

# An Introduction to Variational Inference

Ankush Ganguly\*

Samuel W. F. Earp

Sertis Vision Lab<sup>†</sup>

## Abstract

Approximating complex probability densities is a core problem in modern statistics. In this paper, we introduce the concept of Variational Inference (VI), a popular method in machine learning that uses optimization techniques to estimate complex probability densities. This property allows VI to converge faster than classical methods, such as, Markov Chain Monte Carlo sampling. Conceptually, VI works by choosing a family of probability density functions and then finding the one closest to the actual probability density—often using the Kullback-Leibler (KL) divergence as the optimization metric. We introduce the Evidence Lower Bound to tractably compute the approximated probability density and we review the ideas behind mean-field variational inference. Finally, we discuss the applications of VI to variational auto-encoders (VAE) and VAE-Generative Adversarial Network (VAE-GAN). With this paper, we aim to explain the concept of VI and assist in future research with this approach.

## 1 Introduction

The core principle of Bayesian statistics is to frame inference about unknown variables as a calculation involving a posterior probability density (Blei et al., 2017). This property of Bayesian statistics makes inference a recurring problem; especially when the posterior density is difficult to compute (Barber, 2012). Algorithms like the elimination algorithm (Gagliardi Cozman, 2000), the message-passing algorithm (belief propagation: Barber, 2012), and the junction tree algorithm (Madsen and Jensen, 1999) have been used to solve exact inference. This method involves analytically computing the conditional probability distribution over the variables of interest. However, the time complexity of exact inference on arbitrary graphical models is NP-hard (Dagum and Luby, 1993). In the case of large data-sets and complicated posterior probability densities, exact inference algorithms favour accuracy at the cost of speed. Approximate inference techniques offer an efficient solution by providing an estimate of the actual posterior probability density.

As a solution to approximate inference, various Markov Chain Monte Carlo (MCMC) methods have been extensively studied since the early 1950s. The most notable among these methods include the Metropolis-Hastings algorithm (Metropolis et al., 1953) and Gibbs sampling (Geman and Geman, 1984). MCMC techniques have since evolved into an indispensable statistical tool for solving

approximate inference. However, these methods are slow to converge and do not scale efficiently.

As an alternative to MCMC sampling, variational methods have been used to tractably approximate complicated probability densities. In recent years, Variational Inference (VI) (introduced, by Jordan et al., 1999) has gained popularity in statistical physics (Regier et al., 2015), data modeling (Tabouy et al., 2020), and neural networks (MacKay, 2015). The problem involves using a metric to select a tractable approximation to the posterior probability density (Blei et al., 2017). This methodology formulates the statistical inference problem as an optimization problem. Thus, we get the speed benefits of maximum a posteriori (MAP) estimation (Murphy, 2013) and can easily scale to large data sets (Blei et al., 2017).

We organize the paper as follows. Section 2 outlines the problem statement and introduces the idea of using KL-divergence, the metric used to measure the information gap between the approximate and the actual posterior probability densities. Section 3 discusses the concept of evidence lower bound and its importance. Section 4 introduces the mean-field variational family. Section 5 applies VI to a toy problem. Section 6 outlines a few practical applications of VI in the field of deep learning and computer vision. Finally, section 7 provides a summary of the paper.

\*Email: agang@sertiscorp.com

<sup>†</sup>597/5 Sukhumvit Road, Watthana, Bangkok, 10110, Thailand

## 2 Problem Statement

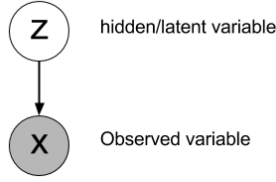


Figure 1: A directed graphical model showing that the observed variable  $X$  is dependent on the latent variable  $Z$ .

Consider the system of random variables illustrated in Figure 1, where  $X$  and  $Z$  represent the observed variable and the hidden (latent) variable, respectively. The arrow drawn from  $Z$  to  $X$  represents the conditional probability density  $p(X|Z)$ , referred to as the *likelihood*. From Bayes' theorem we compute the posterior probability density as,

$$p(Z|X) = \frac{p(X|Z)p(Z)}{p(X)}. \quad (1)$$

The marginal,  $p(X)$ , can be computed as,

$$p(X) = \int_{z \in Z} p(X|z)p(z)dz, \quad (2)$$

where  $z$  is an instance from the sample space of  $Z$ . This marginal probability density of observations is the *evidence* and  $p(Z)$  is referred to as the *prior* because it captures the prior information about  $Z$ . For many models, this evidence integral depends on the selected model and is either unavailable in closed form or requires exponential time to compute (Blei et al., 2017).

The purpose of VI is to provide an analytical approximation of the posterior probability density  $p(Z|X)$  for statistical inference over the latent variables. VI enables efficient computation of a lower bound to the marginal probability density, or the evidence. The idea is that a higher marginal likelihood is indicative of a better fit to the observed data by the chosen statistical model. Additionally, VI addresses the approximation problem by choosing a probability density function  $q$  for the latent variable  $Z$  from a tractable family (Murphy, 2013).

### KL-Divergence

The choice of approximate probability density is done using a metric to measure the difference between it and the actual posterior density (Ranganath et al., 2014). One popular metric used in VI is the Kullback–Leibler (KL) divergence, suggested by Jordan et al. (1999). The KL-divergence is the *relative entropy* between two distributions (Dembo et al., 1991). It is a measure of information that quantifies how similar a probability distribution  $P(X)$  is to a candidate distribution  $Q(X)$  (Shlens, 2014). The *entropy* is a measure of the mean information or uncertainty of a random variable  $X$  (Shannon, 1948), and is defined as

$$\mathbb{H}(P) = - \sum_{x \in X} P(x) \log P(x),$$

where  $X$  is sampled from the distribution  $P$ . Subsequently, the KL-divergence, can be expressed as

$$D_{\text{KL}}(P \parallel Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}, \quad (3)$$

$$D_{\text{KL}}(P \parallel Q) = -\mathbb{H}(P) + \mathbb{H}(P, Q),$$

where  $\mathbb{H}(P, Q)$  is the cross-entropy between the two distributions. In other words, the KL-divergence is the average extra amount of information required to encode the data using the candidate probability distribution instead of the actual distribution (Murphy, 2013). The KL-divergence is non-negative, non-symmetric and is equal to zero or infinite for two perfectly matching and non-matching distributions, respectively (Shlens, 2014).

For a continuous random variable  $X$ , Equation 3 can be extended to the form,

$$D_{\text{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx, \quad (4)$$

where  $P$  and  $Q$  are probability distributions of the continuous random variable  $X$  and  $p$  and  $q$  represent the probability density functions.

Alternatively, we can express the expectation of the logarithmic difference between the probability densities  $p$  and  $q$  as,

$$D_{\text{KL}}(P \parallel Q) = \mathbb{E}_{x \sim P(X)} \left[ \log \frac{p(x)}{q(x)} \right], \quad (5)$$

where the random variable  $x$  is sampled from the probability distribution function  $P$  and  $\mathbb{E}$  is the expectation function.

As established earlier in this Section, the objective of VI is to select an approximate probability density  $q$  from a family of tractable probability densities  $\mathcal{Q}$ . Each  $q(Z) \in \mathcal{Q}$  is a candidate approximation of the actual posterior. The goal is to find the best candidate, i.e. the one with the minimum KL-divergence (Blei et al., 2017). In our formulation, we assume the approximate probability density is not conditioned on the observed variable. Therefore, the inference problem is re-framed as the optimization problem,

$$q^*(Z) = \underset{q(Z) \in \mathcal{Q}}{\operatorname{argmin}} D_{\text{KL}}(P(Z|X) \parallel Q(Z)). \quad (6)$$

We optimize Equation 6 to yield the best approximation  $q^*(\cdot)$  to the actual posterior from the chosen family of densities. The complexity of the optimization depends on our choice for the family of probability densities (Blei et al., 2017; Murphy, 2013) and, therefore, most researchers choose to use the exponential family—motivated by their conjugate nature.

