VAEPP: Variational Auto Encoder with a Pull-back Prior

Abstract

1 Introduction

How to learn deep generative model that are able to capture complex data distribution in high dimension space, *e.g.*image datasets, is one of the major challenges in machine learning. There are many different approaches to train generative model by distinct training objective. Generative Adversarial Networks (GAN) [Goodfellow *et al.*, 2014] are famous models based on adversarial training and flow-based models [Dinh *et al.*, 2016; Kingma and Dhariwal, 2018], PixelCNN [Van den Oord *et al.*, 2016], and variational autoencoders (VAE) [Kingma and Welling, 2014; Rezende *et al.*, 2014] are outstanding based on log-likelihood.

VAE uses the variational inference and re-parameterization trick to optimize the evidence lower bound objective of log-likelihood (ELBO). In the past, numerous researches try to enrich the representation ability of true posterior and variational posterior [Kingma *et al.*, 2016; Tomczak and Welling, 2016], but recently some researches show that a simplistic standard Gaussian prior could lead to poor representation in latent space and it is harmful to the performance of VAE [Tomczak and Welling, 2018]. To enrich the representation ability of prior, several learnable prior are proposed [Tomczak and Welling, 2018; Bauer and Mnih, 2019; Takahashi *et al.*, 2019]. Most of them focus on the aggregated posterior which is the integral of variational posterior and is shown as the optimal prior. **We argue that the aggregated posterior is intractable in practice, and it is advisable to choose another feasible learnable prior, which minimizes Wasserstein distance.**

We introduce Pull-back Prior, to improve the representation ability of prior, through the theory of Wasserstein distance [Arjovsky *et al.*, 2017] and learnable prior. The key idea of inference is to search the analytical optimal prior which minimizes Wasserstein distance between model distribution and empirical distribution by calculus of variations. The intuitive interpretation of Pull-back Prior is easy-understanding: Firstly, a discriminator is trained for assessing the quality of images. Then, this discriminator is pull-back to latent space by true posterior. Finally, we increase the density where pull-back discriminator is good and decrease the density where pull-back discriminator is bad in learnable prior.

We design a simple algorithm to train VAE with Pull-back Prior, called Naive VAEPP. Since training objectives of variational posterior, true posterior and learnable prior are different, the performance of VAEPP is limited and the training process is unstable. To achieve better log-likelihood and more stable training, we propose VAEPP, based on SGVB [Kingma and Welling, 2014] and gradient penalty term, which mixes Wasserstein distance into VAE and extends to a more general VAE framework.

Thanks to the powerful gradient pelnaty term of WGAN-GP [Gulrajani *et al.*, 2017] and WGAN-div [Wu *et al.*, 2018], and the practical implement of Langevin dynamics in MEG [Kumar *et al.*, 2019], we enjoy the stable efficient training and sampling process.

The main contributions of this paper are the following:

- We propose Pull-back Prior, which has powerful presentation ability and rises a novel direction to construct new learnable prior.
- We propose VAEPP framework to use the existing techniques of VAE and WGAN to improve the representation ability, sampling quality and stability of training. It is a general and easy-extend VAE framework and may guide a series of models cross the VAE and GAN.
- In log-likelihood metrics, VAEPP outperforms the models without autoregressive components and is competitive to the autoregressive models in log-likelihood metric on vast common datasets. In FID and IS metrics, it outperforms other VAEs and is competitive to GANs in default setting on vast common datasets.

