

# Amortized Variational Inference: A Systematic Review

**Ankush Ganguly**  
**Sanjana Jain**  
**Ukrit Watchareeruetai**  
*Sertis Vision Lab*  
*Sukhumvit Road, Watthana, Bangkok 10110, Thailand*

AGANG@SERTISCORP.COM  
SJAIN@SERTISCORP.COM  
UWATC@SERTISCORP.COM

## Abstract

The core principle of Variational Inference (VI) is to convert the statistical inference problem of computing complex posterior probability densities into a tractable optimization problem. This property enables VI to be faster than several sampling-based techniques. However, the traditional VI algorithm is not scalable to large data sets and is unable to readily infer out-of-bounds data points without re-running the optimization process. Recent developments in the field, like stochastic-, black box-, and amortized-VI, have helped address these issues. Generative modeling tasks nowadays widely make use of amortized VI for its efficiency and scalability, as it utilizes a parameterized function to learn the approximate posterior density parameters. In this paper, we review the mathematical foundations of various VI techniques to form the basis for understanding amortized VI. Additionally, we provide an overview of the recent trends that address several issues of amortized VI, such as the amortization gap, generalization issues, inconsistent representation learning, and posterior collapse. Finally, we analyze alternate divergence measures that improve VI optimization.

## 1. Introduction

Bayesian inference is an indispensable part of machine learning as it allows for systematic reasoning about parameter uncertainty (Zhang et al., 2019). Bayesian statistics' core principle is to frame all inference about unknown variables as a calculation involving a posterior probability density (Blei et al., 2017). Exact inference, which typically involves analytically computing the posterior probability distribution over the variables of interest, offers a solution to this inference problem. Algorithms in this category include the elimination algorithm (Gagliardi Cozman, 2000), the sum-product algorithm (Kschischang et al., 2001), and the junction tree algorithm (Madsen & Jensen, 1999). For highly complex probability densities, however, exact inference does not guarantee a closed-form solution. In fact, the exact computation of conditional probabilities in belief networks is NP-hard (Dagum & Luby, 1993).

As an alternative to exact inference, approximate inference techniques, which have been in development since the early 1950s, offer an efficient solution to Bayesian inference by providing simpler estimates of complex probability densities. Approximate inference provides solutions to even non-conjugate<sup>1</sup> models for which analytic posteriors are unavailable (Knollmüller & Enßlin, 2019). Markov Chain Monte Carlo (MCMC) methods such as the

---

1. Conjugacy occurs when the posterior density is in the same family of probability density functions as the prior, but with new parameter values which have been updated to reflect the learning from the data.

Metropolis-Hastings algorithm (Metropolis et al., 1953) and Gibbs sampling (Geman & Geman, 1984) fall under this category. However, MCMC methods that rely on sampling (Brooks et al., 2011; Gershman et al., 2012; Robbins & Monro, 1951) are slow to converge and do not scale efficiently.

Variational Inference (VI), a method in machine learning, tackles the problem of inefficient approximate inference by the use of a suitable metric to select a tractable approximation to the posterior probability density. The methodology of VI is, thus, to re-frame the statistical inference problem into an optimization problem giving us the speed benefits of maximum a posteriori (MAP) estimation (Murphy, 2013) and the ability to scale to large data sets (Blei et al., 2017). This makes VI an ideal choice for application in areas like statistical physics (Regier et al., 2015; Smith et al., 2021; Marino & Manic, 2021), diagnostic inference in Quick Medical Reference (QMR) networks (Jaakkola & Jordan, 1999), generative modeling (e.g., Kingma & Welling, 2014; Larsen et al., 2016; Zhao et al., 2019; Higgins et al., 2017; Burgess et al., 2018), and neural networks (e.g., Sun et al., 2019; Shen et al., 2020; Haußmann et al., 2020; Eikema & Aziz, 2019). Other than VI, loopy-belief propagation (Murphy et al., 1999) and expectation maximization (Minka, 2001) also fall within the class of optimization-based inference techniques. We re-iterate that both MCMC methods and VI solve the problem of inference, but their respective approaches are different. While MCMC algorithms rely on sampling to approximate the posterior, VI uses optimization for the approximation.

Since its inception, researchers have developed the traditional VI algorithm (introduced by Jordan et al., 1999) to make it more accurate, efficient, and scalable. The traditional VI algorithm operates by introducing a new set of parameters, characterizing the approximated density, for every observation with the aim to find unbiased estimates for the parameters of the true posterior probability density. This leads to inefficient scalability as these optimizable parameters grow linearly with the observations. On the other hand, *amortized inference*, an improvement over the traditional VI algorithm, uses a stochastic function to estimate the posterior probability density (Zhang et al., 2019). Unlike traditional VI, the parameters of this stochastic function are fixed and shared across all data points, thereby amortizing the inference<sup>2</sup>. Deep neural networks are a popular choice for this stochastic function as they combine probabilistic modeling with the representational power of deep learning (Zhang et al., 2019). Thus, amortized inference combined with deep neural networks has been shown to efficiently scale to large data sets. The variational auto-encoder (VAE) (Kingma & Welling, 2014; Rezende et al., 2014) and its variants are primary examples in this case. Not only does amortizing the inference aid in scalability, but the memoized re-use (Gershman & Goodman, 2014) of the learned parameters of the stochastic function helps test inference on new observations without having to re-run the optimization process, like in the case of traditional VI.

In this paper, we study and provide an intuitive explanation of the different VI techniques and their applications to researchers new to the field, and elucidate the strengths and weaknesses of these methods. In addition, this paper builds off of the mathematical foundations of traditional-, stochastic-, and black box-VI to form the basis of explanation for amortized VI, its properties, and caveats. To the best of our knowledge, while several

---

2. Section 5 explains in detail the intent of the use of this phrase in the context of probabilistic modeling.

excellent reviews of VI exist (e.g., Blei et al., 2017; Zhang et al., 2019), this is the first review paper dedicated to gaining a deeper understanding of the concept of amortized VI while distinguishing it from several other forms of VI. In addition, we unify the mathematical notations from many research papers to ease the readers in understanding the concepts, features, and differences of each VI methodology. Furthermore, we study how the recent developments in the field of amortized VI have addressed its weaknesses, discuss alternate divergence measures, and analyze their effect on VI optimization.

We organize the paper as follows: Section 2 discusses the relevance of statistical inference in real-world problems as well as revisits the core concepts of VI such as the VI optimization problem, the Evidence Lower Bound (ELBO), mean field VI, and the coordinate ascent VI (CAVI) optimization algorithm. Section 3 explains how stochastic VI uses stochastic optimization and natural gradients to make VI a scalable method. Section 4 elucidates the concept of black box VI and the reparameterization trick. Section 5 dives into a deeper understanding of amortized inference, addresses the issues associated with it, and provides an overview of the various advancements. Section 6 analyzes the use of different divergence measures that improve optimization in VI. Finally, we conclude with an overview of active research areas and open problems in Section 7.

## 2. Variational Inference

This section explores the importance of statistical inference in practical problems and reviews core VI concepts like the VI optimization problem, ELBO, mean field VI, and the CAVI optimization algorithm.

### 2.1 Statistical Inference in Real-world Problems

Statistical inference has been applied to solve many real-world problems. As an example, let’s consider the problem of topic modeling (Blei et al., 2003; Hoffman et al., 2010, 2013; Blei, 2012) that aims to uncover hidden thematic structures, i.e., topics, within each document in a corpus. A topic in a document refers to a subject that represents a meaningful pattern of words found in the document. It represents a set of words that are frequently associated with each other within a specific context. A document can be considered as a mixture of topics, where a distribution over words characterizes each topic. In topic modeling, the words in the documents are called observable variables since they can be directly observed from the documents, while topics, the underlying variables influencing the distributions of words in documents, are called latent variables. These latent variables can be inferred using statistical inference.

Another example of a real-world problem is a recommender system, which aims to predict the preferences of a user on a particular product. Collaborative filtering is one of the major approaches for recommender systems. In collaborative filtering, a user-item interaction matrix, in which rows and columns represent users and items, respectively, and each element in the matrix denotes the preference, e.g., rating score, of a particular user on a particular item. This matrix is, however, sparse and incomplete; the values of most elements are unknown. Collaborative filtering aims to fill in this incomplete matrix by assuming that there is a group of latent variables, called a latent vector, affecting the observed values in each row and each column. Once latent vectors of all users and items are obtained, the

Notation	Description
$x = [x_1, \dots, x_N]$	Observed variable
$z = [z_1, \dots, z_N]$	Latent variable
$N$	Total number of data points
$\theta$	Global generative model parameters
$\phi$	Global variational (or recognition model) parameters
$\xi_i$	Local variational parameter for the $i$ -th data point
$p(z x_i; \theta)$	True posterior probability density conditioned on $\theta$
$q(z x_i; \xi_i)$	Variational approximation parameterized by $\xi_i$
$\mathcal{Q}$	Family of tractable probability densities
$D_{\text{KL}}(q(z x; \xi) \parallel p(z x; \theta))$	Kullback-Leibler divergence
$\mathcal{D}$	The total KL-divergence objective function
$M$	Mini-batch size for stochastic optimization
$x^M$	Mini-batch containing $M$ data points
$\xi^M$	Set of the local variational parameters for $M$ samples
$\mathcal{M}$	The set of all the subsets of the data set split into $\frac{N}{M}$ subsets
$\mathcal{L}$	Evidence Lower Bound (ELBO)
$\hat{\mathcal{L}}$	Stochastic estimate of the ELBO
$\tilde{\mathcal{L}}$	Stochastic estimate of the ELBO using Monte Carlo samples
$\tilde{\mathcal{L}}^B$	SGVB estimator for VAE (Kingma & Welling, 2014)
$K$	Monte Carlo samples used in ELBO approximation
$\nabla$	Stochastic gradients of a function
$\hat{\nabla}$	Mean stochastic gradients of a function based on $M$ samples
$\bar{\nabla}$	The natural gradients of a function
$I$	The Fisher Information Matrix

Table 1: Notations used throughout this paper.

user-item interaction matrix can be simply computed using matrix multiplication. Similarly, statistical inference is used to obtain the parameters of a generative process to produce these latent variables (Li & She, 2017; Liang, Krishnan, Hoffman, & Jebara, 2018).

Graphical model representation (Airoldi, 2007; Jordan, 2004; Bishop, 2006a) is a tool allowing us to formulate a real-world problem as a graph that can be solved by statistical inference. A graphical model is a probabilistic model that expresses conditional dependence structure between random variables using a graph. Figure 1(a) is an example of a graphical model. In a graphical model, a node can represent either a random variable or a deterministic parameter. A random variable is represented by a circle while a deterministic parameter is denoted by just a symbol. For example, as shown in the figure, there are two groups of random variables, i.e.,  $x_i$  and  $z_i$  where  $i = 1, \dots, N$ , and a deterministic parameter  $\theta$ . An edge, i.e., a link between two nodes, denotes the conditional dependence between them; for example, the edge from  $\theta$  to  $z_i$  indicates that each random variable  $z_i$  is conditioned on  $\theta$ . Moreover, a graphical model allows one to differentiate observed variables from latent variables by shading the corresponding nodes. From the figure,  $x_i$  are observed variables while

$z_i$  are latent variables. When a node is enclosed by a plate (denoted by the rounded-corner rectangle in Figure 1(b)), it indicates that there is a group of nodes of the same kind.

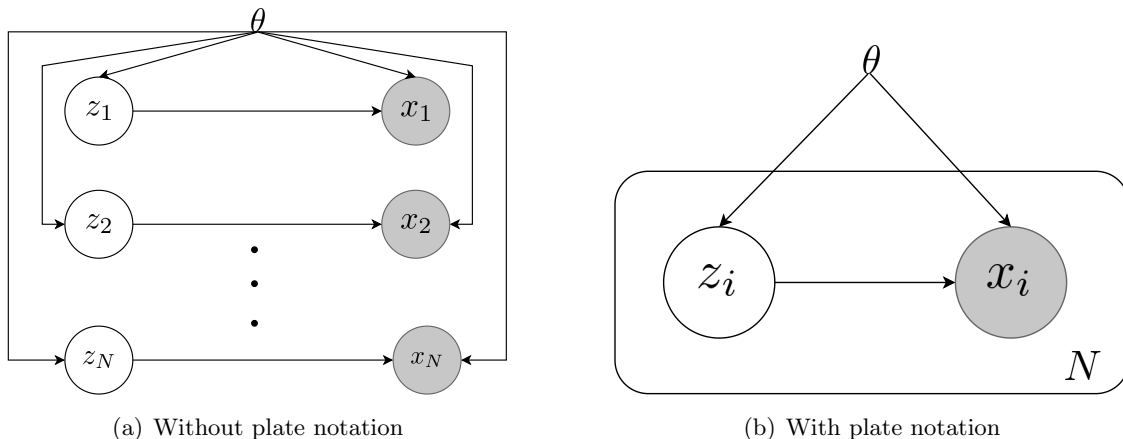


Figure 1: A directed graphical model with  $N$  data points describing joint distributions over  $x_i$  and  $z_i$  as follows:  $p(x_i, z_i; \theta) = p(x_i|z_i; \theta)p(z_i; \theta)$ .

Our assumption is that the observed data points are independent and identically distributed (i.i.d.) and are generated by some random process, involving the unobserved random variable  $z$ . For each data point  $x_i$ , there exists a latent vector  $z_i$ , which is assumed to have some prior probability density  $p(z; \theta)$ . Additionally, the data points are sampled from the conditional probability density  $p(x|z; \theta)$ . Computing the posterior probability density,  $p(z|x; \theta)$ , is useful for a variety of tasks such as coding or data representation, denoising, recognition, and visualization (Kingma & Welling, 2014).

Although a variety of generative processes with more complex directed graphical models are possible, we restrict ourselves to the common case where a latent variable is associated with each i.i.d. observed data point. For example, in our case, the observed data points can be pictured as images while the latent variables as lower dimensional representations of those images. From a coding theory perspective, these latent variables are interpreted as *code* (Kingma & Welling, 2014) and thus form the basis of representation learning.

We use Bayes' theorem to compute the posterior probability density as:

$$p(z|x; \theta) = \frac{p(x|z; \theta)p(z; \theta)}{p(x; \theta)},$$

$$p(x; \theta) = \int p(x|z; \theta)p(z; \theta)dz. \quad (1)$$

The marginal probability density,  $p(x; \theta)$ , in Equation 1 is called the *evidence*, which is high dimensional for most statistical models, and its computation is thus, at times, intractable or of exponential complexity. This computation is significant as a higher marginal likelihood indicates the chosen model's ability to fit the observed data better.

The purpose of VI is, therefore, two-fold:

1. Analytical approximation of the posterior probability density for statistical inference over the latent variables.

2. Provide an alternative to tractably compute the evidence to encourage a better fit to the data by the chosen statistical model.

## 2.2 Statistical Inference as Optimization

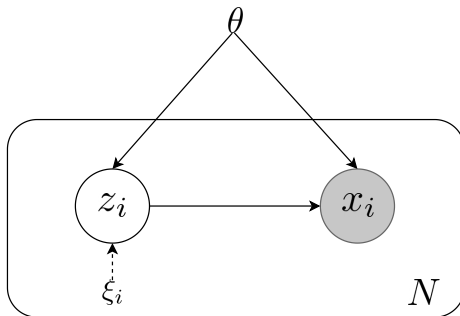


Figure 2: A directed graphical model with  $N$  data points. Solid lines denote the generative model, while dashed lines denote the variational approximation to the intractable posterior density (Kingma & Welling, 2014). The local variational parameters and the global generative model parameters are represented by  $\xi_i$  and  $\theta$ , respectively.

The central idea of VI is to provide simpler approximations to complex posterior probability densities. Traditionally, for each data point  $x_i$ , VI aims to select the approximate density,  $q(z|x_i; \xi_i)$ , from a family of tractable densities  $\mathcal{Q}$  (Figure 2). Each  $q(z|x_i; \xi_i) \in \mathcal{Q}$  is designated by a set of their own variational parameters,  $\xi_i$ , and is a candidate approximation of the actual posterior evaluated at data point  $x_i$ . The goal is to tune these parameters to get an optimal approximation of the actual posterior density. Thus, VI converts this Bayesian inference problem into an optimization problem. The complexity and the accuracy of this optimization are controlled by the choice of the variational family (Plummer et al., 2020). This choice, further, depends on a measure that captures the difference between the approximated posterior and the true posterior density (Ranganath et al., 2014). Usually, this measure is chosen to be the non-negative Kullback–Leibler (KL) divergence which estimates the relative entropy between two densities (Bishop, 2006b; Kullback & Leibler, 1951; Jordan et al., 1999). In the case of VI, it quantifies the relative entropy between the true posterior probability density,  $p(z|x_i; \theta)$ , and the candidate density,  $q(z|x_i; \xi_i)$ . The optimization problem for traditional VI entails reducing the relative entropy by choosing the approximate density with the lowest reverse KL-divergence to the true posterior density, sampling one data point at a time (Blei et al., 2017; Ganguly & Earp, 2021). The objective function for this process can be formulated as:

$$\mathcal{D} = \sum_{i=1}^N D_{\text{KL}}(q(z|x_i; \xi_i) \parallel p(z|x_i; \theta)) = \sum_{i=1}^N \mathbb{E}_q \left[ \log \frac{q(z|x_i; \xi_i)}{p(z|x_i; \theta)} \right], \quad (2)$$

where  $\mathbb{E}_q \left[ \cdot \right] = \mathbb{E}_{q(z|x_i; \xi_i)} \left[ \cdot \right]$  and the output of this optimization process is the set of variational parameters that characterize the best approximation to the true posterior density.