Computing Equation 6 is difficult as taking expectations with respect to  $P$  is assumed to be intractable (Murphy, 2013). Moreover, computing the forward KL-divergence term in Equation 6 would require us to know the posterior. An alternative is to use the reverse KL-divergence where the average cross-entropy between the actual posterior and our approximation is computed by

taking expectations, with respect to the variational distribution. Hence, the optimization problem in Equation 6 can be re-formulated as,

$$q^*(Z) = \arg \min_{q(Z) \in \mathcal{Q}} D_{\text{KL}}(Q(Z) \parallel P(Z|X)). \quad (7)$$

Since  $q(Z)$  is selected from a tractable family of probability densities, computing expectations with respect to  $q$  is also tractable.

### Forward vs. Reverse KL

Let  $P$  and  $Q$  be two distributions with probability density functions  $p$  and  $q$ , where  $q$  is an approximation of  $p$ . As stated earlier, KL-divergence is non-symmetric (Shlens, 2014), i.e.,

$$D_{\text{KL}}(P \parallel Q) \neq D_{\text{KL}}(Q \parallel P),$$

as such, minimizing the forward KL-divergence,  $D_{\text{KL}}(P \parallel Q)$ , yields different results than minimizing the reverse KL-divergence,  $D_{\text{KL}}(Q \parallel P)$ .

The forward KL-divergence is also known as the M-projection or moment projection, (Murphy, 2013) and is defined as,

$$D_{\text{KL}}(P \parallel Q) = \mathbb{E}_{x \sim P(X)} \left[ \log \frac{p(x)}{q(x)} \right].$$

This will be large wherever the approximation fails to *cover up* the actual probability distribution (Murphy, 2013); i.e.

$$\lim_{q(x) \rightarrow 0} \frac{p(x)}{q(x)} \rightarrow \infty, \text{ where } p(x) > 0.$$

So, if  $p(x) > 0$ , we must choose a probability density to ensure that  $q(x) > 0$  (Murphy, 2013). This particular case of optimizing is *zero avoiding* and can intuitively be interpreted as  $q$  over-estimating  $p$ .

The reverse KL-divergence is also known as the I-projection or information projection, (Murphy, 2013) and is defined as,

$$D_{\text{KL}}(Q \parallel P) = \mathbb{E}_{x \sim Q(X)} \left[ \log \frac{q(x)}{p(x)} \right],$$

where,

$$\lim_{p(x) \rightarrow 0} \frac{q(x)}{p(x)} \rightarrow \infty \text{ where } q(x) > 0.$$

The limit indicates the need to force  $q(x) = 0$  wherever  $p(x) = 0$ , otherwise the KL-divergence would be very large. This is *zero forcing* (Murphy, 2013) and can be interpreted as  $q$  under-estimating  $p$ . The difference between the two methods is illustrated in Figure 2; based on Figure 21.1 from Murphy (2013).

### 3 ELBO: Evidence Lower Bound

As mentioned in Section 2, we select a probability density from a tractable family which has the lowest KL-divergence from the actual posterior density. Therefore, inference amounts to solving the optimization problem

defined in Equation 7. However, optimizing Equation 7 is still not tractable because we are required to compute the evidence function. The KL-divergence objective function from Equation 7 can be written as,

$$\mathcal{D} = \mathbb{E}[\log q(z)] - \mathbb{E}[\log p(z|x)], \quad (8)$$

where,

$$\mathcal{D} = D_{\text{KL}}(Q(Z) \parallel P(Z|X)), \quad (9)$$

and all expectations are taken by sampling  $z$  from  $Q(Z)$ . We now expand the conditional probability density  $p(z|x)$  using Equation 1, giving us,

$$\mathcal{D} = \mathbb{E}[\log q(z)] - \mathbb{E}[\log p(z, x)] + \mathbb{E}[\log p(x)]. \quad (10)$$

Since all expectations are under  $Q(Z)$ ,  $\mathbb{E}[\log p(x)]$  is the constant  $\log p(x)$ . Therefore, we can re-write Equation 10 as,

$$\mathcal{D} = \mathbb{E}[\log q(z)] - \mathbb{E}[\log p(z, x)] + \log p(x). \quad (11)$$

The KL-divergence cannot be computed directly as it depends on the evidence. Therefore, we must optimize an alternative objective function that is equivalent to  $D_{\text{KL}}$  up to an added constant,

$$\begin{aligned} -\mathcal{D} + \log p(x) &= \mathbb{E}[\log p(z, x)] - \mathbb{E}[\log q(z)], \\ \text{ELBO}(Q) &= \mathbb{E}[\log p(z, x)] - \mathbb{E}[\log q(z)], \end{aligned} \quad (12)$$

where the term ELBO is an abbreviation for evidence lower bound. The ELBO the sum of the negative KL-divergence and the constant term  $\log p(x)$ . Maximizing the ELBO is equivalent to minimizing the KL-divergence (Blei et al., 2017).

An intuitive explanation of the ELBO can be derived by re-arranging the terms of Equation 12, as

$$\begin{aligned} \text{ELBO}(Q) &= \mathbb{E}[\log p(z, x)] - \mathbb{E}[\log q(z)], \\ &= \mathbb{E}[\log p(x|z)] + \mathbb{E}[\log p(z) - \log q(z)], \\ &= \mathbb{E}[\log p(x|z)] - D_{\text{KL}}(Q(Z) \parallel P(Z)). \end{aligned} \quad (13)$$

Thus, the ELBO is the sum of the expected log likelihood of the data and the KL-divergence between the prior and approximated posterior probability density. The expected log likelihood describes how well the chosen statistical model fits the data. The KL-divergence encourages the variational probability density to be close to the actual prior. Thus, the ELBO can be seen as a regularised fit to the data.

The ELBO lower-bounds the (log) evidence,  $\log p(x)$ . This property was explored by Jordan et al. (1999), where the authors used Jensen's inequality (Klarićić Bakula et al., 2008) to derive the relationship between the ELBO and the evidence function. The derivation is as follows:

$$\begin{aligned} \log p(x) &= \log \int_{z \in Z} p(x, z) dz, \\ &= \log \int_{z \in Z} p(x, z) \frac{q(z)}{q(z)} dz, \\ &= \log \mathbb{E}_{z \sim Q(z)} \left[ \frac{p(x, z)}{q(z)} \right], \\ &\geq \mathbb{E}_{z \sim Q(z)} [\log p(x, z)] - \mathbb{E}_{z \sim Q(z)} [\log q(z)], \\ &\geq \text{ELBO}(Q). \end{aligned} \quad (14)$$

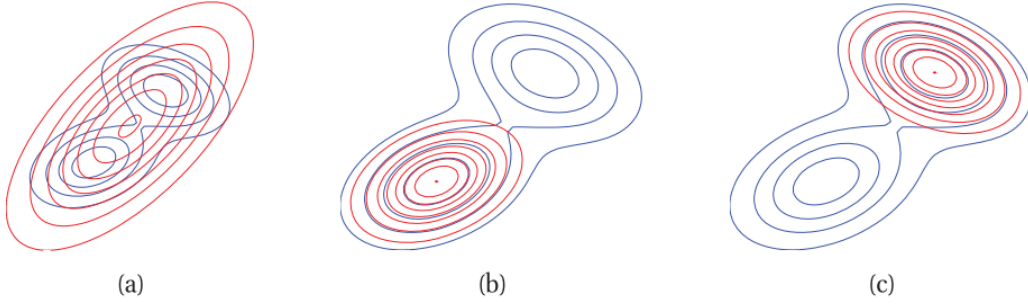


Figure 2: Figure illustrating forward vs reverse KL-divergence on a bimodal distribution. The blue and the red contours represent the actual probability density,  $p$ , and the unimodal approximation,  $q$ , respectively. The left panel shows the forward KL-divergence minimization where  $q$  tends to *cover*  $p$ . The centre and the right panels show the reverse KL-divergence minimization where  $q$  locks on to one of the two modes.

This relationship between the ELBO and  $\log p(x)$  has motivated researchers to use the variational lower bound as the criterion for model selection. This bound serves as a good approximation to the marginal likelihood; providing a basis for model selection. Applications of VI for model selection have been explored in a wide variety of tasks such as for mixture models (McGrory and Titterton, 2007), cross-validation mode selection (Nott et al., 2012) and in a more general setting by Bernardo et al. (2003).

