Epanechnikov Variational Autoencoder

Tian Qin
Math department
Lehigh University
Bethlehem, PA 18015
tiq218@lehigh.edu

Wei-Min Huang
Math department
Lehigh University
Bethlehem, PA 18015
wh02@lehigh.edu

Abstract

In this paper, we bridge Variational Autoencoders (VAEs) [17] and kernel density estimations (KDEs) [25],[23] by approximating the posterior by KDEs and deriving an upper bound of the Kullback-Leibler (KL) divergence in the evidence lower bound (ELBO). The flexibility of KDEs makes the optimization of posteriors in VAEs possible, which not only addresses the limitations of Gaussian latent space in vanilla VAE but also provides a new perspective of estimating the KL-divergence in ELBO. Under appropriate conditions [9],[3], we show that the Epanechnikov kernel is the optimal choice in minimizing the derived upper bound of KL-divergence asymptotically. Compared with Gaussian kernel, Epanechnikov kernel has compact support which should make the generated sample less noisy and blurry. The implementation of Epanechnikov kernel in ELBO is straightforward as it lies in the "location-scale" family of distributions where the reparametrization tricks can be directly employed. A series of experiments on benchmark datasets such as MNIST, Fashion-MNIST, CIFAR-10 and CelebA further demonstrate the superiority of Epanechnikov Variational Autoenocoder (EVAE) over vanilla VAE in the quality of reconstructed images, as measured by the FID score and Sharpness[27].

1 Introduction

In variational inference, an autoencoder learns to encode the original data x using learnable function $f_\phi^{-1}(x)$ and then to reconstruct x using a decoder g_θ. [17] proposed a stochastic variational inference and learning algorithm called Variational Autoencoders (VAEs) which can be further used to generate new data. According to VAE, a lower bound on the empirical likelihood, which is known as ELBO, is maximized so that the fitted model $p_\theta(x|z) = \mathcal{N}(x; g_\theta(x), I_D)$ and $q_\phi(z|x) = \mathcal{N}(x; f_\phi(x), I_D)$ can approximate the true conditional distributions $p(x|z)$ and $p(z|x)$, respectively. As illustrated in [17] employing Guassian as a default choice for prior and posterior distribution is mathematically convenient since the corresponding ELBO is analytic. Due to the flexibility of data type nowadays, many variants of VAEs have been proposed to enrich the expressibility of latent space. There are three main directions for extending VAEs:

Approximate posterior $q_\phi(z|x)$**:**

Normalizing flow (NF) [24] applies a sequence of invertible transformations to initial density $q_0(z_0)$ to achieve more expressive posteriors. β-VAE [14] introduced a new parameter β in balancing the reconstruction loss with disentangled latent representation $q_\phi(z|x)$. Importance weighted Autoencoders (IWAE) [5] improved log-likelihood lower bound by importance weighting. It also approximated the posterior with multiple samples and enriched latent space representations. However, while pursuing more flexible posterior, most literature along this direction don't modify the standard normal prior assumption for the sake of analytic loss function.

Prior distribution of latent variable $p(\mathbf{z})$**:**

Instead of approximating posterior, we can replace Gaussian prior with other flexible distributions. For instance, [8] used a simple Gaussian mixture prior and [28] introduced a mixture prior called "VampPrior" which consists of a mixture distribution with components given by variational posteriors. The possibility of non-parametric priors is explored in [22], which utilized a truncated stick-breaking process. [12] and [6] attempted to replace Gaussian prior by von Mises-Fisher(vMF) distribution. The intuition is that Guassian distribution may have limited coverage if the true latent space is hyperspherical. [6] also pointed out that, as a special case of vMF, the uniform prior on the hypersphere allows the even spread of data concentration with no directional bias.

Manifold learning The vanilla VAE [17] assumes the Euclidean latent space. In more complex data structures, the flatness of Euclidean has hard time to capture the geometric property of latent space. [16] enriched structures of VAE by replacing Gaussian prior with a Riemannian Brownian motion prior. Utilizing the invertible transformations, NF can be connected with the manifold learning of latent space such as Pseudo-invertible encoder(PIE) [2], Manifold flow ($\mathcal{M}$-flow) [4] and Denosing Normalizing Flow(DNF) [15]. They all model the VAE as a manifold learning procedure. Again, the flexibility of those methods is still restricted by the choice of variational family since they all assumed that the distribution of variables in latent spaces and noises in manifolds are all Gaussian.

It's now natural to ask the following question:

- In which sense of optimality there exists an optimal functional form of posteriors given certain technical conditions?

In this paper, we answer this question by the following steps, which are also the main contributions of our work:

1. We first formulate the latent space learning process in VAE as a problem of kernel density estimation.

2. Inspired by the results in [3] , we approximate the posterior in KL-divergence by a kernel density estimator and establish some asymptotic results based on the conclusion in [3]. See details in section 2.2.

3. After that, we utilize the conclusion from [3] and show that the optimal kernel in minimizing an upper bound of KL-divergence is Epanechnikov kernel, which is simple and easy to implement. The derivation connects the expectation of a statistic with KL-divergence, which to our best knowledge, is the first attempt to bridge the concept of asymptotic distribution of KDEs with VAE.

4. Thanks to the reparametrization trick, the implementation of Epanechnikov kernel in VAE is also straightforward. We conduct a detailed comparison of the performance of Epanechnikov VAE (EVAE) and Gaussian VAE in many benchmark image datasets under standard encoder-decoder structure. The experiments indeed demonstrate the superiority of Epanechnikov kernel over vanilla VAE.

The remaining sections of this paper are organized as following schema. In section 2, we provide some preliminaries involving VAE, kernel density estimation and few assumptions. In section 3, we derive the optimal kernel in bounding the KL-divergence is Epanechnikov kernel. Based on results in section 3, we propose the Epanechnikov VAE (EVAE) in section 4. The comparison between EVAE and vanilla VAE in benchmark datasets is illustrated in section 5. Section 6 discusses few characteristics and limitations of EVAE and suggests some future directions. The pytorch codes for the implementation of EVAE and experiments are available in supplementary materials. All simulations and experiments are performed on a laptop with 12th Gen Intel(R) Core(TM) i7-12700H (2.30 GHz) ,16.0 GB RAM and NVIDIA 4070 GPU.

