reporting results in a manner consistent with Huang et al. (2019). We highlight the best approach with the CNN architecture and the best approach with the CNN-GRU architecture. With this, we make the following observations.

(1) As the number of samples increases, the AIWAE and CIWAE approaches start to become competitive, if not better, than the IWAE baseline.
(2) The MLP architecture approaches usually perform better than the MLP-GRU approaches, which again highlights the role of architecture in all the involved approaches.

(3) As the number of samples increases, both CIWAE and AIWAE are observed to perform better than the IWAE baseline.

(4) The performance of the CNN-based approaches is significantly better than that of the MLP-based approaches, which further emphasizes the role of the architecture in the performance of the approaches.

(5) Even when AIWAE and CIWAE approaches do not perform better than the IWAE baseline, they usually remain competitive in performance.

4.7.5. Experiment 3: Comparison with IWAE (Burda et al. (2015))

In this experiment, we compare our CIWAE and AIWAE approaches with the original IWAE baselines by Burda et al. (2015). The results with both the MNIST and OMNIGLOT datasets are documented in Table 4.5. As done in Burda et al. (2015), we compare our best performance values with those of the IWAE approach[1]. Note that the MLP-2 architecture in Table 4.5 refers to the use of two stochastic layers for the IWAE approach. With this, we make the following observations.

(1) As the number of samples increases, both the AIWAE and CIWAE approaches perform significantly better than the IWAE approach.

(2) The performance improvement is more evident in our approaches; at $n = 1$ our approaches perform poorly when compared against the IWAE but at $n = 50$, our approaches perform better.

(3) Our approaches even perform better than IWAE evaluated with two stochastic layers (MLP-2 of Table 4.5).

(4) Performance improvement is indeed related to sample efficiency but it is clearly visible in the experiments with the OMNIGLOT dataset. Note that the CIWAE approach with $n = 5$ performs better than (IWAE, MLP) and (IWAE, MLP-2) with $n = 50$. This emphasizes the desired sample efficiency using the conditional sampling of latent variables.

[1] Another reason for not performing multiple experiments with different seeds is that these experiments are computationally expensive; each run documented in Table 4.5 requires about 48 hours of training on the currently fastest RTX GPUs.