#### 4 Mean field variational family

We briefly introduce the mean field variational family for VI, where the latent variables are assumed to be mutually independent—each governed by a distinct factor in the variational probability density (see Bishop, 2006, for a more detailed explanation). This assumption greatly simplifies the complexity of the optimization process. A generic member of the mean field variational family is

$$q(Z|X) = \prod_{j=1}^m q_j(Z_j), \quad (15)$$

where  $m$  is the number of latent variables. The observed data  $X$  does not appear in Equation 15, therefore any probability density from this variational family is not a model of the data. Instead, it is the ELBO, and the corresponding KL-divergence minimization problem, which connects the fitted variational probability density to the data and model (Blei et al., 2017; Murphy, 2013).

#### 5 A toy problem

In this section, we explore, in detail, how VI can be used to approximate a mixture of Gaussians. Consider a distribution of  $N$  real-valued data-points  $x = x_1, x_2, \dots, x_N$  sampled from a mixture of  $K$  univariate Gaussians with means  $\mu_j = \mu_1, \mu_2, \dots, \mu_K$ . We assume that the variance of the mean’s prior is a fixed hyperparameter  $\sigma^2$  while the observation variance is one. For this problem, we define a single data-point as,

$$x_i \sim \mathcal{N}(c_i^T \mu, 1) \text{ for } i = 1, 2, \dots, N, \quad (16)$$

which is drawn from a distributions with mean,

$$\mu_j \sim \mathcal{N}(0, \sigma^2) \text{ for } j = 1, 2, \dots, K, \quad (17)$$

and is assigned to a cluster using,

$$c_i \sim \mathcal{U}(K) \text{ for } i = 1, 2, \dots, N, \quad (18)$$

where  $c_i$  is a one-hot vector of  $K$ -dimensions, and with latent variables  $\mu$  and  $c$ . In our case, the one-hot vector is a  $K$ -dimensional binary vector where each dimension represents a cluster. A data-point belonging to the  $l$ -th cluster will be represented by a value of one in the  $l$ -th dimension of the one-hot vector, while the  $(K - 1)$  remaining dimensions will have a value of zero.

We assume the approximate posterior probability density to be from the mean-field variational family (Section 4). Thus, the variational parameterization is given by,

$$q(\mu, c) = \prod_{j=1}^K q(\mu_j; m_j, s_j^2) \prod_{i=1}^N q(c_i; \phi_i). \quad (19)$$

Each latent variable is governed by it’s own variational factor (Blei et al., 2017). Here, the mixture components are Gaussian with variational parameters (mean  $m_k$  and variance  $s_k^2$ ) specific to the  $k$ -th cluster. The cluster assignments are categorical with variational parameters ( $K$ -dimensional cluster probabilities  $\phi_i$  vector) specific to the  $i$ -th data point.

The definition of ELBO from Equation 12 applied to this specific case is,

$$\text{ELBO}_{m, s^2, \phi} = \mathbb{E}[\log p(x, \mu, c)] - \mathbb{E}[\log q(\mu, c)], \quad (20)$$

$$\text{ELBO}_{m, s^2, \phi} = \text{ELBO}(m, s^2, \phi).$$

We maximize the ELBO to derive the optimal values of the variational parameters (see Appendix A). The optimal



values for  $m$ ,  $s$  and  $\phi$  are given by,

$$m_j^* = \frac{\sum_i \phi_{ij} x_i}{\frac{1}{\sigma^2} + \sum_i \phi_{ij}}, \quad (21)$$

$$(s_j^2)^* = \frac{1}{\frac{1}{\sigma^2} + \sum_i \phi_{ij}}, \quad (22)$$

$$\phi_{ij}^* \propto e^{-\frac{1}{2}(m_j^2 + s_j^2) + x_i m_j}. \quad (23)$$

We employ the Coordinate Ascent VI (CAVI) algorithm to optimize the ELBO. Algorithm 1 (Blei et al., 2017) describes the steps to optimize the ELBO using CAVI.

For our experimental setup, we select  $K = 3$ , i.e., a mixture of three univariate Gaussians. We generate the data by randomly sampling 1000 data-points for each of the three Gaussians. The left panel in Figure 3 illustrates how the algorithm approximates the parameters of each individual Gaussian by maximizing the ELBO. We select a maximum iteration of 1000 steps, however, the ELBO converges by iteration 60 as illustrated in the right panel of Figure 3.

---

**Algorithm 1:** CAVI for a Gaussian mixture model

---

**Data:** Data  $x_{1:n}$ ,  $K$  mixture components and prior variance of component means  $\sigma^2$   
**Result:** Variational densities  $q(\mu_j; m_j, s_j^2)$  and  $q(c_i; \phi_i)$   
 $m = m_{1:K}, s^2 = s_{1:K}^2, \phi = \phi_{1:N} \leftarrow$  initialize variational parameters  
**while** the ELBO has not converged **do**  
  **for**  $i \in 1, \dots, N$  **do**  
    Set  $\phi_{ij} \propto \exp[-\frac{1}{2}(m_j^2 + s_j^2) + x_i m_j]$   
  **end**  
  **for**  $k \in 1, \dots, K$  **do**  
    Set  $m_j \leftarrow \frac{\sum_i \phi_{ij} x_i}{\frac{1}{\sigma^2} + \sum_i \phi_{ij}}$   
    Set  $s_j^2 \leftarrow \frac{1}{\frac{1}{\sigma^2} + \sum_i \phi_{ij}}$   
  **end**  
  Compute  $\text{ELBO}(m, s^2, \phi)$   
**end**  
**return**  $q(m, s^2, \phi)$

---

## 6 Applications

We have established the optimization process for VI, next we look at some applications in generative modelling.

### 6.1 VAE: Variational Auto-Encoder

An auto-encoder is a neural network that aims to learn or encode a low-dimensional representation for high-dimensional data, e.g. images. Different variants of auto-encoders exist that aim to learn meaningful representations of high-dimensional data. One such variant is the Variational Auto-Encoder (VAE). Introduced by Kingma and Welling (2013), the VAE is a statistical model which is essentially a stochastic variational inference algorithm. The VAE uses the concept of variational inference to compress the high-dimensional data into a latent vector while assuming a multi-variate distribution as a prior for the same latent

vector. The statistical model uses gradient backpropagation to approximate the posterior distribution for the latent vector. For large data sets we update the VAE’s parameters using small mini-batches or even single data points.

Since their inception, VAEs have been widely used for generative modelling. They are easy to implement, converge faster than MCMC methods, and scale efficiently to large data sets; this makes them ideal for generative modelling of image data. However, the images they generate tend to have reduced quality compared to the input images. This is the effect of minimizing the reverse KL-divergence, which results in the approximate distribution being locked to one of the modes—as explained in Section 3.

Kingma and Welling (2013) introduce the stochastic variational inference algorithm (VAE) that reparameterizes the variational lower bound, yielding a lower bound estimator that can be optimized using standard stochastic gradient methods (Kingma and Welling, 2013).

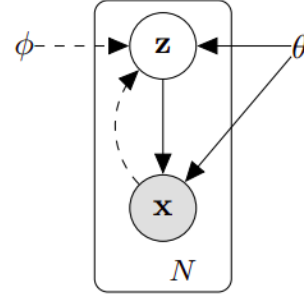


Figure 4: Illustration of the type of directed graphical model under consideration with  $N$  observed data-points. Solid lines denote the generative model, dashed lines denote the variational approximation to the intractable posterior density. The variational parameters  $\phi$  are learned jointly with the generative model parameters  $\theta$  (Kingma and Welling, 2013).

The recognition model,  $q_\phi(Z|X)$ , can be interpreted as a probabilistic encoder, since given a data point  $x$  the encoder produces a latent vector  $z$ . This latent vector is then used to generate a sample from the likelihood density,  $p_\theta(X|Z)$ , and, therefore, the generative model can be interpreted as a probabilistic decoder (Kingma and Welling, 2013). The probability density  $q_\phi(Z|X)$  serves as an approximation of the actual posterior probability density  $p_\theta(Z|X)$ . Using a mini-batch of data points sampled from  $X$ , the encoder transforms these data points into the latent space,  $Z$ , which the decoder uses to generate the samples in  $X$ .