2 Preliminary

2.1 VAE formulation

Consider a dataset $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$ consists of n $i.i.d$ samples from space $\mathcal{X}$ whose dimension is d. In VAE, we assume that every observed data $\mathbf{x}^{(i)} \in \mathcal{X}$ is generated from a latent variable $\mathbf{z}^{(i)} \in \mathcal{Z}$ whose dimension is p. Then the data generation process can be summarized in 2 steps. It first

produces a variable $\mathbf{z}^{(i)}$ from some parametric prior distribution $p_\theta(\mathbf{z})$. Then given the value of $\mathbf{z}^{(i)}$, an observed value $\mathbf{x}^{(i)}$ is generated from certain conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$. We typically assume $d > p$ and prior $p(\mathbf{z})$ and likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ are differentiable parametric distributions w.r.t θ and $\mathbf{z}$.

To maximize the empirical log likelihood $\log p_\theta(\mathbf{x})$, we need to evaluate the intractable integral in the form

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z}) dz = \log \int p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) dz$$

which is not available in most cases. Fortunately, we can instead maximizing its log evidende lower bound (ELBO) $\mathcal{L}$ with the help of Jensen's inequality:

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z}) dz = \log \int q_\phi(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} dz \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] := \mathcal{L}(\theta, \phi)$$

Note that $\mathcal{L}(\theta, \phi)$ can be rewritten as follows:

$$
\begin{aligned}
\mathcal{L}(\theta, \phi) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))
\end{aligned}
\tag{1}
$$

where $KL(\cdot||\cdot)$ represents the KL-divergence between two distributions.

The conditional likelihood $p_\theta(\mathbf{x}|\mathbf{z})$, approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ and the prior distribution $p(\mathbf{z})$ can be chosen independently. For convenience, most applications of VAE employ Gaussian parametrization for all three likelihoods. Since we'd like to investigate optimal posterior and prior rather than the form of conditional likelihood $p_\theta(\mathbf{x}|\mathbf{z})$, we can assume a multivariate Bernoulli or Gaussian model w.r.t $p_\theta(\mathbf{x}|\mathbf{z})$ for simplicity. For example, under multivariate Gaussian, we have $\log p_\theta(\mathbf{x}|\mathbf{z}) = \log \mathcal{N}(\mathbf{x}; g_\theta(\mathbf{z}), I)$. As to multivariate Bernoulli model, we have $\log p_\theta(\mathbf{x}|\mathbf{z}) = \sum_i [\mathbf{x}_i \log g_\theta(\mathbf{z}) + (1 - \mathbf{x}_i) \log(1 - g_\theta(\mathbf{z}))]$ where $g_\theta(\mathbf{z}) : \mathcal{Z} \to \mathcal{X}$ are typically neural-network parametrizations.

To maximize ELBO, we now need to minimize the following target function:

$$\mathbb{E}_{\substack{x \sim p \\ z \sim q_\phi(\cdot|\mathbf{x})}} [-\log p_\theta(\mathbf{x}|\mathbf{z}) + KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] \tag{2}$$

The two terms in equation (2) are named as "reconstruction error" and "divergence" or "regularization term", respectively. The "divergence" term regularizes the mismatch between approximate posterior and prior distribution.

2.2 Kernel density estimation

2.2.1 Approximate the posterior in KL-divergence by kernel functions

A common assumption of latent space in VAE is the factorization of approximate posterior, i.e. dimensions/features of latent space are independent with each other. Under this assumption, we only need to consider the formulation of kernel density estimation in one-dimensional case. Similar arguments can be applied to mulit-dimensional cases by additivity of KL-divergence under independence,

Given $X_1, ..., X_n$ be i.i.d random variables with a continuous density function f, [23] and [25] proposed kernel density density estimate $f_n(x)$ for estimating $f(x)$ at a fixed point $x \in \mathbb{R}$,

$$f_n(x) = \frac{1}{nb_n} \sum_{i=1}^{n} K\left[\frac{x - X_i}{b_n}\right] = \frac{1}{b_n} \int K\left[\frac{x - t}{b_n}\right] dF_n(t) \tag{3}$$

where F_n is the sample distribution function, K is an appropriate kernel function such that $\int K(x) dx = 1$ and the positive number b_n is called bandwidth such that $b_n \to 0, nb_n \to \infty$ as $n \to \infty$[1].

---

[1]We omit the limits of integrals if they are $-\infty$ to ∞.

Let the kernel function be

$$K(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

then the Gaussian VAE is a reminiscent of a kernel density estimator. More specifically, in one-dimensional case we can rewrite the approximate posterior $q_\phi(z|\mathbf{x}^{(i)})$ at data point $x^{(i)}$ as follows:

$$q_\phi(z|x^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma_\phi(\mathbf{x}^{(i)})}\exp\left(\frac{(z-\mu_\phi(\mathbf{x}^{(i)}))^2}{2\sigma_\phi^2}\right)$$

where $\mu_\phi : \mathcal{X} \to \mathcal{Z}$ and $\sigma_\phi : \mathcal{X} \to \mathcal{Z}^+$ are neural network parametrizations of the conditional mean and variance of $q_\phi(z|x^{(i)})$. $\mathcal{Z}^+$ represents the positive region for latent space $\mathcal{Z}$. Note that the bandwidth $\sigma_\phi(x^{(i)})$ is now learnable.

On the other hand, by Taylor expansion of KL-divergence, we can bound the KL-divergence by a quadratic functional which is the first order approximation of KL-divergence. We present the claim in following Lemma 2.1

Lemma 2.1 *Suppose continuous density functions $p(z)$ and $q(z)$ satisfy*

$$\max_{z\in\mathcal{D}_1\cap\mathcal{D}_2} |(q(z)-p(z))/p(z)| \le C$$

where $\mathcal{D}_1$ is the domain of $p(z)$, $\mathcal{D}_2$ is the domain of $q(z)$ and $0 < C \le 1$. then we have

$$KL(q(z)||p(z)) \le \tilde{C} \int \frac{(q(z)-p(z))^2}{p(z)}dz$$

where $\tilde{C}$ is a positive constant.