n	**Approach**	NLL$_{val}$	NLL$_{te}$	$\hat{L}_{tr}$	$\hat{L}_{val}$	$\hat{L}_{te}$
	IWAE (Huang et al. (2019))	$82.64_{\pm 0.11}$	$82.37_{\pm 0.12}$	$83.26_{\pm 0.10}$	$86.57_{\pm 0.11}$	$86.36_{\pm 0.15}$
	HIWAE (Huang et al. (2019))	$\mathbf{82.24_{\pm 0.05}}$	$\mathbf{81.96_{\pm 0.04}}$	$82.92_{\pm 0.17}$	$85.75_{\pm 0.08}$	$85.50_{\pm 0.08}$
	IWAE (Ours, CNN-GRU)	$82.80_{\pm 0.46}$	$82.63_{\pm 0.46}$	$83.46_{\pm 0.23}$	$86.77_{\pm 0.59}$	$86.64_{\pm 0.57}$
1	CIWAE (Ours, CNN-GRU)	$82.80_{\pm 0.46}$	$82.63_{\pm 0.46}$	$83.46_{\pm 0.23}$	$86.77_{\pm 0.59}$	$86.64_{\pm 0.57}$
	AIWAE (Ours, CNN-GRU)	$82.80_{\pm 0.46}$	$82.63_{\pm 0.46}$	$83.46_{\pm 0.23}$	$86.77_{\pm 0.59}$	$86.64_{\pm 0.57}$
	IWAE (Ours, CNN)	$82.30_{\pm 0.18}$	$82.10_{\pm 0.18}$	$83.17_{\pm 0.35}$	$86.20_{\pm 0.11}$	$86.08_{\pm 0.11}$
	AIWAE (Ours, CNN)	$82.30_{\pm 0.18}$	$82.10_{\pm 0.18}$	$83.17_{\pm 0.35}$	$86.20_{\pm 0.11}$	$86.08_{\pm 0.11}$
	IWAE (Huang et al. (2019))	$82.03_{\pm 0.04}$	$81.77_{\pm 0.04}$	$82.36_{\pm 0.20}$	$85.40_{\pm 0.05}$	$85.16_{\pm 0.03}$
	HIWAE (Huang et al. (2019))	$\mathbf{81.88_{\pm 0.35}}$	$\mathbf{81.60_{\pm 0.35}}$	$82.15_{\pm 0.60}$	$85.03_{\pm 0.65}$	$84.76_{\pm 0.64}$
	IWAE (Ours, CNN-GRU)	$\mathbf{81.94_{\pm 0.13}}$	$\mathbf{81.76_{\pm 0.14}}$	$82.25_{\pm 0.24}$	$85.29_{\pm 0.15}$	$85.21_{\pm 0.19}$
2	CIWAE (Ours, CNN-GRU)	$82.05_{\pm 0.23}$	$81.87_{\pm 0.23}$	$82.32_{\pm 0.16}$	$85.45_{\pm 0.22}$	$85.33_{\pm 0.21}$
	AIWAE (Ours, CNN-GRU)	$82.07_{\pm 0.10}$	$81.92_{\pm 0.10}$	$82.28_{\pm 0.16}$	$85.22_{\pm 0.11}$	$85.12_{\pm 0.09}$
	IWAE (Ours, CNN)	$\mathbf{81.72_{\pm 0.18}}$	$\mathbf{81.50_{\pm 0.19}}$	$82.13_{\pm 0.18}$	$85.11_{\pm 0.17}$	$84.95_{\pm 0.17}$
	AIWAE (Ours, CNN)	$81.81_{\pm 0.19}$	$81.61_{\pm 0.19}$	$82.13_{\pm 0.12}$	$84.92_{\pm 0.11}$	$84.79_{\pm 0.14}$
	IWAE (Huang et al. (2019))	$81.63_{\pm 0.04}$	$81.37_{\pm 0.04}$	$81.48_{\pm 0.17}$	$84.45_{\pm 0.06}$	$84.25_{\pm 0.08}$
	HIWAE (Huang et al. (2019))	$81.39_{\pm 0.09}$	$\mathbf{81.13_{\pm 0.09}}$	$81.28_{\pm 0.14}$	$84.04_{\pm 0.16}$	$83.79_{\pm 0.14}$
	IWAE (Ours, CNN-GRU)	$81.40_{\pm 0.09}$	$81.19_{\pm 0.10}$	$81.33_{\pm 0.15}$	$84.16_{\pm 0.10}$	$83.98_{\pm 0.12}$
5	CIWAE (Ours, CNN-GRU)	$81.40_{\pm 0.13}$	$81.20_{\pm 0.13}$	$81.36_{\pm 0.09}$	$84.15_{\pm 0.13}$	$84.01_{\pm 0.14}$
	AIWAE (Ours, CNN-GRU)	$\mathbf{81.33_{\pm 0.19}}$	$\mathbf{81.15_{\pm 0.20}}$	$81.16_{\pm 0.18}$	$84.00_{\pm 0.18}$	$83.86_{\pm 0.20}$
	IWAE (Ours, CNN)	$81.11_{\pm 0.07}$	$\mathbf{80.88_{\pm 0.05}}$	$81.07_{\pm 0.11}$	$83.87_{\pm 0.04}$	$83.67_{\pm 0.03}$
	AIWAE (Ours, CNN)	$\mathbf{81.10_{\pm 0.07}}$	$\mathbf{80.88_{\pm 0.08}}$	$81.19_{\pm 0.19}$	$83.79_{\pm 0.10}$	$83.63_{\pm 0.09}$
	IWAE (Huang et al. (2019))	$81.37_{\pm 0.05}$	$81.13_{\pm 0.02}$	$80.85_{\pm 0.16}$	$83.84_{\pm 0.06}$	$83.62_{\pm 0.03}$
	HIWAE (Huang et al. (2019))	$81.28_{\pm 0.08}$	$81.04_{\pm 0.09}$	$80.89_{\pm 0.13}$	$83.77_{\pm 0.23}$	$83.56_{\pm 0.22}$
	IWAE (Ours, CNN-GRU)	$81.02_{\pm 0.12}$	$80.81_{\pm 0.14}$	$80.73_{\pm 0.19}$	$83.37_{\pm 0.12}$	$83.19_{\pm 0.15}$
10	CIWAE (Ours, CNN-GRU)	$81.05_{\pm 0.04}$	$80.81_{\pm 0.05}$	$80.77_{\pm 0.15}$	$83.40_{\pm 0.06}$	$83.19_{\pm 0.08}$
	AIWAE (Ours, CNN-GRU)	$\mathbf{80.98_{\pm 0.10}}$	$\mathbf{80.76_{\pm 0.11}}$	$80.63_{\pm 0.23}$	$83.29_{\pm 0.09}$	$83.11_{\pm 0.12}$
	IWAE (Ours, CNN)	$\mathbf{80.80_{\pm 0.05}}$	$\mathbf{80.55_{\pm 0.05}}$	$80.66_{\pm 0.08}$	$83.15_{\pm 0.09}$	$82.92_{\pm 0.08}$
	AIWAE (Ours, CNN)	$80.83_{\pm 0.03}$	$80.58_{\pm 0.06}$	$80.72_{\pm 0.16}$	$83.13_{\pm 0.03}$	$82.92_{\pm 0.05}$