As established in Equation 13, maximizing the ELBO is equivalent to minimizing the KL-divergence (Blei et al., 2017). In order to jointly optimize the recognition model and the generative model on mini-batches of data, we differentiate and optimize the lower bound with respect to both the variational and the generative parameters,  $\phi$  and

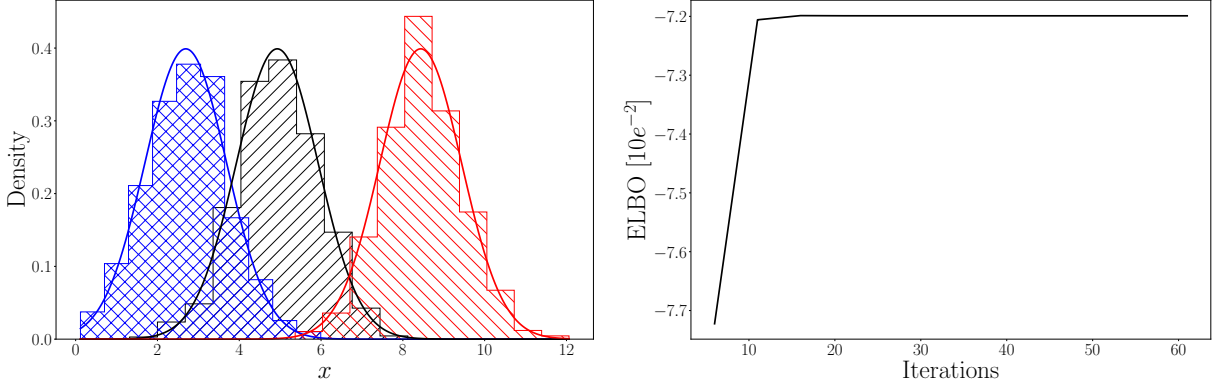


Figure 3: Left: a histogram showing the distribution of data sampled from the three univariate Gaussians. The curved lines indicate the fit by maximising the ELBO. Right: an illustration of the convergence of ELBO using CAVI.

$\theta$ . In this case Equation 15 can be rewritten as,

$$\begin{aligned} \mathcal{D}_{Z,X,\phi,\theta} &= D_{\text{KL}}(Q_\phi(Z|X) \parallel P_\theta(Z)), \\ \mathcal{L}(\phi, \theta; x) &= -\mathcal{D}_{Z,X,\phi,\theta} + \mathbb{E}_{z \sim Q_\phi(Z|X)} [\log p_\theta(x|z)]. \end{aligned} \quad (24)$$

If we look closely at the terms on the right-hand-side of Equation 24, we can see the connection to auto-encoders, where the second term is the expected negative reconstruction error and the KL-divergence term can be interpreted as a regularization term. In order to optimize Equation 24 using standard gradient-based techniques, [Kingma and Welling \(2013\)](#) introduce the Auto-Encoding Variational Bayes (AEVB) algorithm to efficiently compute the gradient of the ELBO from Equation 24. For a chosen approximate posterior  $q_\phi(z|x)$ , we re-parameterize the random variable  $z = q_\phi(z|x)$  with a differentiable transformation  $g_\phi(\epsilon, x)$  of a noise variable  $\epsilon$ , such that,

$$\begin{aligned} z &= g_\phi(\epsilon, x), \\ \epsilon &\sim P(\epsilon). \end{aligned}$$

This re-parameterization allows us to form Monte Carlo estimates of expectations of the function  $\log p_\theta(x|z)$ . Forming Monte Carlo estimates enables us to numerically evaluate the expectation,

$$\mathbb{E}_{z \sim Q_\phi(Z|X)} [\log p_\theta(x|z)].$$

We draw independent samples,  $z^i$ , from the variational distribution,  $Q_\phi(Z|X)$ , and then compute the average of the function evaluated at these samples ([Mohamed et al., 2020](#)). Therefore, the Monte Carlo estimates of the expectation of the function  $\log p_\theta(x^i|z)$  when  $z \sim q_\phi(z|x^{(i)})$  is as follows,

$$\mathbb{E}_{z \sim Q_\phi(Z|x^{(i)})} \simeq \frac{1}{L} \sum_{l=1}^L \log p_\theta(x^{(i)}|z^{(i,l)}),$$

where,  $\epsilon^{(l)} \sim P(\epsilon)$  and  $L$  is the number of samples per data point.

Applying the above re-parameterization to the variational lower-bound of Equation 24 we can re-formulate the

ELBO to a Stochastic Gradient Variational Bayes (SGVB) estimator  $\mathcal{L}(\phi, \theta; x^{(i)})$ , as

$$\mathcal{L}_{\phi,\theta,x^{(i)}} \simeq -\mathcal{D}_{Z,X,\phi,\theta} + \frac{1}{L} \sum_{l=1}^L \log p_\theta(x^{(i)}|z^{(i,l)}), \quad (25)$$

$$\mathcal{L}_{\phi,\theta,x^{(i)}} = \mathcal{L}(\phi, \theta; x^{(i)}).$$

We re-parameterize the variational lower bound in terms of a deterministic random variable which enables us to use gradient based optimizers on mini-batches of data. This further enables the optimization of the parameters of the distribution while still maintaining the ability to randomly sample from that distribution ([Doersch, 2016](#)).

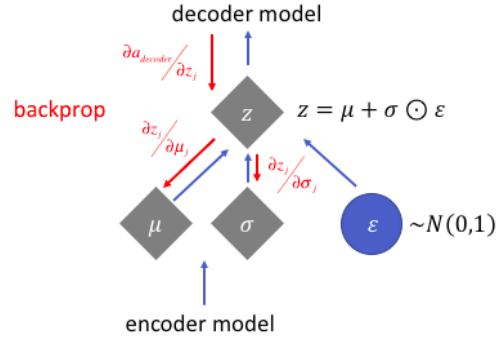


Figure 5: The learning process in a typical VAE using gradient back-propagation.

In order to simplify the calculations, we assume the variational approximate posterior to be a multi-variate Gaussian with diagonal co-variance structure ([Kingma and Welling, 2013](#)). As for the prior we assume a multivariate Gaussian  $\mathcal{N}(z; 0, \mathbf{I})$  where,

$$\begin{aligned} \log q_\phi(z|x^{(i)}) &= \log \mathcal{N}(z; \mu^{(i)}, (\sigma^{(i)})^2 \mathbf{I}), \\ p_\theta(z) &= \mathcal{N}(z; 0, \mathbf{I}), \\ z^{(i)} &= \mu^{(i)} + \sigma^{(i)} \odot \epsilon, \\ \epsilon &\sim \mathcal{N}(0, \mathbf{I}). \end{aligned}$$

Using the above parameterization, the KL-divergence term in Equation 25 can be derived as Equation 27 (as shown in

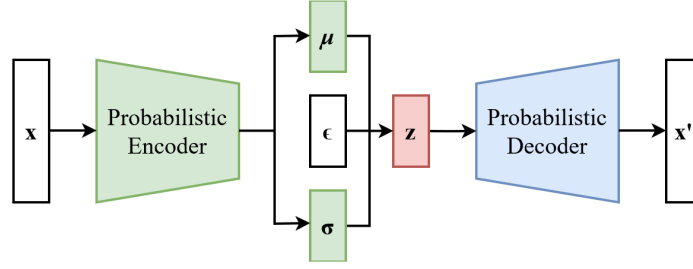


Figure 6: A diagram showing a variational autoencoder model.

Box 1). Subsequently, Equation 25 can be used to define the loss function for the VAE framework at  $x^{(i)}$ , as

$$\begin{aligned} \mathcal{L}_{\phi, \theta; x^{(i)}} \simeq & \frac{1}{2} \sum_{j=1}^J \left[ 1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2 \right] \\ & + \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x^{(i)} | z^{(i, l)}), \end{aligned} \quad (26)$$

where  $J$  is the dimensionality of  $z$ .