The uniform bound condition in Lemma 2.1 restricts the local deviation of posterior and prior. This is consistent with the intuition that the KL-term penalizes the mismatches between two distributions. If we model the posterior $q_\phi(z|\mathbf{x}^{(i)})$ as the expectation of a kernel density estimator of $p(\mathbf{z})$, by Lemma 2.1 and Cauchy–Schwarz inequality, we obtain

$$KL(q_\phi(z|\mathbf{x}^{(i)})||p(z)) \le \tilde{C} \int \frac{[\mathbb{E}_0(q_{m,\phi}^{(i)}(z))-p(z)]^2}{p(z)}dz \le \tilde{C}\mathbb{E}_0\left[\int \frac{(q_{m,\phi}^{(i)}(z)-p(z))^2}{p(z)}dz\right] \quad (4)$$

where

$$q_{m,\phi}^{(i)}(z) = \frac{1}{mb(m)}\sum_{j=1}^{m} K_{\phi,\mathbf{x}^{(i)}}\left[\frac{z-\mathbf{Z}_j}{b(m)}\right], \quad \mathbf{Z}_j \overset{i.i.d}{\sim} p(\mathbf{z})$$

is a kernel density estimator of $q_\phi(z|\mathbf{x}^{(i)})$ and expectation $\mathbb{E}_0$ is take w.r.t the samples from prior distribution $p(\mathbf{z})$. The subscript ϕ of Kernel function implies that the parameters in Kernel can be learned by neural networks. Our main goal is to find the optimal Kernel function $K_{\phi,\mathbf{x}^{(i)}}$ with other model parameters ϕ and data point fixed. For now, we interpret m as the number of latent variable Z generated from prior distribution which should be distinguished from original sample size n. In the estimation theory, [9] showed that the Epanechnikov kernel minimizes $\int \mathbb{E}_0[q_{m,\phi}^{(i)}(\mathbf{z})-p(\mathbf{z})]^2d\mathbf{z}$ asymptotically, which gives us a hint of the potential optimization of kernel K in quantity (4). Before we derive the optimal kernel in VAE, we 'd like to briefly introduce some main assumptions and notations in following section.

2.2.2 Assumptions

We mainly borrow the terminologies and assumptions in [25] which is the pioneering work in measuring deviations of density function estimates. Note that [25] studied the asymptotic distribution of the quadratic functional $\int[\hat{f}_n(x) - f(x)]^2 a(x)dx$ under appropriate weight function a and conditions as sampling size n approaches to infinity. In (4), we just saw that the weight function is the reciprical of prior density in our case.

Assumptions **A1** − **A4** are listed as follows:

(A1): The kernel function K is bounded, integrable, symmetric (about 0) and $\int K(x)dx = 1, \int x^2 K(x)dx < \infty, \int K^2(x)dx < \infty$. Also, K either (a) is supported on an closed and bounded interval $[-B, B]$ and is absolutely continuous on $[-B, B]$ with derivative K' or (b) is absolutely continuous on the whole real line with derivative K' satisfying $\int |K'(x)|^k dx < \infty, k = 1, 2$. Moreover,

$$\int_{x \geq 3} |x|^{\frac{3}{2}} [\log\log|x|]^{\frac{1}{2}} [|K'(x)| + |K(x)|] dx < \infty$$

(A2): The underlying density f is continuous, positive and bounded.

(A3): Squared density $f^{1/2}$ is absolutely continuous and its derivative is bounded in absolute value.

(A4): The second derivative f'' exists and is bounded.

We can see that the Gaussian kernel satisfy those assumptions, indicating that the kernel density estimation theory also works for Gaussian VAE. Similarly, most priors and posteriors we listed in section 1 lie in conditions $\mathbf{A1} - \mathbf{A4}$, demonstrating the promising applications of kernel estimate theory in variants of VAE.

3 Optimal kernel of VAE

Let m be the number of latent variable Z generated from prior distribution, as defined in 4. Let $f_m(x)$ be a kernel density estimate of a continuous density function f at x, as defined in (3), we construct a statistic T_m as follows:

$$T_m = mb(m) \int [f_m(x) - f(x)]^2 a(x)dx$$

where $a(x)$ is an appropriate weight function. We now restate the main result of [3] as Theorem 3.1:

Theorem 3.1 ([3]) *Let $\mathbf{A1} - \mathbf{A4}$ hold and suppose that the weight function a is integrable piecewise continuous and bounded. Suppose $b(n) = o(n^{-\frac{2}{9}})$ and $o(b(n)) = n^{-\frac{1}{4}}(log(n))^{\frac{1}{2}}(loglogn)^{\frac{1}{4}}$ as $n \to \infty$, then $b^{-\frac{1}{2}}(n)(T_n - I(K)\int f(x)a(x)dx)$ is asymptotically normally distributed with mean 0 and variance $2J(K)\int a^2(x)f^2 dx$ as $n \to \infty$, where*

$$I(K) = \int K^2(z)dz, \quad J(K) = \int \left[\int K(z+y)K(z)dz\right]^2 dy \tag{5}$$

In other words, under Theorem 3.1, we have

$$\mathbb{E}[T_m] \to I(K) \int f(x)a(x)dx \quad , \text{as } m \to \infty$$

Let $mb(m)/n = \tilde{C}$ where $\tilde{C}$ is the constant in Lemma 2.1 and $b(m)$ satisfies the conditions in Theorem 3.1, by the asymptotic result of Theorem 3.1, the inequality (4) becomes

$$
\begin{aligned}
KL(q_\phi(z|\mathbf{x}^{(i)})||p(z)) &\leq \frac{mb(m)}{n} \mathbb{E}_0 \left[\int \frac{(q_{m,\phi}^{(i)}(z) - p(z))^2}{p(z)} dz \right] \\
&= \frac{\mathbb{E}[T_m]}{n} \overset{\text{n large}}{\approx} \frac{I(K_{\phi,\mathbf{x}^{(i)}}) \int p(z)\frac{1}{p(z)}dz}{n} = B^{(i)} \frac{I(K_{\phi,\mathbf{x}^{(i)}})}{n}
\end{aligned}
\tag{6}
$$

where $B^{(i)}$ is the length of support interval of $p(z)$ under i-th datapoint. Note that if the prior has infinite length of support, the inequality 6 becomes useless. Instead of minimizing the analytic form of KL-divergence, we can now find an optimal kernel function $I(K_{\phi,\mathbf{x}^{(i)}})$ which minimizes $I(K_{\phi,\mathbf{x}^{(i)}})$ given fixed parameters and data point. Lemma 3.2 shows that Epanechnikov kernel is the optimal choice.

Lemma 3.2 *Let $\mathcal{K}$ be the set of all $L^1(-\infty, \infty)$ functions K satisfying*

$$K \geq 0, \quad \int K(x)dx = 1, \quad \int xK(x)dx = \mu, \int (x-\mu)^2 K(x)dx = \frac{1}{5}r^2$$

where $\mu \in (-\infty, \infty), r > 0$. Then the functional $I(K)$ in (5) is minimized on $\mathcal{K}$ uniquely by

$$K^*(x) = \begin{cases} \frac{3}{4r}\left(1 - \left(\frac{x-\mu}{r}\right)^2\right) & x \in [\mu - r, \mu + r] \\ 0 & otherwise \end{cases}$$

and $\min_K I(K) = I(K^*) = \frac{3}{5r}$. The optimal kernel K^* is named as Epanechnikov kernel[9].