Table 4.1. The results of CIWAE and AIWAE experiments, along with the corresponding IWAE and HIWAE baselines of ours and by by Huang et al. (2019). The experiments are performed on the MNIST (statically binarized) dataset by Larochelle and Murray (2011).

n	**Approach**	NLL_{val}	NLL_{te}	$\hat{L}_{tr}$	$\hat{L}_{val}$	$\hat{L}_{te}$
1	IWAE (Huang et al. (2019))	$102.98_{\pm 1.56}$	$103.28_{\pm 1.53}$	$106.40_{\pm 1.75}$	$109.05_{\pm 1.28}$	$109.90_{\pm 1.21}$
	HIWAE (Huang et al. (2019))	$\mathbf{100.36_{\pm 0.50}}$	$\mathbf{100.79_{\pm 0.52}}$	$104.57_{\pm 0.98}$	$106.48_{\pm 0.48}$	$107.36_{\pm 0.52}$
	IWAE (Ours, CNN-GRU)	$105.31_{\pm 1.00}$	$105.68_{\pm 1.11}$	$106.11_{\pm 0.83}$	$111.50_{\pm 1.09}$	$112.42_{\pm 1.23}$
	CIWAE (Ours, CNN-GRU)	$105.31_{\pm 1.00}$	$105.68_{\pm 1.11}$	$106.11_{\pm 0.83}$	$111.50_{\pm 1.09}$	$112.42_{\pm 1.23}$
	AIWAE (Ours, CNN-GRU)	$105.31_{\pm 1.00}$	$105.68_{\pm 1.11}$	$106.11_{\pm 0.83}$	$111.50_{\pm 1.09}$	$112.42_{\pm 1.23}$
	IWAE (Ours, CNN)	$104.30_{\pm 2.47}$	$104.87_{\pm 2.37}$	$104.84_{\pm 1.66}$	$111.05_{\pm 2.17}$	$111.94_{\pm 2.05}$
	AIWAE (Ours, CNN)	$104.30_{\pm 2.47}$	$104.87_{\pm 2.37}$	$104.84_{\pm 1.66}$	$111.05_{\pm 2.17}$	$111.94_{\pm 2.05}$
2	IWAE (Huang et al. (2019))	$100.14_{\pm 0.24}$	$\mathbf{100.48_{\pm 0.34}}$	$102.66_{\pm 0.59}$	$105.85_{\pm 0.19}$	$106.70_{\pm 0.12}$
	HIWAE (Huang et al. (2019))	$\mathbf{99.68_{\pm 0.40}}$	$\mathbf{100.00_{\pm 0.49}}$	$102.88_{\pm 0.89}$	$105.18_{\pm 0.69}$	$105.93_{\pm 0.73}$
	IWAE (Ours, CNN-GRU)	$102.32_{\pm 0.53}$	$102.82_{\pm 0.52}$	$102.14_{\pm 0.45}$	$108.46_{\pm 0.47}$	$109.53_{\pm 0.51}$
	CIWAE (Ours, CNN-GRU)	$102.48_{\pm 1.05}$	$102.94_{\pm 0.98}$	$102.80_{\pm 0.78}$	$108.39_{\pm 1.40}$	$109.43_{\pm 1.15}$
	AIWAE (Ours, CNN-GRU)	$\mathbf{101.62_{\pm 0.88}}$	$\mathbf{101.91_{\pm 0.74}}$	$101.66_{\pm 0.97}$	$107.42_{\pm 0.61}$	$108.26_{\pm 0.59}$
	IWAE (Ours, CNN)	$100.39_{\pm 0.86}$	$\mathbf{100.81_{\pm 0.84}}$	$101.10_{\pm 0.38}$	$106.51_{\pm 0.78}$	$107.48_{\pm 0.91}$