A variant of the VAE framework,  $\beta$ -VAE, adds an extra hyperparameter to the VAE objective which constricts the effective encoding capacity of the latent space. The  $\beta$ -VAE training objective is,

$$\begin{aligned} \mathcal{L}(\phi, \theta; x) = & -\beta \mathcal{D}_{Z, X, \phi, \theta} \\ & + \mathbb{E}_{z \sim Q_{\phi}(Z|X)} [\log p_{\theta}(x|z)], \end{aligned}$$

where  $\beta = 1$  corresponds to the original VAE formulation of [Kingma and Welling \(2013\)](#). This constriction

encourages the latent representation to be more factorised. However, this can lead to even worse reconstruction quality as compared to the standard VAE framework. This is caused by a trade-off introduced by the modified training objective that punishes reconstruction quality in order to encourage disentanglement between the latent representations ([Burgess et al., 2018](#)). Varying  $\beta$  during training encourages the model to learn different latent representations of the data. A high value of  $\beta$  encourages disentanglement in the latent space but at the cost of reduced reconstruction quality. To mitigate this reconstruction problem, the authors introduce a capacity control parameter  $\mathcal{C}$ . Increasing  $\mathcal{C}$  from zero to a value large enough produces good quality reconstructions during training ([Burgess et al., 2018](#)). The modified  $\beta$ -VAE training objective is given by,

$$\begin{aligned} \mathcal{L}(\phi, \theta; x) = & -\beta [\mathcal{D}_{Z, X, \phi, \theta} - \mathcal{C}] \\ & + \mathbb{E}_{z \sim Q_{\phi}(Z|X)} [\log p_{\theta}(x|z)]. \end{aligned}$$

Box 1: Derivation of the KL-divergence.

$$\begin{aligned} -\mathcal{D}_{Z, x^{(i)}, \phi, \theta} &= D_{\text{KL}}(Q_{\phi}(Z|x^{(i)}) \parallel P_{\theta}(Z)) \\ &= -\mathbb{E}_{z \sim Q_{\phi}(Z|X)} \left[ \log \frac{q_{\phi}(z|x^{(i)})}{p_{\theta}(z)} \right] \\ &= -\int q_{\phi}(z|x^{(i)}) \left[ \log \frac{q_{\phi}(z|x^{(i)})}{p_{\theta}(z)} \right] dz \\ &= \int q_{\phi}(z|x^{(i)}) \log p_{\theta}(z) dz - \int q_{\phi}(z|x^{(i)}) \log q_{\phi}(z|x^{(i)}) dz \\ &= \int \mathcal{N}[z; \mu^{(i)}, (\sigma^{(i)})^2 \mathbf{I}] \log \mathcal{N}[z; 0, \mathbf{I}] dz - \int \mathcal{N}[z; \mu^{(i)}, (\sigma^{(i)})^2 \mathbf{I}] \log \mathcal{N}[z; \mu^{(i)}, (\sigma^{(i)})^2 \mathbf{I}] dz \\ &= \frac{1}{2} \left[ J \log(2\pi) + \sum_{j=1}^J (1 + \log(\sigma_j^{(i)})^2) - J \log(2\pi) - \sum_{j=1}^J ((\mu_j^{(i)})^2 + (\sigma_j^{(i)})^2) \right] \\ &= \frac{1}{2} \sum_{j=1}^J \left[ 1 + \log(\sigma_j^{(i)})^2 - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2 \right] \end{aligned} \quad (27)$$

## 6.2 GAN: Generative Adversarial Network

Generative adversarial networks (GANs), introduced by [Goodfellow et al. \(2014\)](#), are deep-learning based generative models that are extensively used to create realistic

data samples across a range of problems—most notably in computer vision.

A typical GAN framework involves simultaneously training two models; a generator and a discriminator. The

generator tries to capture the data distribution by mapping a latent vector to a data-point, thereby generating new samples with similar statistical properties as the training data. The discriminator aims to estimate the likelihood of a sample being drawn from the training data or created by the generator (Goodfellow et al., 2014). We can train a GAN by minimizing the objective function,

$$\min_{\text{Gen}} \max_{\text{Dis}} \mathcal{L}_{\text{GAN}} = \mathbb{E}_{x \sim P_{\text{data}}(X)} [\log(\text{Dis}(x))] + \mathbb{E}_{z \sim P_z(Z)} [\log(1 - \text{Dis}(\text{Gen}(z)))], \quad (28)$$

where the generator,  $\text{Gen}(z)$ , takes a sample from the latent distribution,  $P_z(Z)$ , and creates a new data-point while the discriminator,  $\text{Dis}(x)$ , takes a data-point from both the real distribution,  $P_{\text{data}}(X)$ , and the new data-point  $\text{Gen}(z)$  and assigns probabilities to both. To minimize the objective, the discriminator will try to assign probabilities close to zero and one for data sampled from the real distribution and anything created by the generator, respectively. The GAN's objective is to train the discriminator to efficiently discriminate between real and generated data while encouraging the generator to reproduce the true data distribution (Goodfellow et al., 2014; Larsen et al., 2016). There is a unique solution where the generator successfully recovers the training data distribution. At the same time, the discriminator ends up assigning equal probability to samples from the training data and the generator.

### 6.3 VAE-GAN

Different variants of the original GAN framework have evolved since its inception, such as the VAE-GAN introduced by Larsen et al. (2016). This approach uses the learned feature representations in the GAN discriminator as a basis for the VAE reconstruction objective.

Larsen et al. (2016) show that unsupervised training, like that of a GAN, can result in the latent image representation with disentangled factors of variation (Bengio et al., 2013). This means that the model learns an embedding space with, abstract, high-level visual features which can be modified using simple arithmetic (Larsen et al., 2016).

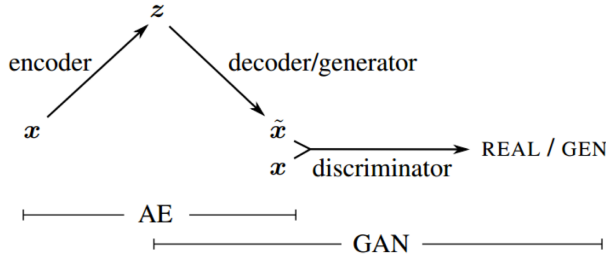


Figure 7: Diagram of the VAE-GAN framework Larsen et al. (2016).

As we see in Section 6.1, the VAE consists of an encoder and a decoder given by,

$$z = \text{Enc}(x) = q(z|x), \\ \tilde{x} = \text{Dec}(z) = p(x|z),$$

where  $z$  is the latent representation of a data sample  $x$  taken from marginal likelihood distribution. The encoder,  $\text{Enc}(x)$ , takes a data-sample,  $x$ , and approximates the posterior density  $q(z|x)$ . Whereas the decoder,  $\text{Dec}(z)$ , takes a sample from the latent space,  $z$ , and generates a sample from the likelihood density  $p(x|z)$ . Larsen et al. (2016) define  $\mathcal{L}_{\text{VAE}}$  as the negative of training objective of a vanilla VAE (given by Equation 24) and provide the objective function,

$$\begin{aligned} \mathcal{L}_{\text{VAE}} &= -\mathcal{L}(\phi, \theta; x), \\ &= D_{\text{KL}}(Q(Z|X) \parallel P(Z)) \\ &\quad - \mathbb{E}_{z \sim Q(Z|X)} [\log p(x|z)]. \end{aligned} \quad (29)$$

The authors go on to describe the terms of Equation 29 as,

$$\begin{aligned} \mathcal{L}_{\text{llike}}^{\text{pixel}} &= -\mathbb{E}_{z \sim Q(Z|X)} [\log p(x|z)], \\ \mathcal{L}_{\text{prior}} &= D_{\text{KL}}(Q(Z|X) \parallel P(Z)), \end{aligned}$$

where  $\mathcal{L}_{\text{llike}}^{\text{pixel}}$  is the negative expected log likelihood and  $\mathcal{L}_{\text{prior}}$  is the KL-divergence between the approximated posterior density and the prior on the latent variable. The KL-divergence term in Equation 29 can also be interpreted as a regularization term. Therefore, the VAE loss is the sum of the negative expected log likelihood (the reconstruction error) and the regularization term (Larsen et al., 2016).