The proof is based on the idea of Lagrange multiplier. Few observations from Lemma 3.2 and Figure S1 in Appendix A.2 (plots of standard Epanechnikov kernel ($\mu = 0, r = 1$) and standard Gaussian) :

1. The support of Epanechnikov kernel is closed and bounded (i.e compact). However, the support of Gaussian distribution is unbounded, which may lead to the noisy or blurry regenerated images. Thus we posit that Epanechnikov kernel could regenerate sharper regenerated. Empirical evidence is reported in section 5. Note that we add $1/5$ just for simplicity of the results.

2. The distribution function of Epanechnikov kernel lies in the "location-scale" family as well, which indicates that the implementation of Epanechnikov kernel VAE is simple and straightforward by reparametrization tricks. See section 4 for details.

3. $I(K^*)$ doesn't include μ, indicating that μ is only optimized in the reconstruction term in ELBO. The parameter $B^{(i)}$ in (6) controls the support of prior distribution. If we set $B^{(i)}$ as a constant all the time, then EVAE can be connected to β-VAE [14] which adds a weight parameter β in front of the KL term.

4 Epanechnikov VAE

Getting inspired by the optimality of Epanechnikov kernel in controlling KL-divergence, we propose the Epanechnikov Variational Autoencoder (EVAE) whose resampling step is based on Epanechnikov kernel. There are two main differences between EVAE and VAE. The latent distribution in EVAE is assumed to be estimated by the Epanechnikov kernel rather than multivariate isotropic Gaussian. And EVAE is trained to minimize a different target function (7), which is an upper bound of the one (2) used in ordinary VAE.

$$\mathbb{E}_{\substack{\mathbf{x} \sim p \\ z \sim q_\phi(\cdot|\mathbf{x})}} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) + \mathbf{B}\frac{I(K^*_\phi)}{n} \right] \tag{7}$$

where n can be the sample size or minibatch size, $\mathbf{B}, \phi$ are outputs of the encoding network and K^* is Epanechnikov kernel. The sample version of (7) at datapoint $\mathbf{x}^{(i)}$ is

$$\tilde{\mathcal{L}}(\theta, \phi, \mathbf{B}; \mathbf{x}^{(i)}) \approx \frac{1}{L}\sum_{l=1}^{L}(\log p_\theta(\mathbf{x}^{(i)})|\mathbf{z}^{(i,l)}) + \sum_{d=1}^{D}\left(\frac{3}{5M} \cdot \frac{B_d^{(i)}}{r_d^{(i)}}\right) \tag{8}$$

where D is the dimension for latent space, $\mathbf{B}^{(i)} = (B_1^{(i)}, ..., B_D^{(i)})$, $\mathbf{r}^{(i)} = (r_1^{(i)}, ..., r_D^{(i)})$ are outputs of the encoding networks with variational parameters ϕ and M is minibatch size. In vanilla VAE, [17] suggested that the number of regenerated samples L can be set to 1 as long as the minibatch size is relatively large, which is also the case for EVAE. To sample from the posterior, we need to derive the density of $q_\phi(z|\mathbf{x}^{(i)})$:

$$\begin{aligned} q_\phi(z|\mathbf{x}^{(i)}) &= \mathbb{E}\left[\frac{1}{mb(m)}\sum_{j=1}^{m} K^*_{\phi,\mathbf{x}^{(i)}}\left(\frac{z - \mathbf{Z}_j}{b(m)}\right)\right] \\ &= \mathbb{E}\left[\frac{1}{b(m)}K^*_{\phi,\mathbf{x}^{(i)}}\left(\frac{z - \mathbf{Z}_1}{b(m)}\right)\right] \\ &= \int \frac{1}{b(m)}K^*_{\phi,\mathbf{x}^{(i)}}\left(\frac{z - \tilde{z}}{b(m)}\right)f_{\mathbf{z}}(\tilde{z})d\tilde{z} \end{aligned} \tag{9}$$

where $f_{\mathbf{z}}$ is the density of prior distribution. Given a datapoint $\mathbf{x}^{(i)}$, let $q_{\phi}(z|\mathbf{x}^{(i)})$ be the density of random variable $Z(\mathbf{x}^{(i)})$, $K^{*}_{\phi,\mathbf{x}^{(i)}}(z)$ be the density of random variable $K^{*}(\mathbf{x}^{(i)})$ and prior $f_{\mathbf{z}}$ be the random variable of Z, then it's clear to see that the posterior is now a convolution between two random variables, i.e

$$Z(\mathbf{x}^{(i)}) = b(m)K^{*}(\mathbf{x}^{(i)}) + Z \tag{10}$$

Identity (10) decomposes the posterior into two parts. One is the prior information, the other represents the incremental updates from new information ($\mathbf{x}^{(i)}$). The Epanechnikov kernel can be viewed as the "optimal" direction of perturbing the prior distribution and the coefficient $b(m)$ can be interpreted as "step size". The resampling step in EVAE can be divided into two parts as well, as described in Algorithm 1. For the sake of finite support and simplicity, we assume that the prior distribution is uniformly distributed and step size $b(m)$ is the same for all dimensions. The theoretical conditions for $b(m)$ is demanding. In practice we found that setting step size as $b(100) = 100^{-2/9} \approx 0.3594$ is good enough.

Algorithm 1 Resampling step in a minibatch of EVAE

Require: Latent space dimension d_z; mean $\mu_{\phi}(\mathbf{x})$, spread $\mathbf{r}_{\phi}(\mathbf{x})$ and prior support interval length $\mathbf{B}(\mathbf{x})$, which are all learned by encoder networks. Minibatch size M. Step size $b(m)$.
 1: Sample a $M \times d_z$ matrix $\mathbf{U}$ where each (i,j) entry of the random matrix $\mathbf{U}$ is sampled from $\text{Unif}[-\mathbf{B}(\mathbf{x})_{i,j}, \mathbf{B}(\mathbf{x})_{i,j}]$.
 2: Sample a $M \times d_z$ matrix $\mathbf{K}$ where each (i,j) entry of the random matrix $\mathbf{K}$ is sampled from an standard Epanechnikov kernel supported on $[-1, 1]$.
 3: Shift and scale sampled $\mathbf{K}$ by the location-scale formula: $\mathbf{Z} = \mu_{\phi}(\mathbf{x}) + \mathbf{r}_{\phi}(\mathbf{x}) \odot \mathbf{K}$.
 4: **Return:** $b(m) \odot \mathbf{Z} + \mathbf{U}$

In Algorithm 1, we apply reparametrization trick to sample $\mathbf{z}$ from general Epanechnikov kernel, i.e $\mathbf{z}^{(i,l)} = \mu^{(i)} + \mathbf{r}^{(i)} \odot \mathbf{k}^{(l)}$, where $\mathbf{k}^{(l)}$ is sampled from standard Epanechnikov kernel supported on $[-1, 1]$ as it lies in the "location-scale" family. We use $\odot$ to signify the element-wise product. There are many ways to sample from standard Epanechnikov kernel, such as accept-rejection method. For efficiency, we provide a faster sampling procedure in Algorithm 2. The theoretical support is given in section A.3.