Thus, for each local variational parameter  $\xi_i$ , inference amounts to solving the following optimization problem,

$$q^*(z|x_i; \xi_i) = \arg \min_{q(z|x_i; \xi_i) \in \mathcal{Q}} D_{\text{KL}}(q(z|x_i; \xi_i) \parallel p(z|x_i; \theta)) \quad (3)$$

The forward KL-divergence,  $D_{\text{KL}}(p(z|x_i; \theta) \parallel q(z|x_i; \xi_i))$ , can also be used as a measure in the objective function in Equation 2 as opposed to the defined reverse KL-divergence, given its asymmetric nature. However, in the case of VI, the forward KL-divergence cannot be computed in closed form as it requires taking expectations with respect to the unknown posterior. For readers interested in understanding the foundational difference between forward and reverse KL-divergence, refer to Murphy (2013).

Throughout this paper, we treat the global generative model parameter as a learnable parameter that is learned jointly with the variational parameters during the optimization process. This is to ensure that the narrative of this paper is in line with the recent developments in the field of generative modeling, particularly relating to popular frameworks such as VAE (Kingma & Welling, 2014; Rezende & Mohamed, 2015) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). While being two separate generative modeling frameworks, Su (2018) proves that GANs, like VAEs, are a special case of VI and proposes a unified framework between the two by re-formulating the VI objective.

### 2.3 Evidence Lower Bound

In addition to approximating the intractable posterior probability density for statistical inference over the latent variables, VI also enables efficient computation of a lower bound to the marginal likelihood or the evidence (Ganguly & Earp, 2021). From the perspective of data modeling, a better fit to the observed data by a statistical model requires a better estimate of the evidence by that model. Thus, it indicates that the chosen statistical model generating data points has a greater chance of being from the true data distribution. Furthermore, this computation of the lower bound offers an alternative to the optimization problem defined in Equation 3 which is non-computable as it requires computing the evidence  $\log p(x_i)$  at each data point.

The KL-divergence objective function, for traditional VI, defined in Equation 2 can be expanded as:

$$\begin{aligned} \mathcal{D} &= \sum_{i=1}^N D_{\text{KL}}(q(z|x_i; \xi_i) \parallel p(z|x_i; \theta)), \\ &= \sum_{i=1}^N \mathbb{E}_q \left[ \log \frac{q(z|x_i; \xi_i)}{p(z|x_i; \theta)} \right], \\ &= \sum_{i=1}^N \mathbb{E}_q \left[ \log q(z|x_i; \xi_i) - \log p(z|x_i; \theta) \right], \\ &= \sum_{i=1}^N \mathbb{E}_q \left[ \log q(z|x_i; \xi_i) - \log \frac{p(x_i, z; \theta)}{p(x_i; \theta)} \right], \end{aligned}$$

$$= \sum_{i=1}^N \mathbb{E}_q \left[ \log q(z|x_i; \xi_i) - \log p(x_i, z; \theta) \right] + \sum_{i=1}^N \mathbb{E}_q \left[ \log p(x_i; \theta) \right]. \quad (4)$$

As the marginal likelihood is composed of a sum over the log marginal likelihoods of  $N$  i.i.d. observations (Kingma & Welling, 2014), i.e.,

$$\sum_i^N \log p(x_i; \theta) = \log p(x_1, \dots, x_N; \theta) = \log p(x; \theta),$$

we re-write Equation 4 as:

$$\begin{aligned} \mathcal{D} &= \sum_{i=1}^N \mathbb{E}_q \left[ \log q(z|x_i; \xi_i) - \log p(x_i, z; \theta) \right] + \log p(x; \theta), \\ -\mathcal{D} + \log p(x; \theta) &= \sum_{i=1}^N \mathbb{E}_q \left[ \log p(x_i, z; \theta) - \log q(z|x_i; \xi_i) \right]. \end{aligned} \quad (5)$$

The sum of the negative KL-divergence and the log evidence in Equation 5 is referred to as the *Evidence Lower Bound* (ELBO). Equation 5 indicates that maximizing the ELBO with respect to  $q$  is equivalent to minimizing the objective function defined in Equation 2. In the case of traditional VI, we denote the ELBO by  $\mathcal{L}(x; \xi_{1:N}, \theta)$  as follows:

$$\mathcal{L}(x; \xi_{1:N}, \theta) = \sum_{i=1}^N \mathbb{E}_q \left[ \log p(x_i, z; \theta) - \log q(z|x_i; \xi_i) \right] = \sum_{i=1}^N \mathcal{L}(x_i; \xi_i, \theta). \quad (6)$$

Furthermore, the ELBO, as evident from its name, is a lower bound estimate on the log marginal probability density of the data. This can be derived using Jensen's inequality (Klaričić Bakula et al., 2008) as:

$$\begin{aligned} \log p(x; \theta) &= \sum_{i=1}^N \log \int p(x_i, z; \theta) dz, \\ &= \sum_{i=1}^N \log \int p(x_i, z; \theta) \frac{q(z|x_i; \xi_i)}{q(z|x_i; \xi_i)} dz, \\ &= \sum_{i=1}^N \log \mathbb{E}_q \left[ \frac{p(x_i, z; \theta)}{q(z|x_i; \xi_i)} \right], \\ &\geq \sum_{i=1}^N \mathbb{E}_q \left[ \log p(x_i, z; \theta) - \log q(z|x_i; \xi_i) \right], \\ \log p(x; \theta) &\geq \mathcal{L}(x; \xi_{1:N}, \theta). \end{aligned} \quad (7)$$

An alternative way to show that  $\mathcal{L}(x; \xi_{1:N}, \theta)$  is the lower bound of evidence is from Equations 5 and 6:

$$\mathcal{L}(x; \xi_{1:N}, \theta) = \log p(x; \theta) - \mathcal{D},$$



$$\log p(x; \theta) = \mathcal{L}(x; \xi_{1:N}, \theta) + \mathcal{D}.$$

Since, the KL-divergence is non-negative, so  $\log p(x; \theta) \geq \mathcal{L}(x; \xi_{1:N}, \theta)$ . This relationship establishes ELBO as a lower bound estimate of the incomplete log-likelihood of the data and, thus, at times, is used as a basis for selecting models to fit the data distribution.

## 2.4 Mean Field Variational Family

As computing the ELBO in Equation 6 requires taking expectations with respect to  $q$ , most applications using VI restrict the family of distributions,  $\mathcal{Q}$ , to be from the exponential family due to their conjugate nature leading to ease of computation (Wainwright & Jordan, 2007). An alternative way to ease this computation is to partition the elements of the latent vector  $z$  into disjoint groups denoted by  $z_k$  where  $k = 1, \dots, N$ . Thus, assuming the variational posterior to be factorized as:

$$q(z|x_i; \xi_i) = \prod_{k=1}^N q_k(z_k|x_i; \xi_i). \quad (8)$$

This factorized form of VI corresponds to a framework developed in physics called mean field theory (Parisi & Shankar, 1988) and is known as mean field VI (Opper & Saad, 2001). In the context of belief networks, the mean field theory was further developed by Bhattacharyya and Keerthi (2001). It is to be noted that each of these variational factors can take on any parametric form appropriate to the corresponding random variable (Blei et al., 2017). The ELBO is maximized with respect to each of these factors in Equation 8 which on substitution into Equation 6 and denoting  $q_k(z_k|x_i; \xi_i)$  as  $q_k$  for notational clarity, we obtain,

$$\begin{aligned} \mathcal{L}(x; \xi_{1:N}, \theta) &= \sum_{i=1}^N \mathbb{E}_q \left[ \log p(x_i, z; \theta) - \log q(z|x_i; \xi_i) \right], \\ &= \sum_{i=1}^N \int \prod_k q_k \left[ \log p(x_i, z; \theta) - \log \prod_k q_k \right] dz, \\ &= \sum_{i=1}^N \left\{ \int q_j \left[ \int \log p(x_i, z; \theta) \prod_{k \neq j} q_k dz_k \right] dz_j - \int q_j \log q_j dz_j \right\}, \\ &= \sum_{i=1}^N \left\{ \int q_j \mathbb{E}_{k \neq j} \left[ \log p(x_i, z; \theta) \right] dz_j - \int q_j \log q_j dz_j - \mathcal{H}_{k \neq j} \right\}, \quad (9) \end{aligned}$$

where  $\mathcal{H}_{k \neq j}$  and  $\mathbb{E}_{k \neq j} [\dots]$  denote the entropy and the expectation with respect to probability densities over all latent variables  $z_k$  for  $k \neq j$ . The full derivation for Equation 9 is shown in Appendix A.

The ELBO in Equation 9 is maximized repeatedly with respect to each of the factors,  $q_j$ , while keeping the remaining factors,  $q_{k \neq j}$  constant. Convergence is guaranteed because the bound is convex with respect to each of the factors  $q_i$  (Boyd & Vandenberghe, 2004).

An extension to the mean field VI formulation is structured VI (Saul et al., 1996; Barber & Wiergerinck, 1998), which adds dependencies between the variables leading to a better approximation of the posterior probability density. There is, however, a trade-off as introducing these dependencies may make the variational optimization problem difficult to solve.

## 2.5 Coordinate Ascent Optimization

As for the optimization process, the coordinate ascent VI algorithm (CAVI) has been a popular choice for solving the traditional VI problem (Bishop, 2006b; Hoffman et al., 2013; Blei et al., 2017; Plummer et al., 2020) as it complements the mean field VI optimization process. The coordinate ascent algorithm can look like the EM algorithm<sup>3</sup> where the “E step” computes approximate conditionals of local latent variables and the “M step” computes a conditional of the global latent variables (Blei et al., 2017). Similar to mean field VI, this optimization process works by repeatedly updating each random variable’s variational parameters based on the variational parameters of the variables in its Markov blanket<sup>4</sup> (Winn & Bishop, 2005), and re-estimating the convergence of the ELBO (described in Algorithm 1). CAVI goes uphill on the ELBO of Equation 6, eventually finding a local optimum (Blei et al., 2017). Ganguly and Earp (2021) illustrate an example to approximate a mixture of Gaussians using coordinate ascent and mean field VI.

---

**Algorithm 1:** CAVI for the traditional VI optimization process

---

**Input:** Data  $x_{1:N}$   
**Output:** Variational parameters  $\xi_{1:N}$ ; generative model parameter  $\theta$   
 $\theta, \xi_{1:N} \leftarrow$  random initialization  
**while**  $\mathcal{L}(x; \xi_{1:N}, \theta)$  has not converged **do**  
    **for**  $i \in 1, \dots, N$  **do**  
         $\xi_i \leftarrow \arg \max_{\xi_i} \mathcal{L}(\xi_i, \theta; x_i)$   
    **end**  
    Compute  $\mathcal{L}(x; \xi_{1:N}, \theta)$   
     $\theta \leftarrow \arg \max_{\theta} \sum_{i=1}^N \mathcal{L}(x_i; \xi_i, \theta)$   
**end**  
**return**  $\xi_{1:N}; \theta$

---

Though this optimization process results in a closed-form solution for the optimal variational parameters, it is inefficient for large data sets as it requires a complete pass through the entire data set, sampling one data point at a time, at each iteration. Generally, the ELBO is a non-convex objective function and CAVI guarantees convergence to a local optimum and is sensitive to initialization (Blei et al., 2017). Furthermore, in combination with the mean field approximations, CAVI may lead to sub-optimal convergence as the former

3. Interested readers are requested to read Chapter 11.4 of Murphy (2013).

4. The Markov Blanket of a target variable is a minimal set of variables that the target variable is conditioned on while all other remaining variables in the model are probabilistically independent of the target variable (Tsamardinos et al., 2003).

explicitly ignores correlations between variables, thereby making the optimization problem more non-convex (Wainwright & Jordan, 2007).

### 3. Stochastic VI

In recent years, with the advent of big data, scalability, and efficiency have become the primary requirements for modern machine learning algorithms. In the field of VI, one notable development has been in the form of *stochastic variational inference* (SVI) (Hoffman et al., 2013) which combines natural gradients (Amari, 1998) and stochastic optimization (Robbins & Monro, 1951) to tackle the scalability issue of the traditional VI algorithm.

#### 3.1 Stochastic Optimization

In contrast to CAVI, which updates the variational parameters one data point at a time, SVI uses stochastic optimization (Robbins & Monro, 1951), following noisy estimates of the gradient of the ELBO, on a sub-sample of the data and updates the parameters based on that sub-sample.

We can re-construct the ELBO, formulated in Equation 6, an estimator of the full data set, based on sets of mini-batches,  $x^M$  as:

$$\mathcal{L}(x; \xi_{1:N}, \theta) \simeq \frac{N}{M} \sum_{i=1}^M \mathcal{L}(x_i; \xi_i, \theta) = \frac{N}{M} \hat{\mathcal{L}}(x^M; \xi^M, \theta), \quad (10)$$

where  $M$  is the randomly drawn sub-sample of data from  $N$  data points,  $x^M$  represents a random mini-batch of the data set, and  $\sum_{i=1}^M \mathcal{L}(x_i; \xi_i, \theta)$  is an estimate of the ELBO based on those  $M$  samples, which we denote as  $\hat{\mathcal{L}}(x^M; \xi^M, \theta)$ .

The methodology of SVI is to get a stochastic estimate of the ELBO based on a set of  $M$  examples at each iteration (with or without replacement). This allows us to take derivatives  $\nabla_{\xi^M, \theta} \hat{\mathcal{L}}(x^M; \xi^M, \theta)$  and update the local variational parameters based on the  $M$  samples as well as the global parameter,  $\theta$ , using stochastic gradient ascent. We repeat this process until the ELBO converges. Computational savings in SVI are obtained only for  $M \ll N$  (Zhang et al., 2019).

#### 3.2 Natural Gradients

Hoffman et al. (2013) proposed the idea of using *natural* gradients as opposed to using standard gradients for SVI to capture the information geometry of the parameter space for probability densities. Natural gradients adjust the direction of the traditional gradient by the use of a Riemannian metric. The classical gradient ascent method for a function  $f(\xi)$  tries to reach the function’s maxima by taking steps of size  $\rho$  in the direction of the steepest ascent for the gradient (when it exists) (Hoffman et al., 2013). This is formulated as:

$$\xi^{t+1} = \xi^t + \rho \nabla_{\xi} f(\xi^t),$$

where the gradient  $\nabla_{\xi} f(\xi)$  points in the same direction as the solution to

$$\arg \max_{\Delta \xi} f(\xi + \Delta \xi) \quad \text{subject to} \quad \|\Delta \xi\|^2 < \epsilon^2 \quad \text{and} \quad \epsilon \rightarrow 0.$$

During optimization, satisfying the condition above enables a movement away from  $\xi$  in the direction of the gradient. It is clear that in classical gradient ascent, the gradient direction depends on the Euclidean distance metric associated with the space where  $\xi$  resides. However, when optimizing an objective involving parameterized probability density functions, the Euclidean distance between two parameter vectors  $\xi$  and  $\xi + \Delta\xi$  is often a poor measure of the dissimilarity of the probability densities  $q(z; \xi)$  and  $q(z; \xi + \Delta\xi)$  (Hoffman et al., 2013). This is because the Euclidean metric fails to offer a meaningful explanation of distance in spaces where the local distance is not defined by the L2 norm.

Natural gradient corrects this issue by redefining the criterion for the gradient’s motion in the direction of the steepest ascent as:

$$\arg \max_{\Delta\xi} f(\xi + \Delta\xi) \quad \text{subject to} \quad D_{\text{KL}}^{\text{sym}}(\xi, \xi + \Delta\xi) < \epsilon^2 \quad \text{and} \quad \epsilon \rightarrow 0,$$

where  $D_{\text{KL}}^{\text{sym}}(\xi, \xi + \Delta\xi)$  is the symmetrized KL-divergence which is defined as:

$$D_{\text{KL}}^{\text{sym}}(\xi, \xi + \Delta\xi) = \mathbb{E}_{q(z; \xi)} \left[ \log \frac{q(z; \xi)}{q(z; \xi + \Delta\xi)} \right] + \mathbb{E}_{q(z; \xi + \Delta\xi)} \left[ \log \frac{q(z; \xi + \Delta\xi)}{q(z; \xi)} \right]. \quad (11)$$

While the Euclidean gradient points in the direction of steepest ascent in an Euclidean space, the natural gradient points in the direction of steepest ascent in the Riemannian space – a space where local distance is defined by KL-divergence rather than the L2 norm (Hoffman et al., 2013). In higher dimensions, using natural gradients, a movement of the same distance in different directions amounts to an equal change in the symmetrized KL-divergence (Blei et al., 2017). do Carmo (1993) introduced a Riemannian metric,  $I(\xi)$ , which defines the distance between  $\xi$  and a nearby vector  $\xi + \Delta\xi$  as:

$$\Delta\xi^T I(\xi) \Delta\xi \approx D_{\text{KL}}^{\text{sym}}(\xi, \xi + \Delta\xi),$$

where  $I(\xi)$  is the Fisher information matrix of  $q(z; \xi)$ . The full derivation is shown in Appendix B. Amari (1982) showed that natural gradients can be obtained by pre-multiplying the gradients with the inverse Fisher information matrix as:

$$\bar{\nabla}_{\xi} f(\xi) \triangleq [I(\xi)]^{-1} \nabla_{\xi} f(\xi).$$

where  $\bar{\nabla}$  and  $\nabla$  denote natural and stochastic gradients respectively.