2 Background

2.1 VAE and learnable prior

Many generative models aim to minimize the KL-divergence between empirical distribution $p^*(x)$ and model distribution $p_\theta(x)$, which lead to the maximization of log-likelihood. VAE [Kingma and Welling, 2014] models the joint distribution $p_\theta(x, z)$ where Z is a latent variable and the marginal distribution is $p_\theta(x) = \int p_\theta(x, z)\mathrm{d}z$. VAE apply variational inference to obtain the evidence lower bound objective

(ELBO):

$$\ln p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\ln p_\theta(x|z) + \ln p_\theta(z) - \ln q_\phi(z|x)] \triangleq \mathcal{L}(x;\theta,\phi) \quad (1)$$

where $q_\phi(z|x)$ is the variational posterior (encoder) and $p_\theta(x|z)$ is the true posterior (decoder). The training objective of VAE is $\mathbb{E}_{p^*(x)}[\mathcal{L}(x;\theta,\phi)]$ and it is optimized by SGVB with re-parameterization trick. In vanilla VAE, prior $p_\theta(z)$ is chosen as the standard Gaussian distribution.

Recently, some researchers realized that the simplistic prior could lead to poor hidden representation and many learnable priors are proposed subsequently to improve the representation ability of prior [Tomczak and Welling, 2018]. The learnable prior is denoted by $p_\lambda(z)$. Most of them focus on the aggregated posterior $q_\phi(z)$, which is shown as the optimal prior for ELBO by following decomposition:

$$\mathcal{L}(\theta,\phi,\lambda) = \mathbb{E}_{p^*(x)}\mathbb{E}_{q_\phi(z|x)}[\ln p_\theta(x|z)] + \mathbb{E}_{p^*(x)}[\mathbb{H}[q_\phi(z|x)]] + \mathbb{E}_{q_\phi(z)}\ln p_\lambda(z) \quad (2)$$

where $p_\lambda(z)$ only appears in the last term and the optimal solution of $p_\lambda(z)$ is $q_\phi(z)$. However, $q_\phi(z)$ is intractable and [Tomczak and Welling, 2018; Takahashi *et al.*, 2019] try to obtain an approximation of it as prior.

2.2 GAN and Wasserstein distance

The key idea of vanilla GAN is to train a generator to generate samples to deceive discriminator, and a discriminator to distinguish the generated samples and real samples. However, vanilla GAN is unstable in training process and WGAN and Wasserstein distance are introduced for tackling this problem. Wasserstein distance is based on transition theory and it vastly extend the theory of GAN. 1-st Wasserstein distance $W^1(\mu,\nu)$ is used for calculating the distance between two measures μ,ν. The dual form of Wasserstein distance is following:

$$W^1(\mu,\nu) = \sup_{Lip(D)\leq 1}\{\mathbb{E}_{\mu(x)}D(x) - \mathbb{E}_{\nu(x)}D(x)\} \quad (3)$$

where $Lip(D) \leq 1$ means D is 1-Lipschitz. WGAN is optimized by minimizing $W^1(p^*, p_\theta)$ which can be expanding to a min-max optimization, whose parameters are D and θ.

WGAN makes progress toward stable training but sometimes fails to converge due to the use of weight clipping to enforce the Lipschitz constrain. WGAN-GP [Gulrajani *et al.*, 2017] pointed out this issue and improved WGAN by gradient penalty technique to implement more stable training and WGAN-div [Wu *et al.*, 2018] proposed alternative method of gradient penalty. These techniques make WGAN framework become more robust and stable.

3 Pull-back Prior

3.1 Intuition

We follow the way of learnable priors and improve the performance of VAE by apply another powerful learnable prior, called by Pull-back Prior. It is the basic idea of Pull-back Prior that the optimal solution of prior, i.e. aggregated prior, is intractable and another feasible prior which is not optimal theoretically, could lead to better performance than an approximation aggregated prior.

The formula of Pull-back Prior is given by:

$$\ln p_\lambda(z) = \ln p_\mathcal{N}(z) - \beta * D(G(z)) - \ln Z \quad (4)$$

where $p_\mathcal{N}(z)$ is a simple prior (*e.g.*standard normal) β is a scalar called pull-back weight, D is a discriminator defined on $\mathcal{X}$ (data space), G is an generator defined by $G(z) = \mathbb{E}_{p_\theta(x|z)}x$ and Z is the partition function $Z = \int_\mathcal{Z} p_\mathcal{N}(z)\exp\{-\beta * D(G(z))\}\mathrm{d}z$ ($\mathcal{Z}$ denotes latent space).