Algorithm 2 Sampling from centered Epanechnikov kernel supported on $[-1, 1]$

 1: Sample $U_1, U_2, U_3 \overset{i.i.d}{\sim} \text{Unif}[-1, 1]$.
 2: Set $U = \text{Median}(U_1, U_2, U_3)$
 3: **Return:** U

As we mentioned in section 3, the regularization term in (7) doesn't include $\mu_{\phi}(\mathbf{x})$. It is only optimized in the reconstruction term in (8). In Gaussian VAE, the prior distribution has infinite support which limits the capability of latent space while the extra parameter $\mathbf{B}^{(i)}$ induced by KDEs controls the support of prior distribution, which makes EVAE more flexible. To maximize (8) empirically, the spread parameter $\mathbf{r}$ tends to be small, which may lead to numerical issue in back propagation. However, the numerator $\mathbf{B}^{(i)}$ on the second part of equation 8 plays an rule to stabilize scaling effect in (8). What's more, if we set $B^{(i)}$ as a constant all the time, then EVAE can be connected to β-VAE [14] which adds a weight parameter β in front of the KL term to control the disentanglement of learned model. We leave the potential disentanglement learned by EVAE as future work.

5 Experiments

In this section, we will compare the proposed EVAE model with (vanilla) VAE whose posterior and prior are modelled by isotropic multivariate Gaussian in real datasets. To assess the quality of reconstructed images, we employ the Frechet Inception Distance (FID) score [13] to measure the distribution of generated images with the distribution real images. The lower, the better. Inception_v3

[26] is employed as default model to generate features of input images, which is a standard implementation in generative models. In section 3, we mentioned that the compact support for Epanechnikov kernel could help EVAE generate less blurry or noisy images. To check this claim empirically, we calculated sharpness[27] for generated images by a 3×3 Laplace filter. See details in Appendix C.

We trained EVAE and VAE on four benchmark datasets: MNIST[7], Fashion-MNIST[29], CIFAR-10[18] and CelebA[20]. The detailed information for training parameters are attached in Appendix C. For comparison, we applied a classical CNN architecture (Appendix B) in encoding and decoding part for all datasets. Consequently, the main differences between EVAE and VAE in implementation stem from the resampling step and target function in training procedure. All FID scores and sharpness are based on hold-out samples. We also evaluated VAE and EVAE with different dimensions d_z of latent spaces. Table 1 and Table 2 summarize the results on four datasets.

Table 1: VAE and EVAE results on MNIST and Fashion-MNIST datasets

Datasets	MNIST				Fashion-MNIST			
	VAE		EVAE		VAE		EVAE	
d_z	FID	Sharpness	FID	Sharpness	FID	Sharpness	FID	Sharpness
8	16.39	0.0217	14.20	0.0245	41.33	0.0154	34.77	0.0186
16	12.66	0.0244	10.52	0.0314	39.01	0.0166	25.05	0.0222
32	12.57	0.0252	8.13	0.0340	37.66	0.0166	17.98	0.0250
64	12.53	0.0245	5.67	0.0359	39.49	0.0163	11.87	0.0276

Table 2: VAE and EVAE results on CIFAR10 and CelebA datasets

Datasets	CIFAR-10				CelebA			
	VAE		EVAE		VAE		EVAE	
d_z	FID	Sharpness	FID	Sharpness	FID	Sharpness	FID	Sharpness
8	229.2	0.0354	217.0	0.0343	189.4	0.0149	193.2	0.0145
16	182.9	0.0349	163.9	0.0346	145.2	0.0153	139.8	0.0159
32	147.2	0.0353	117.9	0.0358	100.8	0.0162	94.40	0.0160
64	146.6	0.0356	79.78	0.0360	76.48	0.0165	58.84	0.0168

In terms of FID score, we observe that EVAE has an edge in high dimensions ($d_z = 32, 64$) , indicating the better quality of regenerated images from EVAE. Additionally, when $d_z = 64$, EVAE generate larger sharpness of reconstructed images for all datasets, as illustrated in Table 1 and Table 2. This result empirically justifies the positive effect of having compact support in posteriors and priors. As to CIFAR-10, EVAE has larger sharpness in high dimensions ($d_z = 32, 64$) while it is mediocre in low dimensions. Possible reasons include low-resolution(blurriness) of original images and deficient expressibility of simple CNN models in low dimensions. But the validation reconstruction loss curves in Appendix E indeed authenticate the superiority of EVAE over benchmark datasets, including CelebA.

Appendix D presents figures for some test reconstructed samples from trained VAE and EVAE with $d_z = 64$. For MNIST and Fashion-MNIST, many images generated from EVAE are able to pick up local features better than the VAE. And reconstructed samples from CIFAR-10 are clearer and closer to the original images, as indicated by large the gap between FID scores. The statistical analysis from binomial test based on total experiments (Appendix C) shows that EVAE significantly outperforms VAE in FID and sharpness.

6 Discussion and Limitation

Approximating the posterior through KDEs, we have derived the optimal kernel in bounding the KL-divergence and built a novel kernel based VAE called Epanechnikov VAE. The finite support for Epanechnikov kernel addresses the issue of generating blurry images under Gaussian posterior. Experiments on benchmark datasets demonstrate the power of EVAE compared to vanilla VAE. The result of presented paper paves the way for various promising research directions for future works.

Connections between EVAE and β-VAE

When the length of support interval **B** is constant, the target function in 8 becomes

$$\tilde{\mathcal{L}}(\theta, \phi, \mathbf{B}; \mathbf{x}^{(i)}) \approx \frac{1}{L} \sum_{l=1}^{L} (\log p_\theta(\mathbf{x}^{(i)}) | \mathbf{z}^{(i,l)}) + \sum_{d=1}^{D} \left(\frac{3}{5} \cdot \frac{B}{M} \cdot \frac{1}{r_d^{(i)}} \right)$$

where the ratio $\frac{B}{M}$ now acts as a weight parameter in penalizing the kernel term. This formula is similar to the ELBO given in β-VAE. It would be interesting to compare the disentenglement and implicit regularization effect [19] of EVAE and β-VAE. In fact, in EVAE the weight parameter $\frac{B}{M}$ is derived from the global deviation result for KDEs (i.e. Theorem [3]), where the constant **B** should be interpreted as the length of support interval for the prior.