	AIWAE (Ours, CNN)	$\mathbf{100.12_{\pm 1.29}}$	$100.82_{\pm 1.08}$	$100.94_{\pm 0.55}$	$106.01_{\pm 1.24}$	$107.04_{\pm 1.11}$
5	IWAE (Huang et al. (2019))	$99.43_{\pm 0.64}$	$99.90_{\pm 0.67}$	$102.01_{\pm 1.05}$	$104.48_{\pm 0.58}$	$105.37_{\pm 0.52}$
	HIWAE (Huang et al. (2019))	$98.75_{\pm 0.79}$	$99.21_{\pm 0.95}$	$101.35_{\pm 1.48}$	$103.72_{\pm 0.94}$	$104.62_{\pm 0.80}$
	IWAE (Ours, CNN-GRU)	$98.98_{\pm 1.01}$	$99.48_{\pm 0.94}$	$99.20_{\pm 0.52}$	$104.19_{\pm 1.11}$	$105.25_{\pm 1.05}$
	CIWAE (Ours, CNN-GRU)	$99.75_{\pm 0.71}$	$100.14_{\pm 0.74}$	$100.43_{\pm 1.36}$	$104.91_{\pm 0.55}$	$105.72_{\pm 0.43}$
	AIWAE (Ours, CNN-GRU)	$\mathbf{98.53_{\pm 0.91}}$	$\mathbf{99.09_{\pm 0.91}}$	$99.24_{\pm 0.70}$	$103.62_{\pm 1.18}$	$104.62_{\pm 1.13}$
	IWAE (Ours, CNN)	$\mathbf{97.76_{\pm 0.45}}$	$\mathbf{98.16_{\pm 0.45}}$	$98.62_{\pm 0.67}$	$102.61_{\pm 0.35}$	$103.70_{\pm 0.39}$
	AIWAE (Ours, CNN)	$97.79_{\pm 0.28}$	$98.46_{\pm 0.35}$	$98.69_{\pm 0.13}$	$102.92_{\pm 0.35}$	$103.88_{\pm 0.46}$
10	IWAE (Huang et al. (2019))	$97.97_{\pm 0.47}$	$98.48_{\pm 0.34}$	$99.71_{\pm 0.95}$	$102.67_{\pm 0.39}$	$103.53_{\pm 0.23}$
	HIWAE (Huang et al. (2019))	$98.48_{\pm 0.67}$	$98.80_{\pm 0.64}$	$99.75_{\pm 1.31}$	$103.21_{\pm 0.77}$	$104.11_{\pm 0.85}$
	IWAE (Ours, CNN-GRU)	$97.14_{\pm 0.29}$	$\mathbf{97.62_{\pm 0.22}}$	$97.59_{\pm 0.36}$	$101.66_{\pm 0.33}$	$102.65_{\pm 0.33}$
	CIWAE (Ours, CNN-GRU)	$97.82_{\pm 0.49}$	$98.30_{\pm 0.44}$	$98.09_{\pm 0.53}$	$102.37_{\pm 0.62}$	$103.36_{\pm 0.51}$
	AIWAE (Ours, CNN-GRU)	$\mathbf{97.13_{\pm 0.36}}$	$97.73_{\pm 0.20}$	$97.60_{\pm 0.27}$	$101.74_{\pm 0.36}$	$102.75_{\pm 0.34}$
	IWAE (Ours, CNN)	$\mathbf{96.16_{\pm 0.19}}$	$\mathbf{96.80_{\pm 0.21}}$	$97.55_{\pm 0.14}$	$100.48_{\pm 0.25}$	$101.51_{\pm 0.35}$
	AIWAE (Ours, CNN)	$96.21_{\pm 0.58}$	$96.82_{\pm 0.56}$	$97.38_{\pm 0.42}$	$100.65_{\pm 0.72}$	$101.73_{\pm 0.66}$