The Fisher information matrix is essential to compute the Cramér-Rao lower bound for the performance analysis of an unbiased estimator — a minimum variance estimator for a parameter (Merberg & Miller, 2008; Yang & Amari, 1997). In VI, for a high dimensional parameter space, studying the covariance matrix for the variational estimator provides an estimate for its unbiasedness. The underlying high dimensional posterior structure might be rich, and the covariance matrix for the variational parameters captures the uncertainty of the KL-divergence being locked onto one of its many local modes. Additionally, it captures the sensitivity of the estimated posterior density with respect to small variations in the variational parameters (Knollmüller & Enßlin, 2019). For each of the variational parameters,  $\xi_i$ , to be unbiased estimators of the true parameters, they must satisfy the Cramér-Rao bound as:

$$\text{cov}(\xi) \geq [I(\xi)]^{-1}. \quad (12)$$

For the ELBO formulation in Equation 6, the Fisher information matrix for a variational parameter  $\xi_i$  is computed as:

$$I(\xi_i) := \mathbb{E}_q \left[ \nabla_{\xi_i} \log q(z|x_i; \xi_i) \nabla_{\xi_i} \log q(z|x_i; \xi_i)^T \right],$$

and for the generative model parameter  $\theta$ , it is computed as:

$$I(\theta) := \mathbb{E}_{p(x_i, z; \theta)} \left[ \nabla_{\theta} \log p(x_i, z; \theta) \nabla_{\theta} \log p(x_i, z; \theta)^T \right].$$

For a given step size  $\rho > 0$ , the natural gradient updates for the parameters at a time step  $t + 1$  is given by:

$$\xi_i^{t+1} = \xi_i^t + \rho [I(\xi_i^t)]^{-1} \nabla_{\xi_i} \mathcal{L}(x_i; \xi_i^t, \theta^t),$$

and

$$\theta^{t+1} = \theta^t + \rho [I(\theta^t)]^{-1} \nabla_{\theta} \mathcal{L}(x_i; \xi_i^t, \theta^t).$$

Additionally, the Fisher information matrix is a measure of the curvature for a probability density function as it is equal to the expected Hessian for that density function (Martens, 2020) (see Appendix C). This property is useful in problems wherein the Fisher information matrix is infeasible to compute, store, or invert. In such cases, simply computing the second moment of the derivatives is equivalent to approximating the Fisher information matrix.

The full SVI algorithm using mini-batches and natural gradients is described in Algorithm 2. This SVI methodology has aided in significant advancements in VI, such as gamma processes (Knowles, 2015) and more specifically in the development of the VAE (Kingma & Welling, 2014; Rezende et al., 2014) and its different variants.

### 3.3 Faster Convergence in SVI

The speed of convergence for the SVI optimization process depends on the variance of the gradient estimates. A lower variance ensures minimum gradient noise allowing for larger learning rates, leading to faster convergence. One approach to reducing the variance is to increase the mini-batch size, which leads to lower gradient noise as suggested by the law of large numbers (Foti et al., 2014). Another alternative is to use non-uniform sampling, such as importance sampling, to select mini-batches with lower gradient noise. Researchers have proposed different variants of importance sampling (Csiba & Richtárik, 2018; Gopalan et al., 2012; Parisi & Shankar, 1988; Zhao & Zhang, 2015) for this purpose. Although effective, the computational complexity of the sampling mechanism, however, relates to the dimensionality of model parameters (Fu & Zhang, 2017).

Increasing the mini-batch size might not always be plausible owing to hardware memory constraints. Recent trends in deep learning have shown that an increase in the speed of the training procedure can also be achieved by adjusting the learning rate while keeping the mini-batch size fixed. The idea is to let the empirical gradient variance guide the adaptation of the learning rate which is inversely proportional to the gradient noise in each iteration (Zhang et al., 2019). Gradually adapting the learning rate guarantees that every point

---

**Algorithm 2:** The SVI optimization process based on Hoffman et al. (2013)

---

**Input:** Data  $x_{1:N}$   
**Output:** Variational parameter  $\xi_{1:N}$ ; generative model parameter  $\theta$   
 $\xi_{1:N}, \theta \leftarrow$  random initialization  
**while**  $\mathcal{L}(x_i; \xi_{1:N}, \theta)$  has not converged **do**  
    **repeat**  
         $\mathcal{M} \leftarrow \{x^M | x^M \sim x_{1:N}, |x^M| = M\}$   
        **for**  $x^M \in \mathcal{M}$  **do**  
            **for**  $x_j \in x^M$  **do**  
                Compute  $\mathcal{L}(x_j; \xi_j, \theta)$   
                Compute  $\nabla_{\xi_j} \mathcal{L}(x_j; \xi_j, \theta); \nabla_{\theta} \mathcal{L}(x_j; \xi_j, \theta)$   
             $\hat{\nabla}_{\xi^M} \mathcal{L}(x^M; \xi^M, \theta) = \frac{1}{M} \sum_j \nabla_{\xi_j} \mathcal{L}(x_j; \xi_j, \theta)$   
             $\hat{\nabla}_{\theta} \mathcal{L}(x^M; \xi^M, \theta) = \frac{1}{M} \sum_j \nabla_{\theta} \mathcal{L}(x_j; \xi_j, \theta)$   
            **for**  $x_j \in x^M$  **do**  
                 $\bar{\nabla}_{\xi_j} \mathcal{L}(x^M; \xi^M, \theta) \triangleq [I(\xi_j)]^{-1} \hat{\nabla}_{\xi^M} \mathcal{L}(x^M; \xi^M, \theta)$   
                 $\xi_j \leftarrow$  Update parameters using  $\bar{\nabla}_{\xi^M} \mathcal{L}(x^M; \xi^M, \theta)$   
             $\bar{\nabla}_{\theta} \mathcal{L}(x^M; \xi^M, \theta) \triangleq [I(\theta)]^{-1} \hat{\nabla}_{\theta} \mathcal{L}(x^M; \xi^M, \theta)$   
             $\theta \leftarrow$  Update parameters using  $\bar{\nabla}_{\theta} \mathcal{L}(x^M; \xi^M, \theta)$   
             $\hat{\mathcal{L}}(x^M; \xi^M, \theta) = \sum_{j=1}^M \mathcal{L}(x_j; \xi_j, \theta)$   
        **until** all  $N$  data points are seen at least once;  
         $\mathcal{L}(x; \xi_{1:N}, \theta) \simeq \frac{N}{M} \hat{\mathcal{L}}(x^M; \xi^M, \theta)$   
**return**  $\xi_{1:N}; \theta$

---

in the parameter space can be reached, while the gradient noise decreases sufficiently fast to ensure convergence (Robbins & Monro, 1951). Several optimization techniques such as Adam (Kingma & Ba, 2015), AdaGrad (Duchi et al., 2010), AdaDelta (Zeiler, 2012) and RMSProp (Hinton et al., 2012), which make use of this idea, have been developed.

Other than increasing the mini-batch size or adapting the learning rate, variance reduction can be achieved using a control variate (Müller et al., 2020), a stochastic term, which when added to the stochastic gradient reduces the variance while keeping its expected value intact (Boyle, 1977). Using control variates for variance reduction is common in Monte Carlo simulation and stochastic optimization (Zhang et al., 2019; Ross, 2006; Wang et al., 2013).

#### 4. Black Box VI

The traditional VI process performs an initial analytical derivation for the ELBO before optimization which requires time and mathematical expertise. Thus, this makes it limited for use with only conditionally conjugate exponential families (Hoffman et al., 2013; Zhang, 2016). For this purpose, Ranganath et al. (2014) introduced the Black Box VI (BBVI) methodology that removes the need for analytical computation of the ELBO, relaxing this limitation. Based on the SVI optimization process, BBVI computes the gradient from Monte Carlo samples generated from the variational probability density.

The gradient estimate for the ELBO formulated in Equation 6 at a data point  $x_i$  can be written as:

$$\nabla_{\xi_i} \mathcal{L}(x_i; \xi_i, \theta) = \nabla_{\xi_i} \left[ \mathbb{E}_q \left[ \log p(x_i, z; \theta) - \log q(z|x_i; \xi_i) \right] \right]. \quad (13)$$

Drawing parallels from reinforcement learning, the variational probability density represents the policy as it is used to generate samples. The reward is the ELBO maximizing which guides the iterative learning process for the optimal variational parameters. An episode is formed by initializing the variational parameters and thus the policy, followed by drawing samples from the variational distribution, subsequently computing the ELBO or the reward, and updating the policy parameters. Thus, the gradient estimate of the ELBO as defined in Equation 13 is similar to computing the derivative of the expected reward while following a policy in a reinforcement learning setting. However, computing the gradient of the objective containing an expectation is non-trivial. For this purpose, reinforcement learning problems make use of the policy gradient theorem (Sutton et al., 2000), which states that the derivative of the expected reward is the expectation of the product of the reward and gradient of the log of the policy. Thus, using the policy gradient theorem (Sutton et al., 2000), Equation 13 can be re-written as:

$$\nabla_{\xi_i} \mathcal{L}(x_i; \xi_i, \theta) = \mathbb{E}_q \left[ \nabla_{\xi_i} \log q(z|x_i; \xi_i) \left[ \log p(x_i, z; \theta) - \log q(z|x_i; \xi_i) \right] \right].$$

The gradient  $\nabla_{\xi_i} \mathcal{L}(x_i; \xi_i, \theta)$  involving expectation with respect to  $q(z|x_i; \xi_i)$  can be approximated by drawing  $K$  independent samples,  $z_k$ , from the variational distribution and then computing the average of the function evaluated at these samples (Mohamed et al., 2020) as:

$$\nabla_{\xi_i} \mathcal{L}(x_i; \xi_i, \theta) \approx \frac{1}{K} \sum_{k=1}^K \left[ \log p(x_i, z_k; \theta) - \log q(z_k|x_i; \xi_i) \right] \nabla_{\xi_i} \log q(z_k|x_i; \xi_i), \quad (14)$$

where  $z_k \sim q(z|x_i; \xi_i)$  and  $\nabla_{\xi_i} \log q(z_k|x_i; \xi_i)$  is known as the score function (Cox & Hinkley, 1994). It is to be noted that the score function and sampling algorithms depend only on the variational distribution, not the underlying model. BBVI thus enables the practitioner to obtain an unbiased gradient estimator by sampling without having to derive the gradient of the ELBO explicitly (Zhang et al., 2019).

However, the variance of the gradient estimator under the Monte Carlo estimate in Equation 14 can be too large to be useful (Ranganath et al., 2014). Unlike SVI, where

sub-sampling from a finite set of data points leads to noisy gradient estimates, in the case of BBVI, it is the possible oversampling of the random variables that contributes to high noise in the gradients. Researchers have developed several variance reduction techniques for BBVI, such as the combination of Rao-Blackwellization and control variates (Ranganath et al., 2014), local expectation gradients (Titsias & Lázaro-Gredilla, 2015), and overdispersed importance sampling (Ruiz et al., 2016) but most notably, the reparameterization trick, introduced by Kingma and Welling (2014) (discussed in Section 4.1), is often used as it enables lower variance gradient estimates than the rest.

BBVI and its extensions have been one of the most significant developments of modern approximate inference techniques making amortized inference feasible in solving several deep learning problems (Lee et al., 2019a, 2019b; Liu et al., 2021, 2022).

### 4.1 The Reparameterization Trick

As established in Section 3.3 it is necessary to maintain a low variance for the stochastic gradient estimates to ensure faster convergence. Both Ranganath et al. (2014) and Kingma and Welling (2014) state that the Monte Carlo gradient estimates in BBVI (Equation 14) exhibit high variance. For this purpose, Kingma and Welling (2014) introduced a more practical gradient estimator for the lower bound in the form of a reparameterization trick. For a chosen approximate posterior  $q(z|x_i; \xi_i)$ , the trick allows a random variable  $z_i$  to be a differentiable transformation  $g_\phi(\epsilon, x_i)$  of a noise variable  $\epsilon$ , such that,

$$\begin{aligned} z_i &= g_\phi(\epsilon, x_i), \\ \epsilon &\sim p(\epsilon). \end{aligned}$$

Given a function  $f(z)$ , Monte Carlo estimates of expectations of it with respect to  $q(z|x_i; \xi_i)$  can be formed as follows:

$$\mathbb{E}_q[f(z)] = \mathbb{E}_{p(\epsilon)}\left[f(g_\phi(\epsilon, x_i))\right] \simeq \frac{1}{K} \sum_{k=1}^K f(g_\phi(\epsilon_k, x_i)) \quad \text{where } \epsilon_k \sim p(\epsilon).$$

Kingma and Welling (2014) show that applying this to the ELBO for VI in Equation 6, yields the stochastic estimator  $\tilde{\mathcal{L}}(x; \xi_{1:N}, \theta) \simeq \mathcal{L}(x; \xi_{1:N}, \theta)$ :

$$\begin{aligned} \tilde{\mathcal{L}}(x; \xi_{1:N}, \theta) &= \sum_{i=1}^N \left[ \frac{1}{K} \sum_{k=1}^K \left[ \log p(x_i, z_{(i,k)}; \theta) - \log q(z_{(i,k)}|x_i; \xi_i) \right] \right], \\ &= \sum_{i=1}^N \tilde{\mathcal{L}}(x_i; \xi_i, \theta), \end{aligned} \tag{15}$$

where  $z_{(i,k)} = g_\phi(\epsilon_{(i,k)}, x_i)$ ,  $\epsilon_k \sim p(\epsilon)$  and  $\tilde{\mathcal{L}}(x_i; \xi_i, \theta)$  is the stochastic estimator of the ELBO at a data point  $x_i$ . The gradient  $\nabla_{\xi_i} \tilde{\mathcal{L}}(x_i; \xi_i, \theta)$  for the estimator in Equation 15 can thus be written as:

$$\nabla_{\xi_i} \tilde{\mathcal{L}}(x_i; \xi_i, \theta) = \frac{1}{K} \sum_{k=1}^K \nabla_{\xi_i} \left[ \log p(x_i, z_{(i,k)}; \theta) - \log q(z_{(i,k)}|x_i; \xi_i) \right]. \tag{16}$$



Comparing Equation 16 with the policy gradient formulation for BBVI in Equation 14, we see that the gradient of the log joint distribution is a part of the expectation. The advantage of taking the gradient of the log joint is that this term is more informed about the direction of the maximum posterior mode (Zhang et al., 2019). This information also contributes to lower variance for the gradient estimates when compared to the policy gradient estimates. However, the ELBO in Equation 16 suffers from injected noise due to the use of Monte Carlo estimation for the lower bound. This noise can further be reduced by the use of control variates (Miller et al., 2017) or Quasi-Monte Carlo methods (Buchholz et al., 2018). Additionally, like BBVI the reparameterization trick allows us to derive the ELBO without having to compute analytic expectations. This reparameterization trick is also the basis of VAEs (Kingma & Welling, 2014; Robbins & Monro, 1951).

Although it promises a lower variance for the gradient estimates, the reparameterization trick, unlike the policy gradient scheme for BBVI, does not trivially extend to discrete distributions. In order for the trick to be applied to discrete distributions, further approximations for the variational posterior are required (see Jang et al., 2017; Maddison et al., 2017; Nalisnick & Smyth, 2017).

## 5. Amortized VI

The traditional VI optimization problem, as described in Section 2, maximizes the ELBO with respect to the variational parameters for each data point, followed by computing the optimal global generative model parameters as:

$$\arg \max_{\theta} \sum_{i=1}^N \arg \max_{\xi_i} \mathcal{L}(x_i; \xi_i, \theta).$$

This repetitive process introduces a new set of variational parameters for every observation, allowing these parameters to grow, at least, linearly with the observations. During this optimization process, each observation is processed independently of others, making the process *memoryless*, thereby guaranteeing that inference using one observation will not interfere with another (Gershman & Goodman, 2014). This further implies that there is no mechanism to re-use the knowledge from previous inferences on newer ones, and as such inferring on the same observation twice requires the same amount of computation which is equivalent to inferring two separate ones (Gershman & Goodman, 2014). When the number of observations is large, it can also lead to extensive computational inefficiency since there is no memory trace of inferences from previous data points. It might, therefore, be helpful to keep a memory trace of the past inferences (*memoizing*), although at a higher cost, to solve this scalability issue. However, it may be inaccurate to re-use a stored inference without modification as newer observations might be related or modifications to previous ones.

In the case of VI, these issues are solved by “amortizing” the optimization process, where instead of optimizing for each data point independently, the optimization cost is spread out across multiple instances, reducing the overall computational burden (Amos, 2023). For this purpose, amortized VI makes use of a stochastic function, which maps the observed variable to the latent variable belonging to the variational posterior density, the parameters of which are learned during the optimization process. Therefore, instead of having separate parameters for each observation, the estimated function can infer latent variables even for

Method	Properties
Traditional VI	<p><b>Methodology:</b></p> <ul style="list-style-type: none"> <li>• Analytical approximation of the posterior probability density for statistical inference over latent variables.</li> <li>• Formulates statistical inference as an optimization problem using a suitable divergence measure.</li> <li>• Introduces a local variational parameter for every observation.</li> <li>• Uses coordinate ascent to optimize the variational parameters for each observation iteratively.</li> </ul> <p><b>Advantages:</b></p> <ul style="list-style-type: none"> <li>• Use the ELBO to tractably compute the evidence to encourage the chosen statistical model to fit the data better.</li> </ul> <p><b>Limitations:</b></p> <ul style="list-style-type: none"> <li>• Inefficient in scaling to large datasets.</li> <li>• Coordinate ascent may encourage convergence to a local optimum.</li> <li>• Requires analytical derivation of the ELBO.</li> </ul>
SVI	<p><b>Methodology:</b></p> <ul style="list-style-type: none"> <li>• Uses gradient-based optimization to update the local variational parameters.</li> <li>• Optimization is based on mini-batches of data rather than iterating over every observation.</li> <li>• Can be combined with natural gradients to capture the dissimilarities of probability densities efficiently.</li> </ul> <p><b>Advantages:</b></p> <ul style="list-style-type: none"> <li>• Variance reduction of the gradients can be achieved by either increasing the mini-batch size or adjusting the learning rate during training or using control variates.</li> <li>• Fast convergence.</li> <li>• Scalable to large datasets.</li> </ul> <p><b>Limitations:</b></p> <ul style="list-style-type: none"> <li>• Still requires analytical derivation of the ELBO.</li> </ul>
BBVI	<p><b>Methodology:</b></p> <ul style="list-style-type: none"> <li>• Uses gradient-based optimization to update the local variational parameters.</li> <li>• Optimization is based on mini-batches of data rather than iterating over every observation.</li> <li>• Uses the reparameterization trick to maintain a low variance for the stochastic gradient estimates for the ELBO.</li> </ul>

Table 2: Comparison of different VI methods (continued on the next page).