Geometric Interpretation of EVAE

Identity (10) provides another interpretation of VAE. By approximating the posterior with kernel densities, the posterior can be decomposed with the sum of prior and distribution of new information and the Epanechnikov kernel can be viewed as the optimal direction of updating posterior from the prior. This decomposition bridges EVAE and Information Geometry[1] which analyzes the relationship between probability distributions through modern differential geometry.

Applying Epanechnikov kernel in other generative models

Empirical results in section 5 exhibited the promising improvement of Epanechnikov kernel in VAE under classical CNN architecture. Extending the implementation of Epanechnikov kernel in more complicated generative models such as GAN[11], normalizing flow [15] and other variants of VAEs is one of our future works.

A more precise approximation of KL-divergence

In Lemma 2.1, we bounded the KL-term by first order approximation from Taylor expansion, which limits the more general cases for posteriors and priors. It's possible to derive sharper bounds with higher order approximation where the Epanechnikov kernel may not be the optimal one. Lemma 2.1 also assumes a small deviation between posterior and prior, which limits the candidates for prior distribution in practice.

Optimal kernel under different criterions

In this paper, we mainly focused on the L_2 deviation of KDEs, which is measured by the functional $I(K)$ proposed in Theorem 3.1. However, in the general theory of KDEs, different functionals correspond to different optimal kernels. For example, we didn't put too much attention on the convolution functional $J(K) = \int \left[\int K(z+y)K(z)dz \right]^2 dy$, which is related to the asymptotic variance of statistic T_m defined in section 3. If we want to minimize the asymptotic variance of T_m, the optimal kernel is just the uniform kernel, as derived by [10].

The factorization of approximate posterior

Our derivation of the new target function (8) is based on the independence among hidden dimensions, which is a fairly strong condition in practice. Exploring the theory of KDEs with dependent latent dimensions would be another compelling direction.

Manifold assumption

Like ordinary VAE, EVAE assumes the Euclidean latent space which is deficient when the data manifold is non-Euclidean, e.g., hyperspherical. With invertible transformations, Normalizing flow [21], Manifold flow ($\mathcal{M}$-flow) [4] and Denosing Normalizing Flow(DNF) [15] are designed to learn complicated structures of manifolds. Combining manifold learning methods with Epanechnikov kernel is also a promising extension of EVAE.

In summary, applying the idea of kernel density estimations in VAE opens a new way of approximating the posterior and enriches the structure of latent space in VAE. Many variants of VAE are based on Gaussian posterior because of it's convenient properties. While the limited expressiveness of Gaussian deteriorates the performance of VAE in many image regeneration tasks. This paper illustrates the potential of replacing the Guassian posterior with Epanechnikov kernel in building generative models. Broader impacts are discussed in Appendix F.

References

[1] Shun-ichi Amari. *Information Geometry and Its Applications*. Springer Publishing Company, Incorporated, 1st edition, 2016.

[2] Jan Jetze Beitler, Ivan Sosnovik, and Arnold Smeulders. PIE: Pseudo-invertible encoder, 2019.

[3] P. J. Bickel and M. Rosenblatt. On Some Global Measures of the Deviations of Density Function Estimates. *The Annals of Statistics*, 1(6):1071 – 1095, 1973.

[4] Johann Brehmer and Kyle Cranmer. Flows for simultaneous manifold learning and density estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[5] Yuri Burda, Roger Baker Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *CoRR*, abs/1509.00519, 2015.

[6] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *ArXiv*, abs/1804.00891, 2018.

[7] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[8] Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, M. J. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *ArXiv*, abs/1611.02648, 2016.

[9] V. A. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158, 1969.

[10] B. K. Ghosh and Wei-Min Huang. The Power and Optimal Kernel of the Bickel-Rosenblatt Test for Goodness of Fit. *The Annals of Statistics*, 19(2):999 – 1009, 1991.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014.

[12] Abul Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, and Liming Chen. von mises-fisher mixture model-based deep learning: Application to face verification. *ArXiv*, abs/1706.04264, 2017.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc.

[14] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

[15] Christian Horvat and Jean-Pascal Pfister. Denoising normalizing flow. In *Neural Information Processing Systems*, 2021.

[16] Dimitrios Kalatzis, David Eklund, Georgios Arvanitidis, and Soren Hauberg. Variational autoencoders with Riemannian brownian motion priors. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5053–5066. PMLR, 13–18 Jul 2020.

[17] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[19] Abhishek Kumar and Ben Poole. On implicit regularization in β-VAEs. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5480–5490. PMLR, 13–18 Jul 2020.

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738. IEEE Computer Society, 2015.

[21] Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *ArXiv*, abs/1509.08731, 2015.

[22] Eric Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. In *International Conference on Learning Representations*, 2017.

[23] Emanuel Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065 – 1076, 1962.

[24] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *ArXiv*, abs/1505.05770, 2015.

[25] Murray Rosenblatt. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27(3):832 – 837, 1956.

[26] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[27] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein autoencoders. In *International Conference on Learning Representations*, 2018.

[28] Jakub Tomczak and Max Welling. Vae with a vampprior. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1214–1223. PMLR, 09–11 Apr 2018.

[29] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

Appendix

A Proofs

A.1 Proof of Lemma 2.1

Recall the Taylor expansion of function $f(x) = x\log x$ around 1 is

$$
\begin{aligned}
x\log x &= f(1) + f'(1)(x-1) + f''(1)\frac{(x-1)^2}{2} + f'''(1)\frac{(x-1)^3}{3!} + \dots \\
&= 0 + x - 1 + \frac{(x-1)^2}{2} - \frac{(x-1)^3}{3!} + \dots
\end{aligned}
\tag{11}
$$

Now for KL-divergence, we have

$$
\begin{aligned}
KL(q||p) &= \int q(z)\log\frac{q(z)}{p(z)}dz = \int \frac{q(z)}{p(z)}\log\frac{q(z)}{p(z)}p(z)dz \\
&= \int \left(\frac{q(z)}{p(z)} - 1 + \frac{1}{2}\left(\frac{q(z)}{p(z)} - 1\right)^2 - \frac{1}{3!}\left(\frac{q(z)}{p(z)} - 1\right)^3 + \dots\right)p(z) \quad \text{Taylor expansion} \\
&= \int \left(\frac{1}{2}\frac{(q(z) - p(z))^2}{p(z)} - \frac{1}{3!}\frac{(q(z) - p(z))^3}{p^2(z)} + \frac{1}{4!}\frac{(q(z) - p(z))^4}{p^3(z)} + \dots\right)dz \\
&= \int \frac{(q(z) - p(z))^2}{p(z)}\left(\frac{1}{2} - \frac{1}{3!}\frac{(q(z) - p(z))}{p(z)} + \frac{1}{4!}\frac{(q(z) - p(z))^2}{p^2(z)} + \dots\right)dz \\
&\leq \tilde{C}\int \frac{(q(z) - p(z))^2}{p(z)}dz
\end{aligned}
\tag{12}
$$

where $\tilde{C}$ is an upper bound for the sum of alternate series in the last equality, which is finite under the assumptions of deviation between $q(z)$ and $p(z)$:

$$
\max_{z\in\mathcal{D}_1\cap\mathcal{D}_2} |(q(z) - p(z))/p(z)| \leq C, \quad 0 < C \leq 1
$$