A simplistic explanation of Pull-back Prior is given following: We would like to get a more powerful prior than simple prior $p_\mathcal{N}$. A simple way is to improve the density of z which generates better data and decrease the density of z which generates worse data. D is a discriminator to assess the quality of x. When $D(x)$ is less, x is more similar to real data and of higher quality. We could pull-back the discriminator from data space to latent space, and function $D(G(z))$ represents the quality of the data generated by z. To improve and decrease the density at better z and worse z, we modify $p_\mathcal{N}(z)$ by $\beta * D(G(z))$ and then normalize it by Z, and finally we obtain the Pull-back Prior.

3.2 Inference

Before the starting of inference of Pull-back Prior, we need to review the inference of aggregated posterior. We divide the optimization of $\min_{\theta,\phi,\lambda}\mathcal{L}(\theta,\phi,\lambda)$ into 2 part $\min_{\theta,\phi}\min_\lambda\mathcal{L}(\theta,\phi,\lambda)$. Considering the 2nd optimization $\min_\lambda\mathcal{L}(\theta,\phi,\lambda)$, the optimal solution is $p_\lambda(z) = q_\phi(z)$. The key idea of the inference of Pull-back Prior is to set another objective function $\hat{\mathcal{L}}$ for 2nd optimization. Noticing that the ELBO is derived by KL-divergence between p^* and p_θ, a candidate objective function is another divergence. This operation is called Double Metrics Analysis (DMA).

Choosing another divergence will lead to a new learnable prior rather than q_ϕ. We could make this new learnable prior feasible and efficient, but however, it will never be the theoretical optimal prior, *i.e.*the essence of DMA is to get an acceptable trade-off between theory and practice.

We choose Wasserstein distance for 2nd optimization, because it shows wonderful performance in WGAN and has stable theoretical basis in transition theory. Considering following optimization:

$$\min_\lambda \hat{\mathcal{L}}(\theta,\phi,\lambda) = \min_\lambda W^1(p_\theta, p^*) = \min_\lambda \sup_{Lip(D)\leq 1}\{\mathbb{E}_{p_\lambda(z)}\mathbb{E}_{p_\theta(x|z)}D(x) - \mathbb{E}_{p^*(x)}D(x)\} \quad (5)$$

It is hard to get an analytical solution of λ directly from eq. (5), therefore we add an assumption to simplify it. Since $p_\theta(x|z)$ is usually a distribution with small variance, it is rational to assume $\mathbb{E}_{p_\theta(x|z)}D(x) = D(\mathbb{E}_{p_\theta(x|z)}x) = D(G(z))$. Even though, the optimization $\sup_{Lip(D)\leq 1}$ is still tough because we need find optimal D for each λ. If we restrict p_λ near the $p_\mathcal{N}$, this optimization may be approximated by a fixed D obtained in $W^1(p^\dagger, p^*)$, where

$p^\dagger(x) = \mathbb{E}_{p_\mathcal{N}(z)}\, p_\theta(x|z)$ (distribution generated by $p_\mathcal{N}$). Consequently, the simplified optimization is following:

$$\min_\lambda \{\mathbb{E}_{p_\lambda(z)}\, D(G(z)) - \mathbb{E}_{p^*(x)}\, D(x)\} \tag{6}$$

$$\textbf{s.t. } KL(p_\lambda, p_\mathcal{N}) \le \alpha, \qquad \int_\mathcal{Z} p_\lambda(z)\mathrm{d}z = 1,$$

$$D = \arg \sup_{Lip(D)\le 1} \{\mathbb{E}_{p_\mathcal{N}(z)}\, D(G(z)) - \mathbb{E}_{p^*(x)}\, D(x)\}$$