	<p><b>Advantages:</b></p> <ul style="list-style-type: none"> <li>• Omits the requirement to derive the ELBO analytically.</li> <li>• Fast convergence.</li> <li>• Scalable to large datasets.</li> </ul> <p><b>Limitations:</b></p> <ul style="list-style-type: none"> <li>• The reparameterization trick does not extend to discrete distributions.</li> </ul>
Amortized VI	<p><b>Methodology:</b></p> <ul style="list-style-type: none"> <li>• Amortizes the inference by the use of a stochastic function, such as a neural network, to map the observed variables to the latent variables.</li> <li>• Uses the BBVI methodology for ELBO optimization.</li> </ul> <p><b>Advantages:</b></p> <ul style="list-style-type: none"> <li>• Flexible memoized re-use of past inferences to compute inference on newer observations.</li> <li>• Omits the requirement to derive the ELBO analytically.</li> <li>• Fast convergence.</li> <li>• Scalable to large datasets.</li> </ul> <p><b>Limitations:</b></p> <ul style="list-style-type: none"> <li>• The use of a stochastic function to amortize the inference leads to inconsistent representation learning.</li> <li>• Sub-optimal inference arising largely due to a coding efficiency gap known as amortization gap.</li> <li>• Generalization gap depends on the capacity of chosen neural network as the stochastic function.</li> </ul>

Table 2: Comparison of different VI methods.

new data points without re-running the optimization process all over again on the new data points. This process allows for computational efficiency and flexible memoized re-use (Michie, 1968) of relevant information from past inferences on previously unseen data. Table 2 compares the methodology, advantages, and limitations of traditional VI, SVI, BBVI, and amortized VI.

With recent advances in deep learning, researchers have extensively used neural networks in the form of this stochastic function to estimate the parameters of the posterior probability density. Neural networks are powerful frameworks that allow for efficient amortization of inference. Additionally, the development of GPU-assisted neural network training has also led to the usage of complex neural network architectures with amortized VI (e.g., Radford et al., 2016; Karras et al., 2019; Chen et al., 2016; Pidhorskyi et al., 2020), allowing extraction of information from high-dimensional data without human supervision (Simonyan et al., 2014).

While a local variational parameter,  $\xi_i$ , is introduced for every observation,  $x_i$ , in traditional VI, as shown in Figure 2, in case of amortized VI, the variational parameters,  $\phi$ , are globally shared by all the observations, illustrated by the graphical model in Figure 3. Thus,

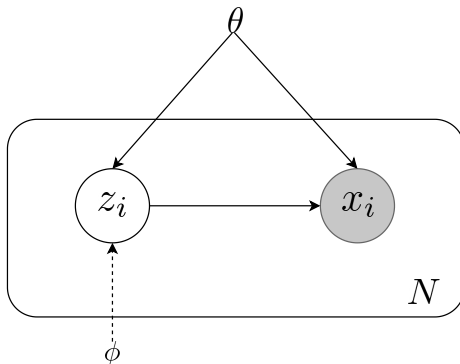


Figure 3: Illustration of the directed graphical model in the case of amortized VI with  $N$  observed data points. The global and the amortized variational parameters are represented by  $\theta$  and  $\phi$  respectively.

the ELBO defined for the traditional VI optimization problem, established in Equation 6, can be modified for amortized VI as:

$$\mathcal{L}(x; \phi, \theta) = \sum_{i=1}^N \mathbb{E}_{q_\phi} \left[ \log p(x_i, z; \theta) - \log q(z|x_i; \phi) \right] = \sum_{i=1}^N \mathcal{L}(x_i; \phi, \theta), \quad (17)$$

where  $\mathbb{E}_{q_\phi} [\cdot] = \mathbb{E}_{q(z|x_i; \phi)} [\cdot]$  is the expectation with respect to the variational posterior  $q(z|x_i; \phi)$ .

Based on randomly drawn mini-batches of size  $M$ , we can re-construct the ELBO for amortized VI formulated in Equations 17 as:

$$\mathcal{L}(x; \phi, \theta) \simeq \frac{N}{M} \sum_{i=1}^M \mathcal{L}(x_i; \phi, \theta) = \frac{N}{M} \hat{\mathcal{L}}(x^M; \phi, \theta). \quad (18)$$

Similar to the ELBO formulation in SVI (see Equation 10), the ELBO estimate based on  $M$  sub-sample of the data is  $\sum_{i=1}^M \mathcal{L}(x_i; \phi, \theta)$  which we denote by  $\hat{\mathcal{L}}(x^M; \phi, \theta)$ . The optimization process for amortized VI (shown in Algorithm 3) usually follows the stochastic gradient ascent to ensure faster convergence.

However, using stochastic gradients does not guarantee an optimal solution as the gradient updates follow the steepest ascent in a Euclidean space without considering the parameter space’s information geometry. Natural gradients offer a solution to this problem as they reformulate the criterion for the gradient updates using the inverse of the Fisher Information matrix. With the use of deep learning models, comprising millions of parameters, in the form of the stochastic function, this computation for the inverse is infeasible as it has a time complexity of  $\mathcal{O}(d^3)$  with  $d$  being the dimension of the parameter space. As discussed in Section 3.2, a simple trick would be to use the Hessian of the gradients to compute the Fisher Information matrix and subsequently, its inverse. The Hessian can be

---

**Algorithm 3:** The amortized VI optimization process using stochastic gradient ascent

---

**Input:** Data  $x_{1:N}$

**Output:** Variational parameter  $\phi$ ; generative model parameter  $\theta$

$\phi, \theta \leftarrow$  random initialization

**while**  $\mathcal{L}(x; \phi, \theta)$  has not converged **do**

**repeat**

$\mathcal{M} \leftarrow \{x^M | x^M \sim x_{1:N} \text{ and } |x^M| = M\}$

**for**  $x^M \in \mathcal{M}$  **do**

**for**  $x_j \in x^M$  **do**

        Compute  $\mathcal{L}(x_j; \phi, \theta)$

        Compute  $\nabla_{\phi} \mathcal{L}(x_j; \phi, \theta); \nabla_{\theta} \mathcal{L}(x_j; \phi, \theta)$

$\hat{\nabla}_{\phi} \mathcal{L}(x^M; \phi, \theta) = \frac{1}{M} \sum_j \nabla_{\phi} \mathcal{L}(x_j; \phi, \theta)$

$\hat{\nabla}_{\theta} \mathcal{L}(x^M; \phi, \theta) = \frac{1}{M} \sum_j \nabla_{\theta} \mathcal{L}(x_j; \phi, \theta)$

$\phi; \theta \leftarrow$  Update parameters using  $\hat{\nabla}_{\phi} \hat{\mathcal{L}}(x^M; \phi, \theta); \hat{\nabla}_{\theta} \hat{\mathcal{L}}(x^M; \phi, \theta)$

$\hat{\mathcal{L}}(x^M; \phi, \theta) = \sum_{j=1}^M \mathcal{L}(x_j; \phi, \theta)$

**until** all  $N$  data points are seen at least once;

$\mathcal{L}(x; \phi, \theta) \simeq \frac{N}{M} \hat{\mathcal{L}}(x^M; \phi, \theta)$

**return**  $\phi; \theta$

---

computed using methods such as automatic differentiation or the reparameterization trick (Khan et al., 2018). It is, however, not common to compute Hessians for deep models due to its high computational cost (Khan & Nielsen, 2018).

The Hessian computation can be avoided using the classical Gauss-Newton method (Schraudolph, 2002; Martens, 2020), in which the Hessian is approximated as the second moment of the gradients. The optimizer Adam (Kingma & Ba, 2015) can be used for this purpose as it computes the first and second moment of the gradients. Further simplification in computation can be achieved by limiting the second moment to be a diagonal matrix, thereby enabling the computation for the inverse Fisher Information matrix to be  $\mathcal{O}(d)$  rather than  $\mathcal{O}(d^3)$ , making it easy to apply to large deep learning problems.

### 5.1 Variational Auto-Encoder (VAE)

The VAE framework, developed by Kingma and Welling (2014) and Rezende et al. (2014), is an example of a statistical model that combines deep neural networks with the amortized VI optimization process. VAEs employ two deep neural networks: a probabilistic decoder, i.e., a top-down generative model that creates a mapping from a latent variable  $z_i$  to a data point  $x_i$ , and a probabilistic encoder, i.e., a bottom-up inference model that approximates the posterior probability density,  $p(z|x; \theta)$ . Correspondingly, these networks are commonly

referred to as *generative* and *recognition* networks, respectively. The graphical model for the VAE framework (Kingma & Welling, 2014) is illustrated in Figure 4.

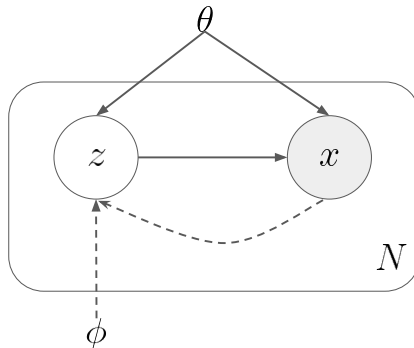


Figure 4: Graphical model for the VAE framework. Solid lines denote the generative model, dashed lines denote the variational approximation to the intractable posterior density. The variational parameters  $\phi$  are learned jointly with the generative model parameters  $\theta$  (Kingma & Welling, 2014).

We can get an intuitive understanding of the ELBO for VAEs by further re-arranging the terms of Equation 17 as:

$$\begin{aligned}
 \mathcal{L}(x; \phi, \theta) &= \sum_{i=1}^N \mathcal{L}(x_i; \phi, \theta) = \sum_{i=1}^N \mathbb{E}_{q_\phi} \left[ \log p(x_i, z; \theta) - \log q(z|x_i; \phi) \right], \\
 &= \sum_{i=1}^N \mathbb{E}_{q_\phi} \left[ \log p(x_i|z; \theta) + \log p(z; \theta) - \log q(z|x_i; \phi) \right], \\
 &= \sum_{i=1}^N \left[ \mathbb{E}_{q_\phi} \left[ \log p(x_i|z; \theta) \right] - D_{\text{KL}}(q(z|x_i; \phi) \parallel p(z; \theta)) \right]. \quad (19)
 \end{aligned}$$

Thus, Equation 19 establishes ELBO to be the sum of the expected log-likelihood and the negative KL-divergence between the approximate density and the prior over the latent variable evaluated at individual data points. The KL-divergence term can then be interpreted as regularizing  $\phi$ , encouraging the approximate posterior to be close to the prior  $p(z; \theta)$  (Kingma & Welling, 2014). Furthermore, Equation 19 establishes a connection to auto-encoders, as the first term is an expected negative reconstruction error while the KL-divergence term acts as a regularizer.

Kingma and Welling (2014) showed that applying the reparameterization trick to the ELBO formulation for VAEs in Equation 19 yields the Stochastic Gradient Variational Bayes (SGVB) estimator  $\tilde{\mathcal{L}}^B(x; \phi, \theta) \simeq \mathcal{L}(x; \phi, \theta)$ :

$$\tilde{\mathcal{L}}^B(x; \phi, \theta) = \sum_{i=1}^N \left[ \frac{1}{K} \sum_{k=1}^K \left[ \log p(x_i|z_{(i,k)}; \theta) \right] - D_{\text{KL}}(q(z|x_i; \phi) \parallel p(z; \theta)) \right]. \quad (20)$$

Often the KL-divergence term in Equation 20 can be integrated analytically such that only the expected reconstruction error requires estimation by sampling (Kingma & Welling, 2014). For the variational posterior, VAEs employ mean field approximation and as a simplifying choice, it is chosen to be a multivariate Gaussian with diagonal covariance structure:

$$q(z|x_i; \phi) = \prod_{j=1}^J q(z_j|x_i; \phi), \quad \log q(z|x_i; \phi) = \log \mathcal{N}(z; \mu_i, \sigma_i^2 \mathbb{I}),$$

where  $J$  is the dimensionality of  $z$ , the mean,  $\mu_i$ , and standard deviation,  $\sigma_i$ , are the outputs of the encoder, i.e. non-linear functions of data point  $x_i$  and the variational parameters  $\phi$ , which summarizes the corresponding neural network parameters (Kingma & Welling, 2014; Zhang et al., 2019).

As discussed in Section 4.1, the posterior is sampled as:

$$z_{(i,k)} \sim q(z|x_i; \phi) \quad \text{using} \quad z_{(i,k)} = g_\phi(x_i, \epsilon_{(i,k)}) = \mu_i + \sigma_i \odot \epsilon_{(i,k)},$$

where  $\epsilon_k \sim \mathcal{N}(0, \mathbb{I})$  and  $\odot$  denotes element-wise product.

Usually for the prior, a multivariate normal is chosen so that the latent variable  $z$  can be drawn as:

$$z = \mathcal{N}(0, \mathbb{I}).$$

However, a standard normal prior often leads to an over-regularization of the approximate posterior, which results in a less informative learned latent representation of the data (Chen et al., 2020). Recent advancements have shown to improve the representation learning in VAEs by modeling the prior to be dependent on additional parameters. Finally, the log-likelihood is computed as:

$$\log p(x_i|z_{(i,k)}; \theta) = \log p(x_i|\mu_i + \sigma_i \odot \epsilon_{(i,k)}; \theta),$$

signifying that the decoder network, parameterized by  $\theta$ , generates a data point  $x_i$  through non-linear transformations of the latent vector  $z_i$ . Thus, the resulting SGVB estimator for VAE is:

$$\tilde{\mathcal{L}}^B(x; \phi, \theta) \simeq \sum_{i=1}^N \left[ \frac{1}{2} \sum_{j=1}^J \left( 1 + \log(\sigma_{(i,j)}^2) - \mu_{(i,j)}^2 - \sigma_{(i,j)}^2 \right) + \frac{1}{K} \sum_{k=1}^K \log p(x_i|\mu_i + \sigma_i \odot \epsilon_{(i,k)}; \theta) \right], \quad (21)$$

where  $\mu_{(i,j)}$  and  $\sigma_{(i,j)}$  denote the variational mean and standard deviation for the  $j$ -th element of these vectors evaluated at data point  $x_i$ , respectively. This formulation allows the stochastic estimate of the ELBO to be differentiated with respect to both  $\phi$  and  $\theta$  for gradient estimation.

In order to obtain a tighter ELBO and hence better variational approximations, importance sampling can be used to get a lower variance estimate of the evidence (Thin et al., 2021). This technique also forms the basis of the Importance Weighted Auto-Encoder (IWAE) (Burda et al., 2016) where the ELBO is computed as:

$$\mathcal{L}(x; \phi, \theta) = \sum_{i=1}^N \mathbb{E}_{q_\phi} \left[ \log \frac{1}{K} \sum_{k=1}^K \left[ \frac{p(x_i|z_k; \theta)p(z; \theta)}{q(z_k|x_i; \phi)} \right] \right], \quad (22)$$

which is a  $K$ -sample importance weighting estimate of the log evidence. Burda et al. (2016) showed that the true marginal likelihood is approached as  $K \rightarrow \infty$ . However, Nowozin (2018) proved that IWAEs introduce a biased estimator for the true marginal likelihood where the bias is reduced at a rate of  $\mathcal{O}(1/K)$ . In addition, importance sampling is known to perform poorly in high dimensions (MacKay, 2002). To address these issues, Thin et al. (2021) proposed the Langevin Monte Carlo VAE (L-MCVAE), based on Sequential Importance Sampling (SIS), that provides a tighter ELBO than standard techniques as well as an unbiased estimator for the evidence. However, Rainforth et al. (2018) provided empirical evidence that increasing the tightness of the ELBO independently to the expressive capacity of the recognition network can prove detrimental to its learning process. In their study, Rainforth et al. (2018) demonstrated that deviating from this conventional training approach, specifically, by employing tighter bounds for training generative networks and looser bounds for recognition networks, more accurate posterior approximations and enhanced generative performance could be achieved. Thus, they proposed three algorithms namely the Partially Importance Weighted Auto-Encoder (PIWAE), the Multiply Importance Weighted Auto-Encoder (MIWAE), and the Combination Importance Weighted Auto-Encoder (CIWAE). Each of these algorithms represents improvements over the IWAE and encompasses the standard IWAE as a special case. Rainforth et al. (2018) evaluated the performance of these algorithms in terms of their ability to balance the training objectives between the recognition and the generative networks. While MIWAE and CIWAE primarily enabled achieving this balance, it was PIWAE that exhibited distinctive characteristics. PIWAE not only successfully balanced the training objectives between their recognition and generative networks but also demonstrated simultaneous improvements in the training of both networks. By focusing on improving the training of its recognition network, PIWAE indirectly influenced and heightened its generative network performance, while resulting in more accurate posterior approximations.