A.2 Proof of Lemma 3.2

For simplicity, we first consider the following constraints

$$
K \geq 0, \quad \int K(x)dx = 1, \quad \int xK(x)dx = 0, \int x^2 K(x)dx = \frac{1}{5}
$$

By the method of undermined multipliers, it's equivalent to minimize the following target functional without constraints:

$$
\int K^2(x) + aK(x) + cz^2 K(x)dz
$$

with simplified constraints above and a, c are undermined real coefficients. We ignore the term $xK(x)$ as it does not contribute to the unconstrained target function now.

For fixed x, denote $y(K) = K^2 + aK + cx^2 K, (K \geq 0)$. Note that the quadratic function $y(K)$ achieves minimum when $K = -\frac{c}{2}x^2 - \frac{a}{2}$.

It follows that $y(K)$ is minimized subject to $K \geq 0$ by

$$
K(x) = \begin{cases} -\frac{c}{2}x^2 - \frac{a}{2} & -\frac{c}{2}x^2 - \frac{a}{2} \geq 0 \\ 0 & -\frac{c}{2}x^2 - \frac{a}{2} < 0 \end{cases}
$$

We can rewrite it as

$$K(x) = \begin{cases} A(B^2 - x^2) & |x| \leq B \\ 0 & \text{otherwise} \end{cases}$$

for some number A, B. By simplified assumptions $\int x K(x) dx = 0$, $\int x^2 K(x) dx = \frac{1}{5}$, we can find that $A = \frac{3}{4}, B = 1$.

Under general moment conditions in Lemma 3.2, optimal K^* can be written as

$$K^*(x) = \begin{cases} \frac{3}{4r} \left(1 - \left(\frac{x-\mu}{r} \right)^2 \right) & |x - \mu| \leq r \\ 0 & \text{otherwise} \end{cases}$$

by location-scale formula. In literature [9], this kernel is called Epanechnikov kernel. We put $1/5$ in front of the constraint of second moment in order to make the resulting support interval cleaner, which won't change the optimal kernel. The corresponding optimal value of $I(K)$ is $I(K^*) = \frac{3}{5r}$.

Plots of Standard Epanechnikov kernel and Guassian kernel

Figure S1: Red curve: Standard Epanechnikov kernel. Green curve: Standard Gaussian kernel.

A.3 Theoretical support for Algorithm 2

Given $U_1, U_2, U_3 \overset{i.i.d.}{\sim} \text{Unif}[-1, 1]$, we only need to show the density of $\text{median}(U_1, U_2, U_3)$ is standard Epanechnikov kernel.

Denote $Y = \text{Median}(U_1, U_2, U_3)$, we have

$$\begin{aligned} P(Y \leq t) &= P(\text{Median}(U_1, U_2, U_3) \leq t) \\ &= P(\text{Exactly two of } U_1, U_2, U_3 \text{ are less than } t) + P(\text{All of } U_1, U_2, U_3 \text{ are less than } t) \\ &= \binom{3}{2} \left(\frac{1+t}{2} \right)^2 \left(\frac{1-t}{2} \right) + \binom{3}{3} \left(\frac{1+t}{2} \right)^3 \\ &= \frac{1}{2} + \frac{3}{4}t - \frac{t^3}{4} \end{aligned}$$

Then the density $f_Y(t)$ of Y is

$$f_Y(t) = \frac{3}{4} - \frac{3}{4}t^2 = \frac{3}{4}(1 - t^2)$$

which is essentially the standard Epanechnikov kernel.

B Model architecture

B.1 MNIST and Fashion-MNIST

We used fully convolutional architectures with 4×4 convolutional filters for both encoder and decoder in EVAE and VAE, as described following. All convolutions in the encoder and decoder employed SAME padding.

We resized images in MNIST and Fashion-MNIST from 28×28 to 32×32 at beginning. In the last conv layer, the sigmoid activation function was used to restrict the range of output as we assumed Bernoulli type model of $p_\theta(\mathbf{x}|\mathbf{z})$ and the binary cross entropy loss employed used (reduction to sum). Dimensions d_z for latent space : $\{8, 16, 32, 64\}$

Encoder q_ϕ:

$$
\begin{aligned}
x \in \mathbb{R}^{32 \times 32} &\to 32 \text{ Conv, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to 64 \text{ Conv, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to 128 \text{ Conv, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to 256 \text{ Conv, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to \text{Fully connected } (1 * 1 * 256 \times d_z) \text{ for each parameters}
\end{aligned}
$$

Decoder p_θ:

$$
\begin{aligned}
z \in \mathbb{R}^{d_z \times d_z} &\to \text{Fully connected } (d_z \times 1 * 1 * 256) \\
&\to 126 \text{ ConvTran, Stride 1} \to \text{BatchNorm} \to \text{RelU} \\
&\to 64 \text{ ConvTran, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to 32 \text{ ConvTran, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to 1 \text{ ConvTran, Stride 2} \to \text{Sigmoid}
\end{aligned}
$$

B.2 CIFAR-10

Again, we used fully convolutional architectures with 4×4 convolutional filters for both encoder and decoder in EVAE and VAE for CIFAR-10 model. In encoder, we employed a layer of Adaptive Average pool filter. Other settings are the same with MNIST and Fashion-MNIST.