Table 4.2. The results of CIWAE and AIWAE experiments, along with the corresponding IWAE and HIWAE baselines of ours and by by Huang et al. (2019). The experiments are performed on the OMNIGLOT (dynamically binarized) dataset by Lake et al. (2015).

n	**Approach**	NLL$_{val}$	NLL$_{te}$	$\hat{L}_{tr}$	$\hat{L}_{val}$	$\hat{L}_{te}$
1	IWAE (Ours, MLP-GRU)	$91.08_{\pm 0.16}$	$90.20_{\pm 0.13}$	$93.62_{\pm 0.16}$	$97.27_{\pm 0.08}$	$96.46_{\pm 0.05}$
	CIWAE (Ours, MLP-GRU)	$91.08_{\pm 0.16}$	$90.20_{\pm 0.13}$	$93.62_{\pm 0.16}$	$97.27_{\pm 0.08}$	$96.46_{\pm 0.05}$
	AIWAE (Ours, MLP-GRU)	$91.08_{\pm 0.16}$	$90.20_{\pm 0.13}$	$93.62_{\pm 0.16}$	$97.27_{\pm 0.08}$	$96.46_{\pm 0.05}$
	IWAE (Ours, MLP)	$90.77_{\pm 0.12}$	$89.92_{\pm 0.12}$	$93.38_{\pm 0.23}$	$97.20_{\pm 0.09}$	$96.42_{\pm 0.16}$
	AIWAE (Ours, MLP)	$90.77_{\pm 0.12}$	$89.92_{\pm 0.12}$	$93.38_{\pm 0.23}$	$97.20_{\pm 0.09}$	$96.42_{\pm 0.16}$
2	IWAE (Ours, MLP-GRU)	$90.61_{\pm 0.13}$	$89.75_{\pm 0.14}$	$92.64_{\pm 0.17}$	$96.32_{\pm 0.08}$	$95.50_{\pm 0.10}$
	CIWAE (Ours, MLP-GRU)	$\mathbf{90.23_{\pm 0.05}}$	$\mathbf{89.43_{\pm 0.07}}$	$92.10_{\pm 0.15}$	$95.69_{\pm 0.07}$	$94.94_{\pm 0.11}$
	AIWAE (Ours, MLP-GRU)	$90.60_{\pm 0.07}$	$89.76_{\pm 0.07}$	$92.39_{\pm 0.19}$	$95.86_{\pm 0.05}$	$95.04_{\pm 0.06}$
	IWAE (Ours, MLP)	$90.18_{\pm 0.18}$	$\mathbf{89.32_{\pm 0.15}}$	$92.20_{\pm 0.30}$	$95.85_{\pm 0.08}$	$95.05_{\pm 0.06}$
	AIWAE (Ours, MLP)	$\mathbf{90.16_{\pm 0.09}}$	$89.34_{\pm 0.10}$	$91.87_{\pm 0.15}$	$95.51_{\pm 0.07}$	$94.74_{\pm 0.13}$
5	IWAE (Ours, MLP-GRU)	$89.84_{\pm 0.18}$	$89.00_{\pm 0.17}$	$91.30_{\pm 0.28}$	$94.65_{\pm 0.14}$	$93.80_{\pm 0.17}$
	CIWAE (Ours, MLP-GRU)	$\mathbf{89.49_{\pm 0.10}}$	$\mathbf{88.68_{\pm 0.07}}$	$90.64_{\pm 0.17}$	$94.19_{\pm 0.06}$	$93.39_{\pm 0.03}$
	AIWAE (Ours, MLP-GRU)	$89.73_{\pm 0.07}$	$88.89_{\pm 0.05}$	$90.96_{\pm 0.10}$	$94.55_{\pm 0.07}$	$93.74_{\pm 0.05}$
	IWAE (Ours, MLP)	$89.45_{\pm 0.13}$	$88.62_{\pm 0.13}$	$90.70_{\pm 0.21}$	$94.29_{\pm 0.10}$	$93.50_{\pm 0.07}$
	AIWAE (Ours, MLP)	$\mathbf{89.37_{\pm 0.23}}$	$\mathbf{88.57_{\pm 0.19}}$	$90.47_{\pm 0.39}$	$94.17_{\pm 0.12}$	$93.41_{\pm 0.11}$
10	IWAE (Ours, MLP-GRU)	$\mathbf{89.35_{\pm 0.12}}$	$\mathbf{88.52_{\pm 0.08}}$	$90.24_{\pm 0.18}$	$93.68_{\pm 0.10}$	$92.92_{\pm 0.08}$
	CIWAE (Ours, MLP-GRU)	$89.49_{\pm 0.10}$	$88.68_{\pm 0.07}$	$90.64_{\pm 0.17}$	$94.19_{\pm 0.06}$	$93.39_{\pm 0.03}$
	AIWAE (Ours, MLP-GRU)	$89.37_{\pm 0.12}$	$88.56_{\pm 0.11}$	$90.29_{\pm 0.16}$	$93.67_{\pm 0.05}$	$92.87_{\pm 0.07}$
	IWAE (Ours, MLP)	$\mathbf{89.02_{\pm 0.10}}$	$\mathbf{88.19_{\pm 0.10}}$	$89.83_{\pm 0.19}$	$93.40_{\pm 0.05}$	$92.61_{\pm 0.07}$
	AIWAE (Ours, MLP)	$89.04_{\pm 0.23}$	$88.23_{\pm 0.23}$	$89.80_{\pm 0.40}$	$93.38_{\pm 0.11}$	$92.65_{\pm 0.14}$

Table 4.3. The results of CIWAE and AIWAE experiments, along with the corresponding IWAE of ours. The experiments are performed on the MNIST (statically binarized) dataset by Larochelle and Murray (2011).