## 5.2 Caveats and Solutions

Although amortizing the inference contributes toward making the VI optimization faster and more scalable, it introduces certain issues. In this section, we describe pertinent issues such as the amortization gap, inconsistent representation learning in VAEs, the generalization gap, and the problem of posterior collapse. Additionally, we cover the various methods that have been proposed in recent years to solve these issues.

### 5.2.1 SUB-OPTIMAL INFERENCE

In VI, the typical choice in the variational family is either factorized independent Gaussians or other mean field approximations for ease of analytical computation. However, this limits the expressibility of the variational approximation by ignoring local dependencies between latent variables. This limiting nature of the variational methodology results in the *approximation gap* which can be reduced by choosing a family of variational densities that is flexible enough to contain the true posterior as one solution (Rezende & Mohamed, 2015).

For this purpose, Rezende and Mohamed (2015) proposed the concept of normalizing flow (Tabak & Turner, 2013; Tabak & Vanden-Eijnden, 2010) as a means to improve upon the expressiveness of the variational approximation. A normalizing flow describes the trans-



formation of a probability density through a sequence of invertible mappings (Rezende & Mohamed, 2015). It involves repeatedly applying change of variables to transform the simple initial variational probability density into a richer approximation to better match the true posterior density. The main idea is to consider an invertible, smooth mapping  $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$  with inverse  $f^{-1} = g$ , such that  $g(f(z)) = z$ . Thus, a random variable  $z \sim q(z)$  can be transformed using the invertible, smooth function  $f$  into a new random variable  $z' = f(z)$  with density  $q(z')$  as:

$$q(z') = q(z) \left| \frac{\partial f^{-1}(z')}{\partial z'} \right| = q(z) \left| \frac{\partial f(z)}{\partial z} \right|^{-1}, \quad (23)$$

which is obtained using change of variables. With an appropriate choice of the transformation function, such that  $\left| \frac{\partial f}{\partial z} \right|$  is easily computable, and applying Equation 23 successively, complex and multi-modal densities can be efficiently constructed from simple factorized distributions such as independent Gaussians. In addition, different variants such as Langevin and Hamiltonian flows, invertible linear-time transformations as well as autoregressive flow (Chen et al., 2017) have been proposed based on the concept of normalizing flows.

Alternatively, capturing the dependencies between latent variables increases the expressiveness of the variational family which mean field approximations, though effective in VI, discard. The idea of auxiliary variables has been employed in hierarchical variational models (HVMs) (Ranganath et al., 2016) where dependencies between latent variables are induced similarly to the induction of dependencies between data in hierarchical Bayesian models.

Furthermore, in the case of amortized VI, using a stochastic function to estimate the variational density parameters instead of optimizing for each data point introduces a coding efficiency gap known as the *amortization gap* (Cremer et al., 2018). While offering significant benefits in computational efficiency, standard amortized inference models can suffer from sizable amortization gaps (Krishnan et al., 2018). On the one hand, where the complexity of the variational density determines the approximation gap, it is the capacity of the stochastic function that results in the amortization gap. The approximation gap, along with the amortization gap, contributes toward the *inference gap*, which is the gap between the marginal log-likelihood and the ELBO.

In their work, Cremer et al. (2018) observed that for VAEs, trained especially on complex data sets, the amortization gap contributes significantly towards the inference gap. They combined normalizing flow with the induction of hierarchical auxiliary variables to increase the expressiveness of the variational approximation. This resulted in generalizing inference in addition to improving the complexity of the variational approximation. Cremer et al. (2018) demonstrated through their experiments that increasing the capacity of the encoder reduces the amortization gap. However, Shu et al. (2018) argue that an over-expressive encoder degrades generalization. Therefore, in their paper, Shu et al. (2018) introduced the concept of amortized inference regularization which is a regularization technique that restricts the capacity of the encoder to prevent both the inference and the generative models from over-fitting to the training set (explained in detail in Section 5.2.3).

A recent research trend has seen an effort toward reducing the amortization gap using an iterative training approach. For instance, Hjelm et al. (2016) proposed a training procedure to iteratively refine the chosen approximate posterior estimated by a recognition network. The proposed learning algorithm follows expectation-maximization (EM), wherein the E-

step the recognition network is used to initialize the parameters of the variational posterior which are then iteratively refined. This refinement procedure provides a tight and asymptotically unbiased estimate of the ELBO, which is used to train both the recognition and generative models during the M-step. Moreover, this refinement procedure results in lower variance Monte Carlo estimates for the approximate posterior and provides a more accurate estimate of the log-likelihood of the model (Hjelm et al., 2016). On a similar note, Marino et al. (2018) proposed an iterative training scheme that reduces the amortization gap in standard VAEs by directly encoding the gradients of the parameters of the approximated posterior. VAEs create direct, static mappings from observations to the parameters of the approximate posterior with the optimization of these parameters replaced with the optimization of a shared, i.e., amortized, set of parameters  $\phi$  for the recognition model (Marino et al., 2018). This optimization process makes the recognition network in a VAE a purely bottom-up inference process which does not correspond well to perception (Sønderby et al., 2016). In other words, inference is as much a top-down as it is a bottom-up process, and therefore, in order to combine the two, Marino et al. (2018) proposed a training regimen that enables a VAE to learn to perform inference by iteratively encoding the gradients of the approximate posterior parameters, which are rarely performed in practice. The results from their experiments showed that this form of iterative training continuously refined the approximate posterior estimate, thereby, reducing the amortization gap. However, this method also required additional computation over VAEs with similar architectures.

A semi-amortized approach proposed by Kim et al. (2018) is another iterative training approach that used amortized variational inference to initialize the variational parameters and then subsequently ran SVI procedure for local iterative refinement of these parameters. The resulting Semi-Amortized VAE (SA-VAE) framework had a smaller amortization gap than vanilla VAEs. Additionally, it avoided the posterior collapse phenomenon, common in VAEs, wherein the variational posterior collapses to the prior (discussed in detail in Section 5.2.4). However, this semi-amortized approach suffered from additional computation overhead, owing to the additional SVI optimization at test time. In order to tackle this issue, Kim and Pavlovic (2021) proposed an approach that aimed at reducing the amortization gap by considering this difference between the true posterior and amortized posterior distribution as random noise. They showed that this approach is more efficient than the recent semi-amortized approaches, being able to perform a single feed-forward pass during inference.

### 5.2.2 INCONSISTENT REPRESENTATION LEARNING

VAEs are powerful frameworks that make use of amortized VI for unsupervised learning. Amortizing the inference enables VAEs to perform scalable variational posterior approximation in deep latent variable models. As discussed in Section 5.1, VAEs amortize the posterior inference by the use of a stochastic function that maps observations to their subsequent representations in the latent space. Once trained, the recognition model of a VAE can be used to obtain low-dimensional representations of data, the quality of which determines the applicability of VAEs. However, the recognition model of a fitted VAE tends to map an observation and its subsequent semantics-preserving transformation (e.g., rotation, translation) to different parts in the latent space (Sinha & Dieng, 2021). This inconsistency

of the recognition network has an adverse effect on the quality of the learned representations as well as generalization. To enforce consistency in VAEs, Sinha and Dieng (2021) proposed a regularization technique; the idea of which is to minimize the KL-divergence between the variational approximations when conditioned on an observation and when conditioned on its randomly transformed semantics-preserving counterpart. Sinha and Dieng (2021) termed the resulting VAE trained with this regularization technique as the consistency-regularized VAE (CR-VAE).

Based on the formulation of the ELBO for amortized VI from Equation 17, Sinha and Dieng (2021) defined a semantics-preserving transformation distribution  $t(\tilde{x}|x_i)$  for a data point  $x_i$  with the argument that a vanilla VAE, once fit to data, fails to output similar latent representations for both  $x_i$  and  $\tilde{x}_i$  in comparison to a good representation learning algorithm. The CR-VAE addresses this issue by re-defining the ELBO objective for VAEs as:

$$\mathcal{L}_{\text{CR-VAE}}(x; \phi, \theta) = \mathcal{L}(x; \phi, \theta) + \sum_{i=1}^N \left[ \mathbb{E}_{t(\tilde{x}|x_i)} \left[ \mathcal{L}(x_i; \phi, \theta) \right] - \lambda \cdot \mathcal{R}(x_i; \phi) \right], \quad (24)$$

s.t.,

$$\tilde{x}_i \sim t(\tilde{x}|x_i) \iff \epsilon \sim \mathcal{U}[-\delta, \delta] \quad \text{and} \quad \tilde{x}_i = g(x_i, \epsilon).$$

The function  $g(x_i, \epsilon)$  is a semantics-preserving transformation of a data point  $x_i$ , e.g., translation with a random length  $\epsilon$  drawn from a uniform distribution  $\mathcal{U}[-\delta, \delta]$  for some threshold  $\delta$  (Sinha & Dieng, 2021). Additionally, the final term in Equation 24 is the regularization term which is defined as:

$$\mathcal{R}(x_i; \phi) = \mathbb{E}_{t(\tilde{x}|x_i)} \left[ D_{\text{KL}}(q(z|\tilde{x}_i; \phi) \parallel q(z|x_i; \phi)) \right].$$

Maximizing the objective in Equation 24 maximizes the likelihood of the data and their augmentations while enforcing consistency through  $\mathcal{R}(x_i; \phi)$  (Sinha & Dieng, 2021). The regularization term only affects the recognition model (with parameters  $\phi$ ), and minimizing it forces the representations of each observation and their corresponding augmentations to lie close to each other in the latent space. The strength of the regularizer is controlled by the hyper-parameter  $\lambda \geq 0$ .

Through their experiments on the MNIST (Lecun et al., 1998), Omniglot (Lake et al., 2015), and CelebA (Liu et al., 2015) data sets, Sinha and Dieng (2021) showed that CR-VAE improved the learned representations over vanilla VAEs and improved generalization performance. Additionally, they applied the regularization technique to IWAE (Burda et al., 2016),  $\beta$ -VAE (Higgins et al., 2017), and nouveau VAE (Vahdat & Kautz, 2020) and demonstrated that CR-VAEs yielded better representations and generalized performance than their base VAEs.

Further research in this direction were conducted to show that vanilla VAE models are not *auto-encoding* (Cemgil et al., 2020), i.e., samples from the generative network are not mapped to corresponding representations by the recognition network. Cemgil et al. (2020) derived the Auto-encoding VAE (AVAE) framework that utilizes a reformed lower bound to achieve adversarial robustness for the learned representations. In addition, an AVAE model optimized with this lower bound facilitates data augmentations and self-supervised density estimation. The central idea of AVAE is making the recognition and the generative

networks to be consistent both on the training data and on the auxiliary observations generated by the generative network (Cemgil et al., 2020). Through their experiments on the colourMNIST and CelebA data sets, Cemgil et al. (2020) showed that their proposed AVAE framework, using both multi-layered perceptron and convolutional neural network architectures, achieved high adversarial accuracy without adversarial training.

### 5.2.3 GENERALIZATION GAP

On examining the amortized VI formulation for the ELBO from Equation 19, Shu et al. (2018) concluded that it is a *data-dependent regularized* maximum likelihood objective, which is a means to restrict the recognition model capacity. While a low-capacity recognition model increases the amortization gap, an over-expressive one harms generalization. This amortized inference regularization (AIR) strategy encourages recognition model smoothness while reducing the inference gap and increasing the log-likelihood on the test set. Shu et al. (2018) proposed a modification to the vanilla VAE by injecting noise into the recognition model resulting in the denoising VAE (DVAE). Although DVAEs were originally proposed by Im et al. (2017), Shu et al. (2018) further demonstrated that the optimal DVAE model is a kernel regression model, and the variance of the injected noise controls the smoothness of the optimal recognition model. Additionally, Shu et al. (2018) proposed the weight-normalized inference (WNI) method which leverages the weight normalization technique introduced by Salimans and Kingma (2016), to control the capacity and the smoothness of the recognition model. Through their experiments on the Caltech 101 Silhouettes (Marlin et al., 2010) and statically binarized MNIST and Omniglot data sets, Shu et al. (2018) showed that regularizing the recognition either by the DVAE or the WNI-VAE method improved the test set log-likelihood performance. From the results on these data sets, a consistent reduction of the test set inference gap was noticed when the inference model was regularized.

As discussed in Section 5.1, the VAE amortizes the inference to scale its training to large data sets, making it a popular choice for several applications such as density estimation, lossless compression, and representation learning (Zhang et al., 2022). However, the use of amortized inference during its training phase can lead to poor generalization performance. In order to tackle this issue, Zhang et al. (2022) introduced a training methodology for the recognition network in a VAE to reduce over-fitting to the training data and hence, improve generalization. Due to the lack of sufficient training data, a flexible posterior approximation can lead the recognition network to reduce the overall inference gap but also over-fit to the training data. Zhang et al. (2022) proposed a self-consistent training method wherein a mixture of samples from the training data set and those generated by the generative model were fed to the recognition network during the training phase. This mixture of distributions could be interpreted as a form of training data augmentation to help overcome the over-fitting caused by the application of amortized inference (Zhang et al., 2022). The results from their experiments showed that this training approach consistently improved the generalization performance, as measured by the negative ELBO on both the binary and grey-scale MNIST data sets (Salakhutdinov & Murray, 2008; Lecun et al., 1998).

### 5.2.4 POSTERIOR COLLAPSE

Posterior collapse is a phenomenon often observed in VAE training, which arises when the variational posterior distribution lies close, or as the name suggests, collapses, to the prior. This causes the generative network to ignore a subset of the latent variables. The model, hence, fails to learn a valuable representation of the data.

Several works (Yang et al., 2017; Semeniuta et al., 2017; Zhao et al., 2019; Tolstikhin et al., 2018; Takida et al., 2021) suggest this phenomenon stems from two main reasons. First, in cases where the generative network is especially powerful, it models the observed variable  $x$  independently, causing the latent variables  $z$  to get ignored. Second, with the training objective of maximizing the ELBO and minimizing the KL-divergence term, as observed in Equation 19, the variational posterior collapses to the prior as the KL-divergence term approaches zero. Moreover, Lucas et al. (2019) suggested that this occurs due to the spurious local maxima in the training objective instead.

Various approaches tackle the posterior collapse by either replacing the generative network with a weaker capacity alternative (Yang et al., 2017; Semeniuta et al., 2017), or by modifying the ELBO training objective (Zhao et al., 2019; Tolstikhin et al., 2018; Takida et al., 2021). Takida et al. (2021) demonstrated that inconsistency in choosing certain hyperparameters, more specifically data variance parameters, leads to over-smoothing and, in turn, posterior collapse. They proposed an adaptively regularized ELBO objective function to control the model smoothing and posterior collapse. Semeniuta et al. (2017), on the other hand, proposed replacing the traditional Recurrent Neural Networks for text generation with a weaker convolution-deconvolution architecture, which results in faster convergence as well as forces the network to learn from the latent dimensions.

Although these approaches successfully tackle posterior collapse, they either require an alteration to the training objective or do not fully utilize the recent advances in deep auto-regressive networks. Alternatively, Razavi et al. (2019) proposed  $\delta$ -VAEs, which still leverages deep auto-regressive networks and the training objective while enforcing a minimum KL-divergence between the variational posterior and prior. Zhu et al. (2020) demonstrated that considering the KL-divergence for the entire data set distribution instead of a single data point is enough to reduce posterior collapse by keeping a positive expectation. They additionally proposed a Batch-Normalized VAE to set a lower bound on the expectation.

## 6. Beyond KL-divergence

The KL-divergence offers a computationally convenient solution to measure the dissimilarity between two distributions. As discussed in Section 2.2, a closed-form solution for the forward KL-divergence is unavailable in the case of VI, and therefore, the reverse KL-divergence is used to formulate the VI objective function. The reverse KL-divergence, also known as I-projection or information projection (Murphy, 2013), in the case of amortized VI, is formulated as:

$$D_{\text{KL}}(q(z|x_i; \phi) \parallel p(z|x_i; \theta)) = \mathbb{E}_q \left[ \log \frac{q(z|x_i; \phi)}{p(z|x_i; \theta)} \right], \quad (25)$$

s.t.,

$$\lim_{p(z|x_i; \theta) \rightarrow 0} \frac{q(z|x_i; \phi)}{p(z|x_i; \theta)} \rightarrow \infty \quad \text{where} \quad q(z|x_i; \phi) > 0.$$

The limit indicates the need to force  $q(z|x_i; \phi) = 0$  wherever  $p(z|x_i; \theta) = 0$ , otherwise the KL-divergence would be very large (Ganguly & Earp, 2021). This *zero forcing* nature of the reverse KL-divergence has been proven to be useful in settings such as multi-modal posterior densities with unimodal approximations (Dieng et al., 2017). In such cases, the zero forcing nature helps to concentrate on one mode rather than spread mass all over them (Bishop, 2006b). However, zero forcing leads to an underestimate of the posterior variance (Dieng et al., 2017). In addition, it leads to degenerate solutions during optimization and is the source of *pruning* in VAEs (Dieng et al., 2017; Burda et al., 2016). As a result of these shortcomings, several other divergence measures have been proposed in recent years. In this section, we discuss a few of the relevant divergence measures and how they are used in the context of VI.