Encoder q_ϕ:

$$
\begin{aligned}
x \in \mathbb{R}^{32 \times 32} &\to 32 \text{ Conv, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to 64 \text{ Conv, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to 128 \text{ Conv, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to 256 \text{ Conv, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to \text{AdaptiveAvgPool2d} \\
&\to \text{Fully connected } (1 * 1 * 256 \times d_z) \text{ for each parameters}
\end{aligned}
$$

Decoder p_θ:

$$
\begin{aligned}
z \in \mathbb{R}^{d_z \times d_z} &\to \text{Fully connected } (d_z \times 1 * 1 * 256) \\
&\to 128 \text{ ConvTran, Stride 1} \to \text{BatchNorm} \to \text{RelU} \\
&\to 64 \text{ ConvTran, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to 32 \text{ ConvTran, Stride 2} \to \text{BatchNorm} \to \text{RelU} \\
&\to 3 \text{ ConvTran, Stride 2} \to \text{Sigmoid}
\end{aligned}
$$

B.3 CelebA

For CelebA dataset, we used 5×5 convolutional filters for both encoder and decoder in EVAE and VAE. Simiar to CIFAR-10, we employed a layer of Adaptive Average pool filter before the fully connected layer in encoder. We first scaled images with Center Crop to 140×140 and resized them to 64×64.

Encoder q_ϕ:

$$x \in \mathbb{R}^{64 \times 64} \rightarrow \text{64 Conv, Stride 2} \rightarrow \text{BatchNorm} \rightarrow \text{RelU}$$
$$\rightarrow \text{128 Conv, Stride 2} \rightarrow \text{BatchNorm} \rightarrow \text{RelU}$$
$$\rightarrow \text{256 Conv, Stride 2} \rightarrow \text{BatchNorm} \rightarrow \text{RelU}$$
$$\rightarrow \text{512 Conv, Stride 2} \rightarrow \text{BatchNorm} \rightarrow \text{RelU}$$
$$\rightarrow \text{AdaptiveAvgPool2d}$$
$$\rightarrow \text{Fully connected } (1 * 1 * 512 \times d_z) \text{ for each parameters}$$

Decoder p_θ:

$$z \in \mathbb{R}^{d_z \times d_z} \rightarrow \text{Fully connected } (d_z \times 8 * 8 * 512)$$
$$\rightarrow \text{256 ConvTran, Stride 1} \rightarrow \text{BatchNorm} \rightarrow \text{RelU}$$
$$\rightarrow \text{128 ConvTran, Stride 2} \rightarrow \text{BatchNorm} \rightarrow \text{RelU}$$
$$\rightarrow \text{64 ConvTran, Stride 2} \rightarrow \text{BatchNorm} \rightarrow \text{RelU}$$
$$\rightarrow \text{3 ConvTran, Stride 2} \rightarrow \text{Sigmoid}$$

C Datasets and Training details

We list details for each benchmark dataset in following table

Datasets	# Training samples	# Hold-out samples	Original image size
MNIST	60000	10000	28*28
Fashion-MNIST	60000	10000	28*28
CIFAR-10	50000	10000	32*32
CelebA	162770	19867	178*218

Note that for MNIST,Fashion-MNIST and CIFAR-10, we used default splittings of training sets and testing sets provided in Pytorch (torchvision.datasets). For CelebA, we used default validation set as hold-out samples.

As to the training details, we used same training parameters for all algorithms and datasets, as described in following table

Latent space dimensions d_z	8,16,32,64
Optimizer	Adam with learning rate 1e-4
Batch size	100
Epochs	50

Calculation of Sharpness

We follow the way in [27] in calculating the sharpness of an image. For each generated image, we first transformed it into grayscale and convolved it with the Laplace filter $\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$, computed the variance of the resulting activations and took the average of all variances. The resulting number is denoted as sharpness (larger is better). The blurrier image will have less edges. As a result, the variance of activations will be small as most activations will be close to zero. Note that we averaged the sharpness of all reconstructed images from hold-out samples for each dataset.

Binomial test for two models

If EVAE and VAE have similar performance in FID score, the probability that EVAE has lower FID score should be 0.5 in each independent experiment. (Same hypothesis for sharpness). However, according to Table 1 and 2, EVAE wins all experiments for FID score in all datasets. The p-value of winning 15 experiments under null hypothesis is

$$P(X \geq 15) = \binom{16}{16}(0.5)^{16} + \binom{16}{15}(0.5)^{16} \approx 2.6 \times 10^{-4} < 0.05.$$

P-value is smaller than 0.05 significance level thus EVAE significantly outperforms VAE in FID. Similar calculation can be applied to sharpness, whose p-value of winning 12 experiments is $P(X \geq 12) = \sum_{i=12}^{16} \binom{16}{i} 0.5^{16} \approx 0.0384 < 0.05$ and we achieved the same conclusion for the significance of EVAE's superiority in sharpness.

D   Sampled reconstructed images

(a) Real images   (b) VAE reconstructed images   (c) EVAE reconstructed images

Figure S2: (a) Sampled real images from hold-out samples in MNIST (b) Reconstructed images by VAE. (c) Reconstructed images by EVAE. Dimension $d_z = 64$ for both models. See section B for Model architectures.

(a) Real images   (b) VAE reconstructed images   (c) EVAE reconstructed images

Figure S3: (a) Sampled real images from hold-out samples in Fashion-MNIST (b) Reconstructed images by VAE. (c) Reconstructed images by EVAE. Dimension $d_z = 64$ for both models.

| | | |
| (a) Real images | (b) VAE reconstructed images | (c) EVAE reconstructed images |

Figure S4: (a) Sampled real images from hold-out samples in CIFAR-10 (b) Reconstructed images by VAE. (c) Reconstructed images by EVAE. Dimension $d_z = 64$ for both models.

| | | |
| (a) Real images | (b) VAE reconstructed images | (c) EVAE reconstructed images |

Figure S5: (a) Sampled real images from hold-out samples in CelebA (b) Reconstructed images by VAE. (c) Reconstructed images by EVAE. Dimension $d_z = 64$ for both models.

E   Validation curves

| | |
| (a) MNIST | (b) Fashion-MNIST |

Figure S6: Reconstruction validation loss curves as function of Epochs. ($d_z = 64$). Red curve is for EVAE and yellow one represents VAE. Binary cross entropy loss is reduced to sum for each batch (a) MNIST (b) Fashion-MNIST.

(a) CIFAR-10

(b) CelebA

Figure S7: Reconstruction validation loss curves as function of Epochs. ($d_z = 64$). Red curve is for EVAE and yellow one represents VAE. Binary cross entropy loss is reduced to sum for each batch (a) CIFAR-10 (b) CelebA.

F   Broader Impacts

This paper aims to propose a new perspective in designing the posterior in generative models. The theoretical results should not have negative societal impacts. One possible negative impact resulting from EVAE might be the misuse of generative models in producing fake images which may lead to security issues in some face recognition based systems. Few mitigation strategies: (1) gate the release of models for commercial use;(2) add a mechanism for monitoring fake images generated by models such as the discriminator in GAN models. We can also restrict the private datasets used in training the generative model. All benchmark datasets used in this paper are public and well known to the machine learning community.