n	**Approach**	NLL_{val}	NLL_{te}	$\hat{L}_{tr}$	$\hat{L}_{val}$	$\hat{L}_{te}$
1	IWAE (Ours, MLP-GRU)	$113.40_{\pm 0.72}$	$113.66_{\pm 0.75}$	$117.13_{\pm 0.72}$	$118.11_{\pm 0.72}$	$118.98_{\pm 0.72}$
	CIWAE (Ours, MLP-GRU)	$113.40_{\pm 0.72}$	$113.66_{\pm 0.75}$	$117.13_{\pm 0.72}$	$118.11_{\pm 0.72}$	$118.98_{\pm 0.72}$
	AIWAE (Ours, MLP-GRU)	$113.40_{\pm 0.72}$	$113.66_{\pm 0.75}$	$117.13_{\pm 0.72}$	$118.11_{\pm 0.72}$	$118.98_{\pm 0.72}$
	IWAE (Ours, MLP)	$110.58_{\pm 0.24}$	$110.71_{\pm 0.15}$	$114.75_{\pm 0.21}$	$115.95_{\pm 0.18}$	$116.65_{\pm 0.16}$
	AIWAE (Ours, MLP)	$110.58_{\pm 0.24}$	$110.71_{\pm 0.15}$	$114.75_{\pm 0.21}$	$115.95_{\pm 0.18}$	$116.65_{\pm 0.16}$
2	IWAE (Ours, MLP-GRU)	$111.29_{\pm 0.82}$	$111.80_{\pm 0.63}$	$114.92_{\pm 0.71}$	$115.98_{\pm 0.54}$	$116.81_{\pm 0.54}$
	CIWAE (Ours, MLP-GRU)	$\mathbf{110.55_{\pm 0.25}}$	$\mathbf{110.85_{\pm 0.29}}$	$114.08_{\pm 0.17}$	$115.18_{\pm 0.10}$	$115.86_{\pm 0.23}$
	AIWAE (Ours, MLP-GRU)	$111.18_{\pm 0.53}$	$111.53_{\pm 0.55}$	$114.61_{\pm 0.72}$	$115.63_{\pm 0.50}$	$116.39_{\pm 0.50}$
	IWAE (Ours, MLP)	$\mathbf{109.37_{\pm 0.36}}$	$\mathbf{109.58_{\pm 0.36}}$	$113.07_{\pm 0.29}$	$114.25_{\pm 0.28}$	$114.89_{\pm 0.32}$
	AIWAE (Ours, MLP)	$110.39_{\pm 0.60}$	$110.74_{\pm 0.56}$	$113.88_{\pm 0.65}$	$115.09_{\pm 0.48}$	$115.64_{\pm 0.51}$
5	IWAE (Ours, MLP-GRU)	$110.33_{\pm 0.52}$	$110.44_{\pm 0.53}$	$113.18_{\pm 0.57}$	$114.17_{\pm 0.46}$	$114.93_{\pm 0.42}$
	CIWAE (Ours, MLP-GRU)	$\mathbf{109.86_{\pm 0.64}}$	$\mathbf{110.00_{\pm 0.61}}$	$112.69_{\pm 0.69}$	$113.78_{\pm 0.59}$	$114.38_{\pm 0.50}$
	AIWAE (Ours, MLP-GRU)	$110.01_{\pm 0.68}$	$110.33_{\pm 0.63}$	$112.99_{\pm 0.74}$	$114.04_{\pm 0.53}$	$114.73_{\pm 0.51}$
	IWAE (Ours, MLP)	$\mathbf{108.45_{\pm 0.41}}$	$\mathbf{108.78_{\pm 0.55}}$	$111.63_{\pm 0.49}$	$112.64_{\pm 0.42}$	$113.33_{\pm 0.46}$
	AIWAE (Ours, MLP)	$109.68_{\pm 0.64}$	$109.98_{\pm 0.68}$	$112.58_{\pm 0.72}$	$113.55_{\pm 0.56}$	$114.27_{\pm 0.57}$
10	IWAE (Ours, MLP-GRU)	$\mathbf{108.62_{\pm 0.49}}$	$\mathbf{109.02_{\pm 0.32}}$	$111.37_{\pm 0.43}$	$112.54_{\pm 0.32}$	$113.23_{\pm 0.26}$
	CIWAE (Ours, MLP-GRU)	$108.93_{\pm 0.44}$	$109.34_{\pm 0.47}$	$111.70_{\pm 0.45}$	$112.66_{\pm 0.39}$	$113.33_{\pm 0.32}$
	AIWAE (Ours, MLP-GRU)	$108.92_{\pm 0.50}$	$109.25_{\pm 0.43}$	$111.55_{\pm 0.42}$	$112.66_{\pm 0.28}$	$113.34_{\pm 0.30}$
	IWAE (Ours, MLP)	$\mathbf{107.85_{\pm 0.99}}$	$\mathbf{108.10_{\pm 0.89}}$	$110.57_{\pm 0.85}$	$111.63_{\pm 0.81}$	$112.24_{\pm 0.72}$
	AIWAE (Ours, MLP)	$108.69_{\pm 0.50}$	$108.98_{\pm 0.46}$	$111.26_{\pm 0.54}$	$112.37_{\pm 0.41}$	$113.01_{\pm 0.42}$

Table 4.4. The results of CIWAE and AIWAE experiments, along with the corresponding IWAE and HIWAE baselines of ours and by by Huang et al. (2019). The experiments are performed on the OMNIGLOT (dynamically binarized) dataset by Lake et al. (2015).