The authors, further, propose a technique to exploit the capacity of the discriminator to differentiate between real and generated images. The capacity of a neural network is defined as an upper bound on the number of bits that can be extracted from the training data and stored in the architecture during learning (Baldi and Vershynin, 2019). Larsen et al. (2016) replace the VAE reconstruction error term in Equation 29, for better quality images, with a reconstruction error expressed in the GAN discriminator. For this, they introduce a Gaussian observation model for the hidden representation of the  $l$ -th layer of the discriminator  $\text{Dis}_l(x)$  with mean  $\text{Dis}_l(\tilde{x})$  and identity covariance,

$$p(\text{Dis}_l(x)|z) = \mathcal{N}(\text{Dis}_l(x) | \text{Dis}_l(\tilde{x}), \mathbf{I}),$$

where  $\tilde{x} = \text{Dec}(z)$  is the output from the decoder for the data point  $x$ . Subsequently, the VAE reconstruction error in Equation 29 is replaced with the following,

$$\mathcal{L}_{\text{llike}}^{\text{Dis}_l} = -\mathbb{E}_{z \sim Q(Z|X)} [\log p(\text{Dis}_l(x)|z)].$$

Thus, the combined training objective for the VAE-GAN is as follows,

$$\mathcal{L} = \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{llike}}^{\text{Dis}_l} + \mathcal{L}_{\text{GAN}}. \quad (30)$$

The parameters for the decoder model  $\theta_{\text{Dec}}$  are updated weighing the decoder's reconstruction ability against the discriminator's discernment. The authors use a parameter,  $\gamma$ , to weigh the VAE's ability to reconstruct against the discriminator. Therefore, the reconstruction error is,

$$\theta_{\text{Dec}} \leftarrow \theta_{\text{Dec}}^+ - \nabla_{\theta_{\text{Dec}}} \left( \gamma \mathcal{L}_{\text{llike}}^{\text{Dis}_l} - \mathcal{L}_{\text{GAN}} \right).$$

The training procedure for VAE-GAN is illustrated in Algorithm 2 and Figure 8.



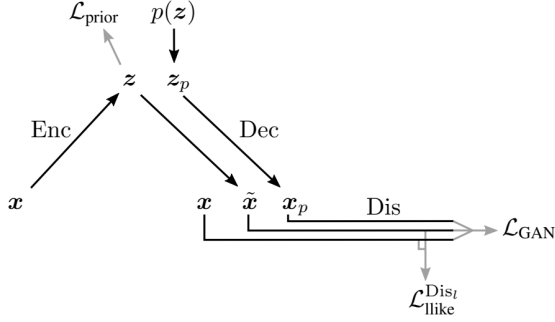


Figure 8: Illustration of the flow through the combined VAE-GAN framework. [Larsen et al. \(2016\)](#) combine a VAE with a GAN by collapsing the decoder and the generator into one. The gray lines represent the training objective.

---

**Algorithm 2:** Training algorithm for VAE-GAN taken from [Larsen et al. \(2016\)](#)

---

$\theta_{\text{Enc}}, \theta_{\text{Dec}}, \theta_{\text{Dis}} \leftarrow$  initialize network parameters  
**repeat**  
 $\mathbf{X} \leftarrow$  random mini-batch from data set  
 $\mathbf{Z} \leftarrow \text{Enc}(\mathbf{X})$   
 $\mathcal{L}_{\text{prior}} \leftarrow D_{\text{KL}}(Q(\mathbf{Z}|\mathbf{X}) \parallel P(\mathbf{Z}))$   
 $\tilde{\mathbf{X}} \leftarrow \text{Dec}(\mathbf{X})$   
 $\mathcal{L}_{\text{llike}}^{\text{Dis}_l} \leftarrow -\mathbb{E}_{q(\mathbf{Z}|\mathbf{X})} \left[ p(\text{Dis}_l(\mathbf{X})|\mathbf{Z}) \right]$   
 $\mathbf{Z}_p \leftarrow$  samples from prior  $\mathcal{N}(0, \mathbf{I})$   
 $\mathbf{X}_p \leftarrow \text{Dec}(\mathbf{Z}_p)$   
 $\mathcal{L}_{\text{GAN}} \leftarrow \log(\text{Dis}(\mathbf{X}))$   
 $+ \log(1 - \text{Dis}(\tilde{\mathbf{X}})) + \log(1 - \text{Dis}(\mathbf{X}_p))$   
  
 UPDATE PARAMETERS ACCORDING TO GRADIENTS  
 $\theta_{\text{Enc}} \leftarrow \theta_{\text{Enc}}^+ - \nabla_{\theta_{\text{Enc}}} (\mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{llike}}^{\text{Dis}_l})$   
 $\theta_{\text{Dec}} \leftarrow \theta_{\text{Dec}}^+ - \nabla_{\theta_{\text{Dec}}} (\gamma \mathcal{L}_{\text{llike}}^{\text{Dis}_l} - \mathcal{L}_{\text{GAN}})$   
 $\theta_{\text{Dis}} \leftarrow \theta_{\text{Dis}}^+ - \nabla_{\theta_{\text{Dis}}} (\mathcal{L}_{\text{GAN}})$   
**until convergence**

---

In recent years, the use of deep convolutional neural networks (CNNs), has resulted in state-of-the-art performance for generative modeling tasks—especially in the field of computer vision (e.g. [Chen et al., 2016](#); [Karras et al., 2019](#); [Pidhorskyi et al., 2020](#); [Radford et al., 2015](#)). Such networks are computationally efficient using convolution operations to extract information from high-dimensional data without human supervision ([Simonyan et al., 2014](#)). Motivated this success, [Larsen et al. \(2016\)](#) train CNNs along with batch-normalisation ([Ioffe and Szegedy, 2015](#)), ReLU activations ([Krizhevsky et al., 2012](#)), consecutive down- and up-sampling layers in both the encoder and discriminator. [Liu et al. \(2015\)](#) train:

- a traditional VAE,

- a VAE with learned distance,  $\text{VAE}_{\text{Dis}_l}$ , where the authors first train a GAN and use the  $l$ -th layer of the discriminator network as a learned similarity measure,
- the proposed VAE-GAN framework,

on the CelebA data set. As shown in Figure 9, the visual realism of the VAE-GAN is superior to the traditional VAE. Additionally, the learned latent space can be used to modify high-level facial features, such as, skin tone and hair colour.



Figure 9: Reconstructions from different auto-encoders [Larsen et al. \(2016\)](#).

The application of VI in machine learning is not limited to these frameworks. Different variants of both VAE and VAE-GAN have been implemented and have continued to produce state of the art research in generative modelling tasks (e.g. [Hou et al., 2017](#); [Li et al., 2020](#); [Purkait et al., 2020](#); [Shu et al., 2018](#); [Simonovsky and Komodakis, 2018](#); [Vahdat and Kautz, 2020](#); [Zhang et al., 2018](#); [Zhao et al., 2019](#)). In addition to image generation tasks, machine learning problems like anomaly detection, time series estimation, language modelling, dimensionality reduction and unsupervised representation learning have all used VI in one form or the other (e.g. [Graving and Couzin, 2020](#); [Li et al., 2021](#); [Pol et al., 2019](#); [Yang et al., 2017](#)).

## 7 Discussion

We have introduced the concept of VI, a tool to perform approximate statistical inference. VI re-structures the statistical problem of estimating the posterior probability density over the latent variable, given an observed variable, into an optimization problem. The key idea is to select a probability density from a family of tractable densities that is closest to the actual posterior probability density. We have demonstrated how:

- the KL-divergence can be used as a metric to measure the closeness between densities,
- the ELBO can be used as a criterion for model selection to better fit the observed data,
- VI can be used to fit a mixture of Gaussians.

Moreover, we briefly presented the scenarios where VI has been applied to modern machine learning tasks, specifically in computer vision and generative modeling, and investigated how combining deep learning and VI enable us to perform inference on extremely complex posterior distributions.

VI is a powerful tool that allows us to approximate the actual probability density of the latent representation. However, there are still many open avenues for statistical research. One such avenue is to develop better approximations (achieving lower KL-divergence) to the posterior density, while maintaining efficient optimization. For example, the mean-field family makes strong independence assumptions which aid in scalable optimization. However, these assumptions may lead the variance of the approximated density to under-represent that of the target density (Blei et al., 2017). As an alternative to the mean-field method, Minka (2005) use a fully-factorized approximation with no explicit exponential family constraint along with loopy belief propagation to achieve a lower KL-divergence. Another possible area of research is to use  $\alpha$ -divergance measures (Zhang et al., 2018) to get a tighter fit to the ELBO. Although research in the field of VI algorithm has grown in recent years, efforts to make VI more efficient, accurate, scalable and easier are still ongoing.