### 6.1 $\chi$ -divergence

Dieng et al. (2017) proposed CHIVI, a VI algorithm that minimizes the  $\chi$ -divergence between the variational approximation and the true posterior density. For amortized VI, the divergence measure is defined as:

$$\mathcal{D}_{\chi^r} = D_{\chi^r}(p(z|x_i; \theta) \parallel q(z|x_i; \phi)) = \mathbb{E}_{q_\phi} \left[ \left( \frac{p(z|x_i; \theta)}{q(z|x_i; \phi)} \right)^r - 1 \right], \quad (26)$$

where  $r$  is chosen depending on the application and data set.

Optimizing Equation 26 leads to a variational density with *zero avoiding* behavior like the forward KL-divergence (Murphy, 2013) or expectation propagation (EP) (Minka, 2005). This indicates that the  $\chi$ -divergence is infinite whenever  $q(z|x_i; \phi) = 0$  and  $p(z|x_i; \theta) > 0$  and thus, minimizing the  $\chi$ -divergence while setting  $p(z|x_i; \theta) > 0$  forces  $q(z|x_i; \phi) > 0$  (Dieng et al., 2017). Therefore,  $q$  avoids allocating zero mass at locations where  $p$  has non-zero mass. In contrast to VI optimization that uses KL-divergence as a means to maximize a lower bound on the model evidence, the main idea behind CHIVI is to optimize an upper bound which Dieng et al. (2017) refer to as the  $\chi$  *upper bound* (CUBO). Minimizing the CUBO is equivalent to minimizing the  $\chi$ -divergence (Dieng et al., 2017). The  $\chi$ -divergence objective function, for amortized VI, over  $N$  data points can be formulated as:

$$\begin{aligned} \prod_{i=1}^N \left( \mathcal{D}_{\chi^r} + 1 \right) &= \prod_{i=1}^N \mathbb{E}_{q_\phi} \left[ \left( \frac{p(z|x_i; \theta)}{q(z|x_i; \phi)} \right)^r \right], \\ \sum_{i=1}^N \log \left( \mathcal{D}_{\chi^r} + 1 \right) &= \sum_{i=1}^N \log \mathbb{E}_{q_\phi} \left[ \left( \frac{p(z|x_i; \theta)}{q(z|x_i; \phi)} \right)^r \right], \\ &= \sum_{i=1}^N \log \mathbb{E}_{q_\phi} \left[ \left( \frac{p(z, x_i; \theta)}{p(x_i; \theta)q(z|x_i; \phi)} \right)^r \right], \\ &= -r \sum_{i=1}^N \log p(x_i; \theta) + \sum_{i=1}^N \log \mathbb{E}_{q_\phi} \left[ \left( \frac{p(z, x_i; \theta)}{q(z|x_i; \phi)} \right)^r \right], \\ &= -r \log p(x; \theta) + \sum_{i=1}^N \log \mathbb{E}_{q_\phi} \left[ \left( \frac{p(z, x_i; \theta)}{q(z|x_i; \phi)} \right)^r \right], \end{aligned}$$

$$\begin{aligned}
 \log p(x; \theta) &= \frac{1}{r} \sum_{i=1}^N \log \mathbb{E}_{q_\phi} \left[ \left( \frac{p(z, x_i; \theta)}{q(z|x_i; \phi)} \right)^r \right] - \frac{1}{r} \sum_{i=1}^N \log \left( \mathcal{D}_{\chi^r} + 1 \right), \\
 &= \text{CUBO}_r - \frac{1}{r} \sum_{i=1}^N \log \left( \mathcal{D}_{\chi^r} + 1 \right), \tag{27}
 \end{aligned}$$

where

$$\text{CUBO}_r = \frac{1}{r} \sum_{i=1}^N \log \mathbb{E}_{q_\phi} \left[ \left( \frac{p(z, x_i; \theta)}{q(z|x_i; \phi)} \right)^r \right]$$

is a non-decreasing function of the order of the  $\chi$ -divergence  $\forall r \geq 1$ . By non-negativity of the  $\chi$ -divergence in Equation 27, an upper bound to the log-likelihood of data is established as:

$$\log p(x; \theta) \leq \text{CUBO}_r.$$

When  $r \geq 1$ ,  $\text{CUBO}_r$  is an upper bound to the model evidence enabling a higher precision approximation of  $\log p(x; \theta)$  as  $r$  approaches 1. Dieng et al. (2017) stated that the gap induced by  $\text{CUBO}_r$  and ELBO increases with  $r$ ; however, as  $r$  decreases to 0,  $\text{CUBO}_r$  becomes a lower bound as tends to the ELBO, i.e.,  $\lim_{r \rightarrow 0} \text{CUBO}_r = \text{ELBO}$ .

## 6.2 $\alpha$ -divergence

The KL-divergence is a special case of a family of divergence measures known as the  $\alpha$ -divergence. Different formulations of the  $\alpha$ -divergence exist (Amari, 2009; Zhu & Rohwer, 1995); however, we focus on Rényi's formulation, which defines the divergence measure for amortized VI as:

$$\mathcal{D}_\alpha = D_\alpha(q(z|x_i; \phi) \parallel p(z|x_i; \theta)) = \frac{1}{\alpha - 1} \log \int q(z|x_i; \phi)^\alpha p(z|x_i; \theta)^{1-\alpha} dz, \tag{28}$$

where  $\alpha \in [0, 1) \cup (1, \infty)$  and as  $\alpha \rightarrow 1$ , we recover the standard reverse KL-divergence for VI (van Erven & Harremos, 2014). A special case of  $\alpha = 2$  results in a measure that is proportional to the  $\chi^2$ -divergence.

Using  $\alpha$ -divergence, a bound on the marginal likelihood can be derived as:

$$\begin{aligned}
 \mathcal{L}_\alpha(x; \phi, \theta) &= \log p(x; \theta) - \frac{1}{\alpha - 1} \sum_{i=1}^N \mathcal{D}_\alpha, \\
 &= \sum_{i=1}^N \log p(x_i; \theta) - \frac{1}{\alpha - 1} \sum_{i=1}^N \mathcal{D}_\alpha, \\
 &= \sum_{i=1}^N \left[ \frac{\log p(x_i; \theta)^{\alpha-1}}{\alpha - 1} - \frac{\mathcal{D}_\alpha}{\alpha - 1} \right], \\
 &= \frac{1}{\alpha - 1} \sum_{i=1}^N \left[ -\log p(x_i; \theta)^{1-\alpha} - \mathcal{D}_\alpha \right], \\
 &= \frac{1}{1 - \alpha} \sum_{i=1}^N \left[ \log p(x_i; \theta)^{1-\alpha} + \log \int q(z|x_i; \phi)^\alpha p(z|x_i; \theta)^{1-\alpha} dz \right],
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{1-\alpha} \sum_{i=1}^N \left[ \log p(x_i; \theta)^{1-\alpha} \int q(z|x_i; \phi)^\alpha p(z|x_i; \theta)^{1-\alpha} dz \right], \\
 &= \frac{1}{1-\alpha} \sum_{i=1}^N \left[ \log \int q(z|x_i; \phi) p(x_i; \theta)^{1-\alpha} q(z|x_i; \phi)^{\alpha-1} p(z|x_i; \theta)^{1-\alpha} dz \right], \\
 &= \frac{1}{1-\alpha} \sum_{i=1}^N \log \mathbb{E}_{q_\phi} \left[ p(x_i; \theta)^{1-\alpha} q(z|x_i; \phi)^{\alpha-1} p(z|x_i; \theta)^{1-\alpha} \right], \\
 &= \frac{1}{1-\alpha} \sum_{i=1}^N \log \mathbb{E}_{q_\phi} \left[ q(z|x_i; \phi)^{\alpha-1} p(z, x_i; \theta)^{1-\alpha} \right], \\
 &= \frac{1}{1-\alpha} \sum_{i=1}^N \log \mathbb{E}_{q_\phi} \left[ \left( \frac{p(z, x_i; \theta)}{q(z|x_i; \phi)} \right)^{1-\alpha} \right]. \tag{29}
 \end{aligned}$$

The bound in Equation 29, also known as Variational Rényi (VR) bound (Li & Turner, 2016), can be extended to  $\alpha < 0$  and is continuous and non-increasing on  $\alpha \in \{\alpha : |\mathcal{L}_\alpha(x)| < +\infty\}$  (Li & Turner, 2016). Especially for all  $0 < \alpha_+ < 1$  and  $\alpha_- < 0$ ,

$$\mathcal{L}(x; \phi, \theta) < \mathcal{L}_{\alpha_+}(x; \phi, \theta) < \log p(x; \theta) < \mathcal{L}_{\alpha_-}(x; \phi, \theta),$$

indicating that the VR bound can be useful for model selection by sandwiching the marginal likelihood with bounds computed using positive and negative  $\alpha$  values. In their work, Li and Turner (2016) demonstrated how choosing different *alpha* values allows the variational approximation to balance between zero forcing ( $\alpha \rightarrow \infty$ ) and mass-covering ( $\alpha \rightarrow -\infty$ ) behavior.

$\alpha$ -divergences are a subset of a more general family of divergences known as  $f$ -divergences (Ali & Silvey, 1966), which for amortized VI can be formulated as:

$$D_f(q(z|x_i; \phi) \parallel p(z|x_i; \theta)) = \int p(z|x_i; \theta) f\left(\frac{q(z|x_i; \phi)}{p(z|x_i; \theta)}\right) dz, \tag{30}$$

where  $f$  is a convex function with  $f(1) = 0$ , and the reverse KL-divergence for the above formulation can be obtained by choosing the  $f$ -divergence as  $f(\omega) = \omega \log(\omega)$ . In general, only specific choices of  $f$  result in a bound that is trivially dependent on the marginal likelihood, and which is, therefore, useful for VI (Zhang et al., 2019).

### 6.3 Stein Discrepancy

Introduced by Stein (1972) as an error bound to measure how well an approximate distribution fits a distribution of interest, the Stein discrepancy as a divergence measure for amortized VI can be defined as:

$$D_{\text{stein}}(p(z|x_i; \theta) \parallel q(z|x_i; \phi)) = \sup_{f \in \mathcal{F}} \left[ \mathbb{E}_{q_\phi} \left[ f(z) \right] - \mathbb{E}_{p_\theta} \left[ f(z) \right] \right]^2, \tag{31}$$

where  $\mathcal{F}$  denotes a set of smooth, real-valued functions (Zhang et al., 2019). The smaller this divergence is, the more similar  $p$  and  $q$  are. When this divergence is zero, the two densities are identical.



The second term in Equation 31 involves taking expectations with respect to the intractable posterior. Therefore, in VI, the Stein discrepancy can only be used for classes of functions  $\mathcal{F}$  for which the second term is zero. A suitable class with this property can be defined by applying a differential operator  $\mathcal{A}$  on an arbitrary smooth function  $g$  as:

$$f(z) = \mathcal{A}g(z),$$

where  $z \sim p(z)$  and the operator  $\mathcal{A}$  are constructed in a way such that the second expectation in Equation 31 is zero. A popular choice for the operator that fulfills this requirement is the Stein operator, which is defined as:

$$\mathcal{A}g(z) = g(z)\nabla_z \log p(x_i, z; \theta) + \nabla_z g(z),$$

where  $\nabla_z \log p(x_i, z; \theta)$  is the score function, which can be calculated without knowing the normalization constant of  $p(x_i, z; \theta)$ . This indicates the Stein discrepancy’s robustness in handling model misspecifications in the form of unnormalized distributions (Liu & Wang, 2016).

Several research in VI have used the Stein discrepancy in recent years (Han & Liu, 2017; Liu et al., 2016; Liu & Wang, 2016; Liu et al., 2017). Of particular interest is the Stein Variational Gradient Descent (SVGD) methodology, proposed by Liu and Wang (2016), which makes use of the kernelized Stein discrepancy (KSD) (Liu et al., 2016) along with the Stein operator to construct the variational objective. With a specific choice of the kernel and  $q$ , KSD enables the SVGD algorithm to determine the optimal perturbation in the direction of the steepest gradient of the KL-divergence (Liu & Wang, 2016). Similar to the normalizing flow approach (Tabak & Turner, 2013; Tabak & Vanden-Eijnden, 2010; Rezende & Mohamed, 2015), SVGD leads to a scheme where samples in the latent space are sequentially transformed to approximate the true posterior distribution. Furthermore, the SVGD algorithm does not require explicitly specified parametric forms, allowing it to be flexible and easily implementable (Liu & Wang, 2016).

Moreover, Liu et al. (2016) demonstrated how the KSD acts as an unbiased statistic for measuring the goodness-of-fit test, which determines how likely it is that a set of given samples were generated from a target density function (Chwialkowski et al., 2016). Thus, in conjunction with being formulated as a gradient operator in SGVD, the Stein discrepancy can equivalently be expressed as a test function formulation for goodness-of-fit tests. This duality property of the Stein discrepancy is what makes it particularly useful in statistical inference and hypothesis testing. However, both KSD and SVGD in their original form can only be applied to continuous distributions. Hence, in their work, Han et al. (2020) developed the gradient-free versions of both KSD and SVGD for goodness-of-fit testing on discrete distributions and to sample from discrete-valued distributions while performing approximate inference respectively.

## 7. Open Problems

In this paper, we provide a mathematical foundation for amortized VI through studying and gaining an intuitive understanding of traditional-, stochastic-, and black box-VI, and the strengths and weaknesses of these methods. Additionally, we elucidate the recent advancements in the field of amortized VI, particularly in addressing its issues - sub-optimal

inference, inconsistent representation learning, generalization gap, and posterior collapse. Furthermore, we provide an overview of the various alternate divergence measures that can potentially replace the KL-divergence measure in the VI optimization process. Although the use of amortized VI in deep generative modeling has grown in recent years, the research to make it scalable, efficient, accurate, and easier to formulate is still ongoing. We outline some of the active research areas and open problems in the field of VI :

- **Amortized VI and Deep Learning.** With the recent advancements in the field of deep learning, researchers have successfully combined VI along with deep neural networks, in the form of VAEs, for generative modeling tasks. However, VAEs lack the ability to take into account the uncertainty in posterior approximation in a principled manner (Kim & Pavlovic, 2021). Recent research (e.g., Tomczak et al., 2021; Shridhar et al., 2018) has been aimed towards making the posterior approximation in VAEs more interpretable by using Bayesian neural networks (Neal, 1996) as the choice for the parametric functions for both the inference and generative models in VAEs.
- **VAE latent space geometry.** Generally, the distance between points in the latent space in a VAE does not reflect the true similarity of corresponding points in the observation space (Chen et al., 2018). Notable research in this area includes treating the latent space of VAEs as a Riemannian manifold (Chen et al., 2018). Iterating on this idea, Chen et al. (2020) developed the *flat manifold* VAE which defines the latent space as a Riemannian manifold and regularizes the Riemannian metric tensor to be a scaled identity matrix. This extension to vanilla VAEs allowed for learning flat latent manifolds, where the Euclidean distance is a proxy for the similarity between data points. Although there has been some progress to improve the representation learning in VAEs (see Section 5.2.2), the geometrical properties of the latent space in VAEs are not well understood.
- **Alternatives to the non-convex ELBO.** Another area of research is to address the non-convex nature of the ELBO. With the recent introduction of thermodynamic integration techniques (Lartillot & Philippe, 2006), researchers have paved the way for the development of a new VI framework that uses the Variational Hölder (VH) bound (Bouchard & Lakshminarayanan, 2015) as an alternative to the ELBO. Unlike the ELBO, the VH bound is a convex upper bound to the intractable evidence (Bouchard & Lakshminarayanan, 2015), the minimization of which is a convex optimization problem that can be solved using existing convex optimization algorithms. Additionally, the approximation error of the VH bound can also be analyzed using tools from convex optimization literature (Bouchard & Lakshminarayanan, 2015). Furthermore, promising work in this area by Chen et al. (2021) has shown to achieve a one-step approximation of the exact marginal log-likelihood using the VH bound.
- **Automatic VI.** Finally, the development of probabilistic programming tools, such as PyMC (Salvatier et al., 2016), Stan (Carpenter et al., 2017), Infer.Net (Minka et al., 2018), Zhusuan (Shi et al., 2017), have enabled researchers to automatize their experiment pipelines and thus allowing them to revise and improve their models with ease. Despite the progress in the development of these toolboxes, their usage is not straightforward to researchers new to the field.

## Acknowledgments

We are grateful to our colleagues Aubin Samacoits and Dhruba Pujary at Sertis Vision Lab for their constructive input and feedback during the writing of this paper.

## Appendix A. Derivation of Equation 9

$$\begin{aligned}\mathcal{L}(x; \xi_{1:N}, \theta) &= \sum_{i=1}^N \mathbb{E}_q \left[ \log p(x_i, z; \theta) - \log q(z|x_i; \xi_i) \right], \\ &= \sum_{i=1}^N \int \prod_k q_k \left[ \log p(x_i, z; \theta) - \log \prod_k q_k \right] dz.\end{aligned}$$

Now, we shall focus on the second term of the above equation as:

$$\begin{aligned}& \int \prod_k q_k \log \prod_k q_k dz \\ &= \int q_j \prod_{k \neq j} q_k \log \left[ q_j \prod_{k \neq j} q_k \right] dz, \\ &= \int q_j \prod_{k \neq j} q_k \left[ \log q_j + \log \prod_{k \neq j} q_k \right] dz, \\ &= \int \left[ q_j \log q_j \prod_{k \neq j} q_k + q \log \prod_{k \neq j} q_k \right] dz, \\ &= \int \int \left[ q_j \log q_j \prod_{k \neq j} q_k + q \log \prod_{k \neq j} q_k \right] dz_j dz_{k \neq j}, \\ &= \int q_j \log q_j \left[ \int \prod_{k \neq j} q_k dz_{k \neq j} \right] dz_j + \int \int q \log \prod_{k \neq j} q_k dz_j dz_{k \neq j}, \\ &= \int q_j \log q_j dz_j + \int \prod_{k \neq j} q_k \log \prod_{k \neq j} q_k \int q_j dz_j dz_{k \neq j}, \\ &= \int q_j \log q_j dz_j + \int \prod_{k \neq j} q_k \log \prod_{k \neq j} q_k dz_{k \neq j}, \\ &= \int q_j \log q_j(z_j) dz_j + \mathcal{H}_{k \neq j},\end{aligned}$$

where

$$\mathcal{H}_{k \neq j} = \int \prod_{k \neq j} q_k \log \prod_{k \neq j} q_k dz_{k \neq j},$$

which is the entropy of all the factorized probability densities  $k \neq j$ .