		MNIST	**OMNIGLOT**
n	**Approach**	NLL_{te}	NLL_{te}
1	IWAE (Burda et al. (2015), MLP)	88.71	108.11
	IWAE (Burda et al. (2015), MLP-2)	**88.08**	**107.58**
	CIWAE (Ours, MLP-GRU)	90.11	109.96
	AIWAE (Ours, MLP)	89.75	108.12
5	IWAE (Burda et al. (2015), MLP)	88.83	107.62
	IWAE (Burda et al. (2015), MLP-2)	**87.63**	106.31
	CIWAE (Ours, MLP-GRU)	88.37	107.04
	AIWAE (Ours, MLP)	88.34	**105.79**
50	IWAE (Burda et al. (2015), MLP)	89.05	107.80
	IWAE (Burda et al. (2015), MLP-2)	87.86	106.30
	CIWAE (Ours, MLP-GRU)	**87.43**	107.02
	AIWAE (Ours, MLP)	87.45	**104.16**

Table 4.5. The results of CIWAE and AIWAE experiments, along with the corresponding IWAE baselines of ours and by Burda et al. (2015). The experiments are performed on both the MNIST (statically binarized) dataset by Larochelle and Murray (2011) and the OMNIGLOT (dynamically binarized) dataset by Lake et al. (2015).

Chapter 5

Discussion

5.1. Conclusions and Future Work

In this Section, we discuss the contributions of our work, the conclusions of the experimentation, and describe some directions for future work.

In Chapter 3, we discussed the theoretical properties of multi-sample estimators and their variational gaps. In Theorem 3.3.1, we developed a simple upper bound on the variational gap of an estimator in terms of its variance under certain amicable conditions on the estimator. One of the conditions requires bounded variance and the other condition is the strong lower-bounded estimator assumption. In Section 3.6, we provided justifications for the lower-bounded estimator and argued that despite being strong, it is also reasonable. Thus, with the upper bound, we reduced the problem of improving the variational inference to the design of multi-sample low-variance estimators. We observed that there are mechanisms to design multi-sample estimators such that their variance can be made to vanish at the rate of $\mathcal{O}\left(\frac{1}{n}\right)$. Based on this, we defined the family of variationally-asymptotic Monte-Carlo estimators, for which the variational gap goes to 0 at the rate of $\mathcal{O}\left(\frac{1}{n}\right)$.

However, having reduced the problem of better variational inference to multi-sample low-variance estimators, the next natural step is to consider the sample efficiency; we want to improve the variational inference with as few samples as possible. We argued that sampling the latent variables conditionally, where each new sample is generated conditioned on either all or some of the previous samples, can lead to sample efficiency. Towards this, in Chapter 4, we considered two approaches: **1.** Conditional-IWAE (CIWAE) and **2.** Antithetic-IWAE (which is originally considered in Klys et al. (2018) as well). In CIWAE, each new sample is conditioned on all the previous samples and the IWAE estimator is corrected to the CIWAE estimator. As proved in Section 4.3, the corrected CIWAE estimators result in making CIWAE an instance of variationally asymptotic Monte-Carlo estimator; its variational gap can be proved to vanish at the rate of $\mathcal{O}\left(\frac{1}{n}\right)$. In AIWAE, we showed that it is not necessary to make all the individual estimators uncorrelated; we can generate samples in an antithetic manner so that sufficiently many IWAE estimators become uncorrelated. Thus, as proved in Section 4.5, AIWAE is also an instance of variationally asymptotic Monte-Carlo estimator with variational gap of $\mathcal{O}\left(\frac{1}{n}\right)$. In Section 4.7, we performed experiments on MNIST and OMNIGLOT datasets and compare the performances of CIWAE and AIWAE as well as the IWAE and HIWAE baselines of Huang et al. (2019) and Burda et al. (2015). We showed that as the number of samples increases, CIWAE and AIWAE become competitive and even outperform the baselines. With this, we showed the utility of modeling the joint distribution over latent variables in improving the multi-sample variational inference.