## 8 Acknowledgements

We would like to thank our colleagues Sanjana Jain, Dhruva Pujary and, Ukrit Watchareeruetai for their constructive input and feedback during the writing of this paper.

## References

- P. Baldi and R. Vershynin. The capacity of feedforward neural networks. *Neural networks*, 116:288–311, 2019.
- D. Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.
- Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, M. West, et al. The variational bayesian algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian statistics*, 7(453-464):210, 2003.
- C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.
- D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.
- C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in  $\beta$ -vae. *arXiv preprint arXiv:1804.03599*, 2018.
- X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2180–2188, 2016.
- P. Dagum and M. Luby. Approximating probabilistic inference in bayesian belief networks is np-hard. *Artificial intelligence*, 60(1):141–153, 1993.
- A. Dembo, T. M. Cover, and J. A. Thomas. Information theoretic inequalities. *IEEE Transactions on Information theory*, 37(6):1501–1518, 1991.
- C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.
- F. Gagliardi Cozman. Generalizing Variable Elimination in Bayesian Networks. *Probabilistic Reasoning in Artificial Intelligence*, pages 1–11, 2000.
- S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.
- I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.
- J. Graving and I. Couzin. VAE-SNE: a deep generative model for simultaneous dimensionality reduction and clustering. 2020. doi: 10.1101/2020.07.17.207993.
- X. Hou, L. Shen, K. Sun, and G. Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1133–1141. IEEE, 2017.
- S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.
- M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.
- T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.
- D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- M. Klarićić Bakula, M. Matić, and J. Pečarić. On some general inequalities related to jensen’s inequality. *Inequalities and Applications*, pages 233–243, 2008.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105, 2012.

- A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016.
- W. Li, W. Hu, N. Chen, and C. Feng. Stacking vae with graph neural networks for effective and interpretable time series anomaly detection. *arXiv preprint arXiv:2105.08397*, 2021.
- Z. Li, R. Togo, T. Ogawa, and M. Haseyama. Variational autoencoder based unsupervised domain adaptation for semantic segmentation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2426–2430. IEEE, 2020.
- Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. *Proceedings of the IEEE International Conference on Computer Vision, 2015 International Conference on Computer Vision, ICCV 2015:3730–3738*, 2015. ISSN 15505499. doi: 10.1109/ICCV.2015.425.
- D. J. MacKay. *Information Theory, Inference, and Learning Algorithms* David, volume 44. 2015. ISBN 9780521642989. doi: 10.1198/jasa.2005.s54.
- A. L. Madsen and F. V. Jensen. Lazy propagation: a junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, 113(1-2):203–245, 1999.
- C. A. McGrory and D. Titterton. Variational approximations in bayesian model selection for finite mixture distributions. *Computational Statistics & Data Analysis*, 51(11):5352–5367, 2007.
- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.
- T. P. Minka. Divergence measures and message passing. *Microsoft Research Technical Report*, (MSR-TR-2005-173):17, 2005. ISSN 0735-0015.
- S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih. Monte carlo gradient estimation in machine learning. *J. Mach. Learn. Res.*, 21(132):1–62, 2020.
- K. P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013. ISBN 9780262018029 0262018020.
- D. J. Nott, S. L. Tan, M. Villani, and R. Kohn. Regression density estimation with variational methods and stochastic approximation. *Journal of Computational and Graphical Statistics*, 21(3):797–820, 2012.
- S. Pidhorskyi, D. A. Adjeroh, and G. Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14104–14113, 2020.
- A. A. Pol, V. Berger, C. Germain, G. Cerminara, and M. Pierini. Anomaly detection with conditional variational autoencoders. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, pages 1651–1657. IEEE, 2019.
- P. Purkait, C. Zach, and I. Reid. Sg-vae: Scene grammar variational autoencoder to generate new indoor scenes. In *European Conference on Computer Vision*, pages 155–171. Springer, 2020.
- A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.
- J. Regier, A. Miller, J. McAuliffe, R. Adams, M. Hoffman, D. Lang, D. Schlegel, and M. Prabhat. Celeste: Variational inference for a generative model of astronomical images. In *International Conference on Machine Learning*, pages 2095–2103. PMLR, 2015.
- C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.
- J. Shlens. Notes on kullback-leibler divergence and likelihood. *arXiv preprint arXiv:1404.2000*, 2014.
- R. Shu, H. H. Bui, S. Zhao, M. J. Kochenderfer, and S. Ermon. Amortized inference regularization. *arXiv preprint arXiv:1805.08913*, 2018.
- M. Simonovsky and N. Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pages 412–422. Springer, 2018.
- K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, pages 1–8, 2014.
- T. Tabouy, P. Barbillon, and J. Chiquet. Variational inference for stochastic block models from sampled data. *Journal of the American Statistical Association*, 115(529):455–466, 2020.
- A. Vahdat and J. Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.
- Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *International conference on machine learning*, pages 3881–3890. PMLR, 2017.
- C. Zhang, J. Bøtche, H. Kjellström, and S. Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.
- S. Zhao, J. Song, and S. Ermon. Infvae: Balancing learning and inference in variational autoencoders. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 5885–5892, 2019.

## A Appendix

We formulate the variational approximation for the mixture of Gaussians in Equation 19 as follows:

$$q(\mu, c) = \prod_{j=1}^K q(\mu_j; m_j, s_j^2) \prod_{i=1}^N q(c_i; \phi_i). \quad (31)$$

From Equation 20, we have the following definition of ELBO:

$$\begin{aligned} \text{ELBO}(m, s^2, \phi) &= \mathbb{E}[\log p(x, \mu, c)] \\ &\quad - \mathbb{E}[\log q(\mu, c)], \end{aligned} \quad (32)$$

$$\text{ELBO}_{m, s^2, \phi} = \text{ELBO}(m, s^2, \phi),$$

where expectations are taken under  $q$  and  $m, s$  and  $\phi$  are the variational parameters.

In order to derive the optimal values of the variational parameters, we will first have to express the ELBO in Equation 32 in terms of  $m, s$  and  $\phi$ . We start with simplifying the first term,  $\log p(x, \mu, c)$ , on the RHS of Equation 32 as follows,

$$\begin{aligned} \log p(x, \mu, c) &= \log p(x|\mu, c)p(\mu, c), \\ &= \log p(x|\mu, c)p(\mu)p(c), \\ &= \log p(x|\mu, c) + \log p(\mu) + \log p(c), \\ &= \sum_j \log p(\mu_j) + \sum_i [\log p(c_i) \\ &\quad + \log p(x_i|c_i, \mu)], \end{aligned} \quad (33)$$

where  $p(c_i) = \frac{1}{K}$  is a constant and expanding  $p(\mu_j)$  we have,

$$\begin{aligned} \log p(\mu_j) &= \log \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{\mu_j^2}{2\sigma^2} \right] \right\}, \\ &\propto -\frac{\mu_j^2}{2\sigma^2}. \end{aligned} \quad (34)$$

For  $p(x_i|c_i, \mu)$ , in Equation 33, we can make use of the fact that  $c_i$  is a one-hot vector. Therefore,  $\log p(x_i|c_i, \mu)$  can be expressed as:

$$\begin{aligned} \log p(x_i|c_i, \mu) &= \log \prod_j p(x_i|\mu_j)^{c_{ij}}, \\ &= \sum_j c_{ij} \log p(x_i|\mu_j). \end{aligned} \quad (35)$$

In Equation 16 we define actual density function for the real-valued data-point  $x_i$  as follows:

$$\begin{aligned} \log p(x_i|\mu_j) &= \log \left\{ \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{(x_i - \mu_j)^2}{2} \right] \right\}, \\ &\propto -\frac{(x_i - \mu_j)^2}{2}. \end{aligned} \quad (36)$$

We now re-write Equation 33 by combining the derivations from equations 34, 35 and 36 as,

$$\begin{aligned} \log p(x, \mu, c) &\propto \sum_j -\frac{\mu_j^2}{2\sigma^2} \\ &\quad + \sum_i \sum_j -c_{ij} \frac{(x_i - \mu_j)^2}{2}. \end{aligned} \quad (37)$$