## Appendix B. Fisher and Symmetrized KL-divergence

Given a probability density function  $q(z; \xi)$ , the symmetrized KL-divergence captures the movement of a distance of  $\Delta\xi$  in the direction of the steepest ascent as the dissimilarity of the probability densities  $q(z; \xi)$  and  $q(z; \xi + \Delta\xi)$ , and is formulated as:

$$D_{\text{KL}}^{\text{sym}}(\xi, \xi + \Delta\xi) = \mathbb{E}_{q(z; \xi)} \left[ \log \frac{q(z; \xi)}{q(z; \xi + \Delta\xi)} \right] + \mathbb{E}_{q(z; \xi + \Delta\xi)} \left[ \log \frac{q(z; \xi + \Delta\xi)}{q(z; \xi)} \right]. \quad (33)$$

Additionally, the second order Taylor expansion for a function  $f(\xi)$  at a point  $\xi_i$  is given by:

$$f(\xi) \approx f(\xi_i) + \nabla_{\xi} f(\xi_i)^T (\xi - \xi_i) + \frac{1}{2} (\xi - \xi_i)^T \nabla_{\xi}^2 f(\xi_i) (\xi - \xi_i). \quad (34)$$

Substituting  $\xi = \xi_i + \Delta\xi$  (such that  $\Delta\xi \rightarrow 0$ ) in Equation 34, we get:

$$f(\xi_i + \Delta\xi) = f(\xi_i) + \nabla_{\xi} f(\xi_i)^T \Delta\xi + \frac{1}{2} \Delta\xi^T \nabla_{\xi}^2 f(\xi_i) \Delta\xi. \quad (35)$$

Now, using the result in Equation 35 we can expand the terms  $\log q(z; \xi + \Delta\xi)$  from the right hand side of Equation 33 as follows:

$$\log q(z; \xi + \Delta\xi) = \log q(z; \xi) + \{\nabla_{\xi} \log q(z; \xi)\}^T \Delta\xi + \frac{1}{2} \Delta\xi^T \nabla_{\xi}^2 \log q(z; \xi) \Delta\xi. \quad (36)$$

Using the result from Equation 36 we expand the first term on the right-hand side of Equation 33 as:

$$\begin{aligned} \log \frac{q(z; \xi)}{q(z; \xi + \Delta\xi)} &= \log q(z; \xi) - \log q(z; \xi + \Delta\xi) \\ &= -\{\nabla_{\xi} \log q(z; \xi)\}^T \Delta\xi - \frac{1}{2} \Delta\xi^T \nabla_{\xi}^2 \log q(z; \xi) \Delta\xi, \\ &= -\frac{\{\nabla_{\xi} q(z; \xi)\}^T \Delta\xi}{q(z; \xi)} - \frac{1}{2} \Delta\xi^T \nabla_{\xi} \left\{ \frac{\nabla_{\xi} q(z; \xi)}{q(z; \xi)} \right\} \Delta\xi, \\ &= -\frac{\{\nabla_{\xi} q(z; \xi)\}^T \Delta\xi}{q(z; \xi)} - \frac{1}{2} \Delta\xi^T \left\{ \frac{q(z; \xi) \nabla_{\xi}^2 q(z; \xi)}{q(z; \xi) q(z; \xi)} \right. \\ &\quad \left. - \frac{\nabla_{\xi} q(z; \xi) \nabla_{\xi} q(z; \xi)^T}{q(z; \xi) q(z; \xi)} \right\} \Delta\xi, \\ &= -\frac{\{\nabla_{\xi} q(z; \xi)\}^T \Delta\xi}{q(z; \xi)} - \frac{1}{2} \Delta\xi^T \left\{ \frac{\nabla_{\xi}^2 q(z; \xi)}{q(z; \xi)} \right\} \Delta\xi \\ &\quad + \frac{1}{2} \Delta\xi^T \left\{ \nabla_{\xi} \log q(z|x; \xi) \nabla_{\xi} \log q(z|x; \xi)^T \right\} \Delta\xi. \end{aligned} \quad (37)$$

Taking expectations with respect to  $q(z; \xi)$  on both sides of Equation 37, we get:

$$\begin{aligned} \mathbb{E}_{q(z; \xi)} \left[ \log \frac{q(z; \xi)}{q(z; \xi + \Delta\xi)} \right] &= -\mathbb{E}_{q(z; \xi)} \left[ \frac{\{\nabla_{\xi} q(z; \xi)\}^T \Delta\xi}{q(z; \xi)} \right] - \mathbb{E}_{q(z; \xi)} \left[ \frac{1}{2} \Delta\xi^T \left\{ \frac{\nabla_{\xi}^2 q(z; \xi)}{q(z; \xi)} \right\} \Delta\xi \right] \\ &\quad + \mathbb{E}_{q(z; \xi)} \left[ \frac{1}{2} \Delta\xi^T \left\{ \nabla_{\xi} \log q(z|x; \xi) \nabla_{\xi} \log q(z|x; \xi)^T \right\} \Delta\xi \right], \end{aligned}$$

$$\begin{aligned}
 &= - \left[ \int q(z; \xi) \frac{\{\nabla_{\xi} q(z; \xi)\}^T}{q(z; \xi)} dz \right] \Delta \xi \\
 &\quad - \frac{1}{2} \Delta \xi^T \left[ \int q(z; \xi) \left\{ \frac{\nabla_{\xi}^2 q(z; \xi)}{q(z; \xi)} dz \right\} \right] \Delta \xi \\
 &\quad + \frac{1}{2} \Delta \xi^T \left\{ \mathbb{E}_{q(z; \xi)} \left[ \nabla_{\xi} \log q(z|x; \xi) \nabla_{\xi} \log q(z|x; \xi)^T \right] \right\} \Delta \xi. \quad (38)
 \end{aligned}$$

The terms on the right-hand side of Equation 38 can each be evaluated as:

$$\begin{aligned}
 \left[ \int q(z; \xi) \frac{\{\nabla_{\xi} q(z; \xi)\}^T}{q(z; \xi)} dz \right] \Delta \xi &= \left[ \int \{\nabla_{\xi} q(z; \xi)\}^T dz \right] \Delta \xi \\
 &= \left[ \nabla_{\xi} \int q(z; \xi) dz \right]^T \Delta \xi \\
 &= \left[ \nabla_{\xi} 1 \right]^T \Delta \xi \\
 &= 0, \quad (39)
 \end{aligned}$$

$$\begin{aligned}
 \frac{1}{2} \Delta \xi^T \left[ \int q(z; \xi) \left\{ \frac{\nabla_{\xi}^2 q(z; \xi)}{q(z; \xi)} dz \right\} \right] \Delta \xi &= \frac{1}{2} \Delta \xi^T \left[ \int \nabla_{\xi}^2 q(z; \xi) dz \right] \Delta \xi \\
 &= \frac{1}{2} \Delta \xi^T \left[ \nabla_{\xi}^2 \left\{ \int q(z; \xi) dz \right\} \right] \Delta \xi \\
 &= \frac{1}{2} \Delta \xi^T \left[ \nabla_{\xi}^2 1 \right] \Delta \xi \\
 &= 0, \quad (40)
 \end{aligned}$$

and

$$\frac{1}{2} \Delta \xi^T \left\{ \mathbb{E}_{q(z; \xi)} \left[ \nabla_{\xi} \log q(z|x; \xi) \nabla_{\xi} \log q(z|x; \xi)^T \right] \right\} \Delta \xi = \frac{1}{2} \Delta \xi^T I(\xi) \Delta \xi, \quad (41)$$

where  $I(\xi)$  is the Fisher Information matrix.

Thus substituting the results from Equations 39, 40, and 41 in Equation 38, we get:

$$\mathbb{E}_{q(z; \xi)} \left[ \log \frac{q(z; \xi)}{q(z; \xi + \Delta \xi)} \right] = \frac{1}{2} \Delta \xi^T I(\xi) \Delta \xi. \quad (42)$$

As  $\Delta \xi \rightarrow 0$ , therefore,  $q(z; \xi)$  and  $q(z; \xi + \Delta \xi)$  are the same probability density. Thus, expanding the second term on the right-hand side of Equation 33, in a similar manner, we can show that:

$$\mathbb{E}_{q(z; \xi + \Delta \xi)} \left[ \log \frac{q(z; \xi + \Delta \xi)}{q(z; \xi)} \right] = \frac{1}{2} \Delta \xi^T I(\xi) \Delta \xi. \quad (43)$$

Combining the results from Equations 42 and 43 in Equation 33 as follows:

$$\begin{aligned}
 D_{\text{KL}}^{\text{sym}}(\xi, \xi + \Delta \xi) &\approx \frac{1}{2} \Delta \xi^T I(\xi) \Delta \xi + \frac{1}{2} \Delta \xi^T I(\xi) \Delta \xi \\
 &\approx \Delta \xi^T I(\xi) \Delta \xi, \quad (44)
 \end{aligned}$$

which corresponds to computing the inner product of a vector with itself in the Riemannian manifold (Chen et al., 2018).

## Appendix C. Fisher and Hessian

Given a function  $f(x)$ , its Jacobian  $\mathbb{J}[f(x)]$  and Hessian  $H[f(x)]$  are computed as

$$H[f(x)] = \mathbb{J}[\nabla f(x)].$$

For a probability density  $q(z|x_i; \xi_i)$ , the Hessian for  $\log q(z|x_i; \xi_i)$  is given by:

$$\begin{aligned} H[\log q(z|x_i; \xi_i)] &= \mathbb{J} \left[ \nabla_{\xi_i} \log q(z|x_i; \xi_i) \right], \\ &= \mathbb{J} \left[ \frac{\nabla_{\xi_i} q(z|x_i; \xi_i)}{q(z|x_i; \xi_i)} \right], \\ &= \nabla_{\xi_i} \left[ \frac{\nabla_{\xi_i} q(z|x_i; \xi_i)}{q(z|x_i; \xi_i)} \right], \\ &= \frac{q(z|x_i; \xi_i) H[q(z|x_i; \xi_i)] - \nabla_{\xi_i} q(z|x_i; \xi_i) \nabla_{\xi_i} q(z|x_i; \xi_i)^T}{q(z|x_i; \xi_i) q(z|x_i; \xi_i)}, \\ &= \frac{H[q(z|x_i; \xi_i)]}{q(z|x_i; \xi_i)} - \left( \frac{\nabla_{\xi_i} q(z|x_i; \xi_i)}{q(z|x_i; \xi_i)} \right) \left( \frac{\nabla_{\xi_i} q(z|x_i; \xi_i)}{q(z|x_i; \xi_i)} \right)^T, \\ &= \frac{H[q(z|x_i; \xi_i)]}{q(z|x_i; \xi_i)} - \nabla_{\xi_i} \log q(z|x_i; \xi_i) \nabla_{\xi_i} \log q(z|x_i; \xi_i)^T. \end{aligned}$$

Therefore, taking expectations with respect to  $q(z|x_i; \xi_i)$ , we get:

$$\begin{aligned} \mathbb{E}_q \left[ H[\log q(z|x_i; \xi_i)] \right] &= \mathbb{E}_q \left[ \frac{H[q(z|x_i; \xi_i)]}{q(z|x_i; \xi_i)} \right] - \mathbb{E}_q \left[ \nabla_{\xi_i} \log q(z|x_i; \xi_i) \nabla_{\xi_i} \log q(z|x_i; \xi_i)^T \right], \\ &= \int \frac{H[q(z|x_i; \xi_i)]}{q(z|x_i; \xi_i)} q(z|x_i; \xi_i) dz - I(\xi_i), \\ &= \int H[q(z|x_i; \xi_i)] dz - I(\xi_i), \\ &= \mathbb{J} \left[ \int \nabla_{\xi_i} q(z|x_i; \xi_i) dz \right] - I(\xi_i), \\ &= \mathbb{J} \left[ \nabla_{\xi_i} \int q(z|x_i; \xi_i) dz \right] - I(\xi_i), \\ &= \mathbb{J} \left[ \nabla_{\xi_i} 1 \right] - I(\xi_i), \\ &= -I(\xi_i), \\ I(\xi_i) &= -\mathbb{E}_q \left[ H[\log q(z|x_i; \xi_i)] \right]. \end{aligned}$$

## References

Airoldi, E. M. (2007). Getting started in probabilistic graphical models. *PLoS Computational Biology*, 3(12).

- Ali, S. M., & Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(1), 131–142.
- Amari, S.-I. (1982). Differential geometry of curved exponential families—curvatures and information loss. *The Annals of Statistics*, 10(2).
- Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2), 251–276.
- Amari, S.-I. (2009).  $\alpha$ -divergence is unique, belonging to both f-divergence and Bregman divergence classes. *IEEE Transactions on Information Theory*, 55(11), 4925–4931.
- Amos, B. (2023). Tutorial on amortized optimization. *Foundations and Trends® in Machine Learning*, 16(5), 592–732.
- Barber, D., & Wiering, W. (1998). Tractable variational structures for approximating graphical models. In Kearns, M., Solla, S., & Cohn, D. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 11. MIT Press.
- Bhattacharyya, C., & Keerthi, S. S. (2001). Mean field methods for a special class of belief networks. *Journal of Artificial Intelligence Research*, 15, 91–114.
- Bishop, C. M. (2006a). *Pattern Recognition and Machine Learning (Information Science and Statistics)*, chap. Chapter 8: Graphical Models. Springer-Verlag, Berlin, Heidelberg.
- Bishop, C. M. (2006b). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.
- Blei, D. M. (2012). Probabilistic topic models. *Commun. ACM*, 55(4), 77–84.
- Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518), 859–877.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(null), 993–1022.
- Bouchard, G., & Lakshminarayanan, B. (2015). Approximate inference with the variational hölder bound. arXiv preprint arXiv:1506.06100.
- Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge.
- Boyle, P. P. (1977). Options: A Monte Carlo approach. *Journal of Financial Economics*, 4(3), 323–338.
- Brooks, S., Gelman, A., Jones, G., & Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo*. CRC Press, Boca Raton, FL.
- Buchholz, A., Wenzel, F., & Mandt, S. (2018). Quasi-Monte Carlo variational inference. In *International Conference on Machine Learning*, pp. 668–677. PMLR.
- Burda, Y., Grosse, R. B., & Salakhutdinov, R. (2016). Importance weighted autoencoders. In Bengio, Y., & LeCun, Y. (Eds.), *International Conference on Learning Representations (ICLR)*.
- Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., & Lerchner, A. (2018). Understanding disentangling in  $\beta$ -VAE. arXiv preprint arXiv:1804.03599.

- Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, *76*(1), 1–32.
- Cemgil, T., Ghaisas, S., Dvijotham, K., Gowal, S., & Kohli, P. (2020). The autoencoding variational autoencoder. *Advances in Neural Information Processing Systems*, *33*, 15077–15087.
- Chen, J., Lu, D., Xiu, Z., Bai, K., Carin, L., & Tao, C. (2021). Variational inference with Hölder bounds. arXiv preprint arXiv:2111.02947.
- Chen, N., Klushyn, A., Ferroni, F., Bayer, J., & Van Der Smagt, P. (2020). Learning flat latent manifolds with VAEs. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.
- Chen, N., Klushyn, A., Kurlle, R., Jiang, X., Bayer, J., & Smagt, P. (2018). Metrics for deep generative models. In *International Conference on Artificial Intelligence and Statistics*, pp. 1540–1550. PMLR.
- Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2180–2188.
- Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., & Abbeel, P. (2017). Variational lossy autoencoder. In *International Conference on Learning Representations*.
- Chwialkowski, K., Strathmann, H., & Gretton, A. (2016). A kernel test of goodness of fit. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, p. 2606–2615. JMLR.org.
- Cox, D. R., & Hinkley, D. V. (1994). *Theoretical Statistics*. Chapman and Hall, London.
- Cremer, C., Li, X., & Duvenaud, D. (2018). Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pp. 1078–1086. PMLR.
- Csiba, D., & Richtárik, P. (2018). Importance sampling for minibatches. *Journal of Machine Learning Research*, *19*(1), 962–982.
- Dagum, P., & Luby, M. (1993). Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, *60*(1), 141–153.
- Dieng, A. B., Tran, D., Ranganath, R., Paisley, J., & Blei, D. (2017). Variational inference via  $\chi$  upper bound minimization. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.
- do Carmo, M. P. (1993). *Riemannian Geometry*. Birkhäuser, Boston, MA.
- Duchi, J. C., Hazan, E., & Singer, Y. (2010). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*, 2121–2159.