However, we could not theoretically prove or disprove whether CIWAE and AIWAE can be made to perform better than the IWAE baseline; we only demonstrated their variationally asymptotic nature and that they perform better empirically. This is one of the directions for future work; we can try to characterize how modeling joint distribution over latent variables can improve upon the i.i.d. sampling of IWAE. We also observed that the choice of architecture is important in experimentation with all the approaches. Note that CIWAE requires a proposal encoder of the form $q_\phi\left(\mathbf{z}^i \mid \mathbf{z}^{1:i-1}, \mathbf{x}\right)$, which can process variable number of samples $\mathbf{z}^{1:i}$. Thus, in our experimentation, we used GRU cell (Cho et al. (2014)) for modeling this proposal. However, we observed that modeling this distribution with a large number of samples does not perform well as GRUs are known to have problems in modeling long sequences. Another direction of work could be the use of transformer networks (Vaswani et al. (2017)), or its derivatives, to better model the involved conditional sampling as transformer networks are shown to better mitigate the problem of learning long-term dependencies. In addition, a recent work called SUMO by Luo et al. (2020) has demonstrated an unbiased estimator of the log marginal data likelihood $\log p\left(\mathbf{x}\right)$, effectively closing the variational gap. Thus, further research might be oriented towards variance reduction in such unbiased estimators of $\log p\left(\mathbf{x}\right)$.

References

[1] Robert A Becker. The variance drain and jensen's inequality. 2012.

[2] Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[3] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[4] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[5] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[6] Augustin Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

[7] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[9] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

[10] Chris Cremer, Quaid Morris, and David Duvenaud. Reinterpreting importance-weighted autoencoders. *arXiv preprint arXiv:1704.02916*, 2017.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] Justin Domke and Daniel R Sheldon. Importance weighting and variational inference. In *Advances in neural information processing systems*, pages 4470–4479, 2018.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[15] Chin-Wei Huang and Aaron Courville. Note on the bias and variance of variational inference. *arXiv preprint arXiv:1906.03708*, 2019.

[16] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. *arXiv preprint arXiv:1804.00779*, 2018.

[17] Chin-Wei Huang, Kris Sankaran, Eeshan Dhekane, Alexandre Lacoste, and Aaron Courville. Hierarchical importance weighted autoencoders. In *International Conference on Machine Learning*, pages 2869–2878, 2019.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Jack Klys, Jesse Bettencourt, and David Duvenaud. Joint importance sampling for variational inference. 2018.

[20] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[21] Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 29–37, 2011.

[22] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2:18, 2010.

[23] Yucen Luo, Alex Beatson, Mohammad Norouzi, Jun Zhu, David Duvenaud, Ryan P Adams, and Ricky TQ Chen. Sumo: Unbiased estimation of log marginal probability for latent variable models. *arXiv preprint arXiv:2004.00353*, 2020.

[24] Chris J Maddison, John Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Teh. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pages 6573–6583, 2017.

[25] Dmitry Molchanov, Valery Kharitonov, Artem Sobolev, and Dmitry Vetrov. Doubly semi-implicit variational inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2593–2602, 2019.

[26] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *International Conference on Artificial Intelligence and Statistics*, pages 968–977, 2018.

[27] Sebastian Nowozin. Debiasing evidence approximations: On importance-weighted autoencoders and jackknife variational inference. In *International Conference on Learning Representations*, 2018.

[28] Art B Owen. Monte carlo theory. *Methods and Examples*, 665, 2013.

[29] 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. *arXiv preprint arXiv:1802.04537*, 2018.

[30] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

[31] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[32] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.

[33] Artem Sobolev and Dmitry P Vetrov. Importance weighted hierarchical variational inference. In *Advances in Neural Information Processing Systems*, pages 603–615, 2019.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[35] George Tucker, Dieterich Lawson, Shixiang Gu, and Chris J Maddison. Doubly reparameterized gradient estimators for monte carlo objectives. *arXiv preprint arXiv:1810.04152*, 2018.

[36] A. M. Turing. Computing machinery and intelligence. *Mind*, LIX(236):433–460, 10 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL `https://doi.org/10.1093/mind/LIX.236.433`.

[37] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[39] James R Wilson. Antithetic sampling with multivariate inputs. *American Journal of Mathematical and Management Sciences*, 3(2):121–144, 1983.

[40] Mike Wu, Noah Goodman, and Stefano Ermon. Differentiable antithetic sampling for variance reduction in stochastic variational inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2877–2886, 2019.

[41] Mingzhang Yin and Mingyuan Zhou. Semi-implicit variational inference. *arXiv preprint arXiv:1805.11183*, 2018.