We now factorize the variational joint probability density  $q(\mu, c)$  in Equation 32 as,

$$\begin{aligned} \log q(\mu, c) &= \log \prod_j q(\mu_j; m_j, s_j^2) \prod_i q(c_i; \phi_i), \\ &= \sum_j \log q(\mu_j; m_j, s_j^2) + \sum_i \log q(c_i; \phi_i). \end{aligned} \quad (38)$$

We further expand the terms on the RHS of Equation 38 as follows,

$$\begin{aligned} \log q(\mu_j; m_j, s_j^2) &= \log \left\{ \frac{1}{\sqrt{2\pi s_j^2}} \exp \left[ -\frac{(\mu_j - m_j)^2}{2s_j^2} \right] \right\}, \\ &= -\frac{1}{2} \log(2\pi s_j^2) - \frac{(\mu_j - m_j)^2}{2s_j^2}. \end{aligned} \quad (39)$$

$$\begin{aligned} \log q(c_i; \phi_i) &= \log \prod_j \phi_{ij}, \\ &= \sum_j \log \phi_{ij}. \end{aligned} \quad (40)$$

We combine the derivations for the joint variational probability density from equations 39 and 40 to re-write the Equation 38 as,

$$\begin{aligned} \log q(\mu, c) &= \frac{1}{2} \sum_j \left[ -\log(2\pi s_j^2) - \frac{(\mu_j - m_j)^2}{2s_j^2} \right] \\ &\quad + \sum_i \sum_j \log \phi_{ij}. \end{aligned} \quad (41)$$

The final step towards deriving the ELBO in terms of the variational parameters is to factor the results from Equation 37 and 41 into Equation 32 as follows,

$$\begin{aligned} \text{ELBO}_{m, s^2, \phi} &\propto \sum_j -\mathbb{E} \left[ \frac{\mu_j^2}{2\sigma^2} \right] \\ &\quad + \sum_i \sum_j \mathbb{E} \left[ -c_{ij} \right] \mathbb{E} \left[ \frac{(x_i - \mu_j)^2}{2} \right] \\ &\quad - \frac{1}{2} \sum_j \mathbb{E} \left[ -\log(2\pi s_j^2) - \frac{(\mu_j - m_j)^2}{s_j^2} \right] \\ &\quad - \sum_i \sum_j \mathbb{E} \left[ \log \phi_{ij} \right]. \end{aligned}$$

The final ELBO objective in terms of the variational parameters is as follows:

$$\begin{aligned} \text{ELBO}_{m,s^2,\phi} &\propto \sum_j -\mathbb{E} \left[ \frac{\mu_j^2}{2\sigma^2} \right] \\ &+ \sum_i \sum_j \mathbb{E} \left[ -\phi_{ij} \right] \mathbb{E} \left[ \frac{(x_i - \mu_j)^2}{2} \right] \\ &+ \frac{1}{2} \sum_j \mathbb{E} \left[ \log(s_j^2) \right] \\ &- \sum_i \sum_j \mathbb{E} \left[ \log \phi_{ij} \right], \end{aligned} \quad (42)$$

where all expectations are taken under  $q$ .

Now, to derive the optimal values of the variational parameters we take partial derivatives of the ELBO in Equation 42 with respect to the variational parameters and equate them to zero.

**Deriving  $\phi_{ij}^*$ , the optimal value of  $\phi_{ij}$ :**

$$\begin{aligned} \frac{\partial}{\partial \phi_{ij}} \text{ELBO}_{m,s^2,\phi} &\propto \frac{\partial}{\partial \phi_{ij}} \left\{ -\phi_{ij} \mathbb{E} \left[ \frac{(x_i - \mu_j)^2}{2} \right] \right. \\ &\quad \left. - \mathbb{E} \left[ \log \phi_{ij} \right] \right\}, \\ &\propto \frac{\partial}{\partial \phi_{ij}} \left\{ -\phi_{ij} \mathbb{E} \left[ \frac{(x_i - \mu_j)^2}{2} \right] \right. \\ &\quad \left. - \phi_{ij} \log \phi_{ij} \right\}, \\ &\propto \mathbb{E} \left[ -\frac{(x_i - \mu_j)^2}{2} \right] - \log \phi_{ij} - 1, \\ &\propto -\frac{1}{2}(m_j^2 + s_j^2) + x_i m_j - \log \phi_{ij}. \end{aligned}$$

We derive the optimal value of  $\phi_{ij}$ , as follows:

$$\begin{aligned} 0 &= \frac{\partial}{\partial \phi_{ij}} \text{ELBO}_{m,s^2,\phi}, \\ \phi_{ij}^* &\propto \exp \left[ -\frac{1}{2}(m_j^2 + s_j^2) + x_i m_j \right] \end{aligned}$$

**Deriving  $m_j^*$ , the optimal value of  $m_j$ :**

$$\begin{aligned} \frac{\partial}{\partial m_j} \text{ELBO}_{m,s^2,\phi} &\propto \frac{\partial}{\partial m_j} \left\{ -\sum_i \phi_{ij} \mathbb{E} \left[ \frac{(x_i - \mu_j)^2}{2} \right] \right. \\ &\quad \left. - \mathbb{E} \left[ \frac{\mu_j^2}{2\sigma^2} \right] \right\} \\ &\propto \frac{\partial}{\partial m_j} \left\{ \sum_i \phi_{ij} \left[ -\frac{1}{2}(m_j^2 + s_j^2) + x_i m_j \right] \right. \\ &\quad \left. - \frac{1}{2\sigma^2}(m_j^2 + s_j^2) \right\}, \\ &\propto \frac{\partial}{\partial m_j} \left\{ \sum_i \left[ -\frac{1}{2}\phi_{ij} m_j^2 + \phi_{ij} x_i m_j \right] \right. \\ &\quad \left. - \frac{1}{2\sigma^2} m_j^2 \right\}, \\ &\propto \sum_i \left[ \phi_{ij} m_j + \phi_{ij} x_i \right] - \frac{m_j}{2\sigma^2} \end{aligned}$$

We derive the optimal value of  $m_j$ , as follows:

$$\begin{aligned} 0 &= \frac{\partial}{\partial m_j} \text{ELBO}_{m,s^2,\phi}, \\ m_j^* &= \frac{\sum_i \phi_{ij} x_i}{\frac{1}{\sigma^2} + \sum_i \phi_{ij}}. \end{aligned}$$

**Deriving  $(s_j^2)^*$ , the optimal value of  $s_j^2$ :**

$$\begin{aligned} \frac{\partial}{\partial s_j^2} \text{ELBO}_{m,s^2,\phi} &\propto \frac{\partial}{\partial s_j^2} \left\{ -\sum_i \phi_{ij} \mathbb{E} \left[ \frac{(x_i - \mu_j)^2}{2} \right] \right. \\ &\quad \left. - \mathbb{E} \left[ \frac{\mu_j^2}{2\sigma^2} \right] + \frac{1}{2} \mathbb{E} \left[ \log(s_j^2) \right] \right\}, \\ &\propto \frac{\partial}{\partial s_j^2} \left\{ \sum_i \phi_{ij} \left[ -\frac{1}{2}(m_j^2 + s_j^2) + x_i m_j \right] \right. \\ &\quad \left. - \frac{1}{2\sigma^2}(m_j^2 + s_j^2) + \frac{1}{2} \log(s_j^2) \right\}, \\ &\propto \frac{\partial}{\partial s_j^2} \left\{ -\sum_i \frac{1}{2} \phi_{ij} s_j^2 - \frac{1}{2\sigma^2} s_j^2 + \frac{1}{2} \log s_j^2 \right\}, \\ &\propto -\sum_i \frac{1}{2} \phi_{ij} - \frac{1}{2\sigma^2} + \frac{1}{2s_j^2}. \end{aligned}$$

We derive the optimal value of  $s_j^2$ , as follows:

$$\begin{aligned} 0 &= \frac{\partial}{\partial s_j^2} \text{ELBO}_{m,s^2,\phi}, \\ (s_j^2)^* &= \frac{1}{\frac{1}{\sigma^2} + \sum_i \phi_{ij}}. \end{aligned}$$