- Eikema, B., & Aziz, W. (2019). Auto-encoding variational neural machine translation. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*.
- Foti, N., Xu, J., Laird, D., & Fox, E. (2014). Stochastic variational inference for hidden Markov models. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., & Weinberger, K. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc.
- Fu, T., & Zhang, Z. (2017). CPSG-MCMC: Clustering-based preprocessing method for stochastic gradient MCMC. In Singh, A., & Zhu, X. J. (Eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Vol. 54 of *Proceedings of Machine Learning Research*, pp. 841–850. PMLR.
- Gagliardi Cozman, F. (2000). Generalizing variable elimination in Bayesian networks. In *Workshop on Probabilistic Reasoning in Artificial Intelligence*, pp. 27–32.
- Ganguly, A., & Earp, S. W. (2021). An introduction to variational inference. arXiv preprint arXiv:2108.13083.
- Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-6*(6), 721–741.
- Gershman, S., & Goodman, N. (2014). Amortized inference in probabilistic reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 36.
- Gershman, S. J., Hoffman, M. D., & Blei, D. M. (2012). Nonparametric variational inference. In *Proceedings of the 29th International Conference on Machine Learning, ICML’12*, p. 235–242, Madison, WI, USA. Omnipress.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc.
- Gopalan, P., Mimno, D., Gerrish, S., Freedman, M., & Blei, D. (2012). Scalable inference of overlapping communities. In *Advances in Neural Information Processing Systems*, Vol. 25.
- Han, J., Ding, F., Liu, X., Torresani, L., Peng, J., & Liu, Q. (2020). Stein variational inference for discrete distributions. In Chiappa, S., & Calandra, R. (Eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Vol. 108 of *Proceedings of Machine Learning Research*, pp. 4563–4572. PMLR.
- Han, J., & Liu, Q. (2017). Stein variational adaptive importance sampling. In Elidan, G., Kersting, K., & Ihler, A. T. (Eds.), *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*. AUAI Press.
- Haußmann, M., Hamprecht, F. A., & Kandemir, M. (2020). Sampling-free variational inference of Bayesian neural networks by variance backpropagation. In *Uncertainty in Artificial Intelligence*, pp. 563–573. PMLR.

- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., & Lerchner, A. (2017). beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.
- Hinton, G., Srivastava, N., & Swersky, K. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *Neural Networks for Machine Learning*, University of Toronto.
- Hjelm, D., Salakhutdinov, R. R., Cho, K., Jojic, N., Calhoun, V., & Chung, J. (2016). Iterative refinement of the approximate posterior for directed belief networks. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 29. Curran Associates, Inc.
- Hoffman, M., Bach, F., & Blei, D. (2010). Online learning for latent dirichlet allocation. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., & Culotta, A. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 23. Curran Associates, Inc.
- Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14, 1303–1347.
- Im, D. J., Ahn, S., Memisevic, R., & Bengio, Y. (2017). Denoising criterion for variational auto-encoding framework. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI’17*, p. 2059–2065. AAAI Press.
- Jaakkola, T. S., & Jordan, M. I. (1999). Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research*, 10, 291–322.
- Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations (ICLR)*.
- Jordan, M. I. (2004). Graphical models. *Statistical Science*, 19(1), 140 – 155.
- Jordan, M. I., Ghahramani, Z., & Jaakkola, Tommi S. and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2), 183–233.
- Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410.
- Khan, M. E., & Nielsen, D. (2018). Fast yet simple natural-gradient descent for variational inference in complex models. In *International Symposium on Information Theory and Its Applications, ISITA 2018, Singapore, October 28-31, 2018*, pp. 31–35. IEEE.
- Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., & Srivastava, A. (2018). Fast and scalable Bayesian deep learning by weight-perturbation in Adam. In Dy, J., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 2611–2620. PMLR.
- Kim, M., & Pavlovic, V. (2021). Reducing the amortization gap in variational autoencoders: A Bayesian random function approach. arXiv preprint arXiv:2102.03151.
- Kim, Y., Wiseman, S., Miller, A. C., Sontag, D., & Rush, A. M. (2018). Semi-amortized variational autoencoders. In Dy, J., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 2678–2687.

- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y., & LeCun, Y. (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. In Bengio, Y., & LeCun, Y. (Eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Klaričić Bakula, M., Matić, M., & Pečarić, J. (2008). On some general inequalities related to Jensen’s inequality. In Bandle, C., Losonczi, L., Gilányi, A., Páles, Z., & Plum, M. (Eds.), *Inequalities and Applications*, pp. 233–243.
- Knollmüller, J., & Enßlin, T. A. (2019). Metric Gaussian variational inference. arXiv preprint arXiv:1901.11033.
- Knowles, D. A. (2015). Stochastic gradient variational Bayes for gamma approximating distributions. arXiv preprint arXiv:1509.01631.
- Krishnan, R., Liang, D., & Hoffman, M. (2018). On the challenges of learning with inference networks on sparse, high-dimensional data. In Storkey, A., & Perez-Cruz, F. (Eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, Vol. 84 of *Proceedings of Machine Learning Research*, pp. 143–151. PMLR.
- Kschischang, F. R., Frey, B. J., & Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47, 498–519.
- Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1), 79–86.
- Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266), 1332–1338.
- Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (2016). Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, pp. 1558–1566. PMLR.
- Lartillot, N., & Philippe, H. (2006). Computing Bayes factors using thermodynamic integration. *Systematic Biology*, 55(2), 195–207.
- Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., & Teh, Y. W. (2019a). Set Transformer: A framework for attention-based permutation-invariant neural networks. In Chaudhuri, K., & Salakhutdinov, R. (Eds.), *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR.
- Lee, J., Lee, Y., & Teh, Y. W. (2019b). Deep amortized clustering. arXiv preprint arXiv:1909.13433.
- Li, X., & She, J. (2017). Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge*

- Discovery and Data Mining*, KDD '17, p. 305–314, New York, NY, USA. Association for Computing Machinery.
- Li, Y., & Turner, R. E. (2016). Rényi divergence variational inference. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 29. Curran Associates, Inc.
- Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. (2018). Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, p. 689–698, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Liu, H., Wang, J., & Jing, L. (2021). Cluster-wise hierarchical generative model for deep amortized clustering. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15104–15113.
- Liu, H., Zhou, T., & Wang, J. (2022). Deep amortized relational model with group-wise hierarchical generative process. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36, pp. 7550–7557.
- Liu, Q., Lee, J., & Jordan, M. (2016). A kernelized stein discrepancy for goodness-of-fit tests. In Balcan, M. F., & Weinberger, K. Q. (Eds.), *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48 of *Proceedings of Machine Learning Research*, pp. 276–284, New York, New York, USA. PMLR.
- Liu, Q., & Wang, D. (2016). Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, p. 2378–2386. Curran Associates Inc.
- Liu, Y., Ramachandran, P., Liu, Q., & Peng, J. (2017). Stein variational policy gradient. In Elidan, G., Kersting, K., & Ihler, A. (Eds.), *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press.
- Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.
- Lucas, J., Tucker, G., Grosse, R., & Norouzi, M. (2019). Understanding posterior collapse in generative latent variable models. In *International Conference on Learning Representations Workshop*.
- MacKay, D. J. C. (2002). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge.
- Maddison, C. J., Mnih, A., & Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*.
- Madsen, A. L., & Jensen, F. V. (1999). LAZY propagation: A junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, 113(1-2), 203–245.
- Marino, D. L., & Manic, M. (2021). Physics enhanced data-driven models with variational Gaussian processes. *IEEE Open Journal of the Industrial Electronics Society*, 2, 252–265.

- Marino, J., Yue, Y., & Mandt, S. (2018). Iterative amortized inference. In *International Conference on Machine Learning*, pp. 3403–3412. PMLR.
- Marlin, B., Swersky, K., Chen, B., & Freitas, N. (2010). Inductive principles for restricted Boltzmann machine learning. In Teh, Y. W., & Titterton, M. (Eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Vol. 9 of *Proceedings of Machine Learning Research*, pp. 509–516, Chia Laguna Resort, Sardinia, Italy. PMLR.
- Martens, J. (2020). New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146), 1–76.
- Merberg, A., & Miller, S. J. (2008). The Cramér-Rao inequality. Course Notes for Math 162: Mathematical Statistics, William College.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6), 1087–1092.
- Michie, D. (1968). “Memo” functions and machine learning. *Nature*, 218(5136), 19–22.
- Miller, A., Foti, N., D’Amour, A., & Adams, R. P. (2017). Reducing reparameterization gradient variance. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.
- Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI’01, p. 362–369, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Minka, T. (2005). Divergence measures and message passing. Tech. rep. MSR-TR-2005-173, Microsoft Research.
- Minka, T., Winn, J. M., Guiver, J. P., Zaykov, Y., Fabian, D., & Bronskill, J. (2018). Infer.NET 0.3.. Microsoft Research Cambridge. <http://dotnet.github.io/infer>.
- Mohamed, S., Rosca, M., Figurnov, M., & Mnih, A. (2020). Monte Carlo gradient estimation in machine learning.. *Journal of Machine Learning Research*, 21(132), 1–62.
- Müller, T., Rousselle, F., Keller, A., & Novák, J. (2020). Neural control variates. *ACM Transactions on Graphics*, 39(6).
- Murphy, K. P. (2013). *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA.
- Murphy, K. P., Weiss, Y., & Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI’99, p. 467–475, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Nalisnick, E. T., & Smyth, P. (2017). Stick-breaking variational autoencoders. In *International Conference on Learning Representations (ICLR)*.
- Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer-Verlag, New York, NY.

- Nowozin, S. (2018). Debiasing evidence approximations: On importance-weighted autoencoders and jackknife variational inference. In *International Conference on Learning Representations (ICLR)*.
- Opper, M., & Saad, D. (2001). *Advanced Mean Field Methods: Theory and Practice*. The MIT Press, Cambridge, MA.
- Parisi, G., & Shankar, R. (1988). *Statistical Field Theory*. Westview Press.
- Pidhorskyi, S., Adjeroh, D. A., & Doretto, G. (2020). Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14104–14113.
- Plummer, S., Pati, D., & Bhattacharya, A. (2020). Dynamics of coordinate ascent variational inference: A case study in 2D Ising models. *Entropy*, 22(11), 1263.
- Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In Bengio, Y., & LeCun, Y. (Eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Rainforth, T., Kosior, A., Le, T. A., Maddison, C., Igl, M., Wood, F., & Teh, Y. W. (2018). Tighter variational bounds are not necessarily better. In Dy, J., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 4277–4285. PMLR.
- Ranganath, R., Gerrish, S., & Blei, D. (2014). Black box variational inference. In *Artificial Intelligence and Statistics*, pp. 814–822. PMLR.
- Ranganath, R., Tran, D., & Blei, D. M. (2016). Hierarchical variational models. In Balcan, M. F., & Weinberger, K. Q. (Eds.), *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48 of *Proceedings of Machine Learning Research*, pp. 324–333, New York, New York, USA. PMLR.
- Razavi, A., van den Oord, A., Poole, B., & Vinyals, O. (2019). Preventing posterior collapse with delta-VAEs. In *International Conference on Learning Representations*.
- Regier, J., Miller, A., McAuliffe, J., Adams, R., Hoffman, M., Lang, D., Schlegel, D., & Prabhat, M. (2015). Celeste: Variational inference for a generative model of astronomical images. In *International Conference on Machine Learning*, pp. 2095–2103. PMLR.
- Rezende, D., & Mohamed, S. (2015). Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538. PMLR.
- Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, Vol. 32 of *JMLR Workshop and Conference Proceedings*, pp. 1278–1286. JMLR.org.
- Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3), 400–407.
- Ross, S. M. (2006). *Simulation, Fourth Edition*. Academic Press, Inc., Orlando, FL.

- Ruiz, F. J. R., Titsias, M. K., & Blei, D. M. (2016). Overdispersed black-box variational inference. In *UAI'16: Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 647—656.
- Salakhutdinov, R., & Murray, I. (2008). On the quantitative analysis of deep belief networks. In *Proceedings of the 25th international conference on Machine learning*, pp. 872–879. ACM.
- Salimans, T., & Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, p. 901–909. Curran Associates Inc.
- Salvatier, J., Wiecki, T. V., & Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2, e55.
- Saul, L. K., Jaakkola, T., & Jordan, M. I. (1996). Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4, 61–76.
- Schraudolph, N. N. (2002). Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7), 1723–1738.
- Semeniuta, S., Severyn, A., & Barth, E. (2017). A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 627–637, Copenhagen, Denmark. Association for Computational Linguistics.
- Shen, G., Chen, X., & Deng, Z. (2020). Variational learning of Bayesian neural networks via Bayesian dark knowledge. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*.
- Shi, J., Chen, J., Zhu, J., Sun, S., Luo, Y., Gu, Y., & Zhou, Y. (2017). ZhuSuan: A library for Bayesian deep learning. arXiv preprint arXiv:1709.05870.
- Shridhar, K., Laumann, F., Maurin, A. L., Olsen, M. A., & Liwicki, M. (2018). Bayesian convolutional neural networks with variational inference. arXiv preprint arXiv:1806.05978v5.
- Shu, R., Bui, H. H., Zhao, S., Kochenderfer, M. J., & Ermon, S. (2018). Amortized inference regularization. *Advances in Neural Information Processing Systems*, 31.
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, pp. 1–8.
- Sinha, S., & Dieng, A. B. (2021). Consistency regularization for variational auto-encoders. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., & Vaughan, J. W. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 34, pp. 12943–12954. Curran Associates, Inc.
- Smith, J. D., Ross, Z. E., Azizzadenesheli, K., & Muir, J. B. (2021). HypoSVI: Hypocentre inversion with Stein variational inference and physics informed neural networks. *Geophysical Journal International*, 228(1), 698–710.

- Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., & Winther, O. (2016). Ladder variational autoencoders. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, p. 3745–3753. Curran Associates Inc.
- Stein, C. (1972). A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, Vol. 6, pp. 583–603. University of California Press.
- Su, J. (2018). Variational inference: A unified framework of generative models and some revelations. arXiv preprint arXiv:1807.05936.
- Sun, S., Zhang, G., Shi, J., & Grosse, R. (2019). Functional variational Bayesian neural networks. In *International Conference on Learning Representations*.
- Sutton, R. S., Mcallester, D., Singh, S., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pp. 1057–1063. MIT Press.
- Tabak, E., & Vanden-Eijnden, E. (2010). Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8.
- Tabak, E. G., & Turner, C. V. (2013). A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2), 145–164.
- Takida, Y., Liao, W.-H., Lai, C.-H., Uesaka, T., Takahashi, S., & Mitsufuji, Y. (2021). Preventing oversmoothing in VAE via generalized variance parameterization. *Neurocomputing*, 509, 137–156.
- Thin, A., Kotelevskii, N., Doucet, A., Durmus, A., Moulines, E., & Panov, M. (2021). Monte carlo variational auto-encoders. In *International Conference on Machine Learning*, pp. 10247–10257. PMLR.
- Titsias, M., & Lázaro-Gredilla, M. (2015). Local expectation gradients for black box variational inference. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., & Garnett, R. (Eds.), *NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems*, pp. 2638—2646.
- Tolstikhin, I., Bousquet, O., Gelly, S., & Schoelkopf, B. (2018). Wasserstein auto-encoders. In *International Conference on Learning Representations*.
- Tomczak, M., Swaroop, S., Foong, A., & Turner, R. (2021). Collapsed variational bounds for Bayesian neural networks. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., & Vaughan, J. W. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 34, pp. 25412–25426. Curran Associates, Inc.
- Tsamardinos, I., Aliferis, C. F., & Statnikov, A. R. (2003). Algorithms for large scale markov blanket discovery. In Russell, I., & Haller, S. M. (Eds.), *FLAIRS Conference*, pp. 376–381. AAAI Press.
- Vahdat, A., & Kautz, J. (2020). NVAE: A deep hierarchical variational autoencoder. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20. Curran Associates Inc.



- van Erven, T., & Harremos, P. (2014). R enyi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7), 3797–3820.
- Wainwright, M. J., & Jordan, M. I. (2007). Graphical models, exponential families, and variational inference. *Foundations and Trends<sup>®</sup> in Machine Learning*, 1(1–2), 1–305.
- Wang, C., Chen, X., Smola, A. J., & Xing, E. P. (2013). Variance reduction for stochastic gradient optimization. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., & Weinberger, K. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 26. Curran Associates, Inc.
- Winn, J., & Bishop, C. (2005). Variational message passing. *Journal of Machine Learning Research*, 6, 661–694.
- Yang, H., & Amari, S.-i. (1997). The efficiency and the robustness of natural gradient descent learning rule. *Advances in Neural Information Processing Systems*, 10.
- Yang, Z., Hu, Z., Salakhutdinov, R., & Berg-Kirkpatrick, T. (2017). Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML’17*, p. 3881–3890. JMLR.org.
- Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. arXiv preprint arXiv:1212.5701.
- Zhang, C. (2016). *Structured representation using latent variable models*. Ph.D. thesis, KTH Royal Institute of Technology.
- Zhang, C., Butepage, J., Kjellstrom, H., & Mandt, S. (2019). Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8), 2008–2026.
- Zhang, M., Hayes, P., & Barber, D. (2022). Generalization gap in amortized inference. In *NeurIPS*.
- Zhao, P., & Zhang, T. (2015). Stochastic optimization with importance sampling for regularized loss minimization. In Bach, F., & Blei, D. (Eds.), *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37 of *Proceedings of Machine Learning Research*, pp. 1–9, Lille, France. PMLR.
- Zhao, S., Song, J., & Ermon, S. (2019). InfoVAE: Balancing learning and inference in variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, p. 5885–5892.
- Zhu, H., & Rohwer, R. (1995). Information geometric measurements of generalisation. Tech. rep., Aston University, Birmingham, UK.
- Zhu, Q., Bi, W., Liu, X., Ma, X., Li, X., & Wu, D. (2020). A batch normalized inference network keeps the KL vanishing away. In Jurafsky, D., Chai, J., Schluter, N., & Tetreault, J. R. (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pp. 2636–2649. Association for Computational Linguistics.