We could solve the simplified optimization eq. (6) by Lagrange multiplier method introduced by calculus of variation [Gelfand *et al.*, 2000]. The Lagrange function with Lagrange multiplier η, γ is following:

$$F(p_\lambda, \eta, \gamma) = \mathbb{E}_{p_\lambda(z)}\, D(G(z)) - \mathbb{E}_{p^*(x)}\, D(x) + \eta(\int_\mathcal{Z} p_\lambda(z)\mathrm{d}z - 1) + \gamma(KL(p_\lambda, p_\mathcal{N}) - \alpha) \tag{7}$$

We solve eq. (7) by Euler-Lagrange equation:

$$D(G(z)) + \eta - \gamma(\ln p_\lambda(z) + 1 - \ln p_\mathcal{N}(z)) = 0 \tag{8}$$

Therefore, $\ln p_\lambda(z) = \frac{1}{\gamma} D(G(z)) + \ln p_\mathcal{N}(z) + (\frac{\eta}{\gamma} - 1)$ is the optimal solution, which could be organized into eq. (4). From this inference, we could explain the meaning of $\beta = \frac{1}{\gamma}$ and $Z = \frac{\eta}{\gamma} - 1$. β represents how far p_λ is from $p_\mathcal{N}$, since γ is the Lagrange multiplier of constraint $KL(p_\lambda, p_\mathcal{N}) \le \alpha$. Z is the partition function since η is the Lagrange multiplier of constraint $\int_\mathcal{Z} p_\lambda(z)\mathrm{d}z = 1$.

We obtain the basic formula of Pull-back Prior. However, it remains some troubles about how to optimize it and calculate partition function Z in VAE architecture.

4 VAEPP

In section 3, we focus on the 2nd optimization and propose the Pull-back Prior as the analytical solution of it. In this section, we will return to the original objective ELBO, and discuss how to use Pull-back Prior to optimize ELBO.

4.1 Determine β

β in eq. (4) represents how far p_λ is from $p_\mathcal{N}$, but how to decide the value of β? When β is smaller, the difference between p_λ and $p_\mathcal{N}$ is less, i.e. the representation ability of p_λ is severely limited. When β is larger, p_λ is farther from $p_\mathcal{N}$. But noticing that in eq. (6), we simplify the optimization of D by a fixed D obtained in $W^1(p^\dagger, p^*)$, if p_λ is too far from $p_\mathcal{N}$, this approximation will become invalid. Consequently, β should be set to an appropriate value which can't limit the representation ability of p_λ and could make sure the approximation D is valid. It is important to realize that the Pull-back Prior is serving for better ELBO. Whatever the representation ability of p_λ is limited or approximation D is invalid, the ELBO will suffer. Therefore, it is reasonable to search β by the optimization for ELBO (λ contains β and ω, which is the parameters of D):

$$\beta = \arg\min_\beta \mathcal{L}(\theta, \phi, \lambda) = \arg\min_\beta \mathcal{L}(\theta, \phi, \beta, \omega) \tag{8}$$

The optimization process of β depends on $\partial\mathcal{L}/\partial\beta$:

$$\frac{\partial\mathcal{L}}{\partial\beta} = \mathbb{E}_{q_\phi(z)}[-D(G(z))] - \frac{\partial Z}{\partial\beta} = \mathbb{E}_{p_\lambda(z)}[D(G(z))] - \mathbb{E}_{q_\phi(z)}[D(G(z))] \tag{9}$$

The 1st term in eq. (9) is the mean of discriminator on data generated from p_λ. The 2nd term in eq. (9) is the mean of discriminator on reconstructed data which is nearly same as real data when reconstruction is well-trained. Hence, $\partial\mathcal{L}/\partial\beta = 0$ represents that the discriminator can't distinguish the reconstructed data (nearly same as real data) and generated data. It coincides the philosophy of GAN.

4.2 Determine Z

We have known that $Z = \int_\mathcal{Z} p_\mathcal{N}(z) \exp\{-\beta * D(G(z))\}\mathrm{d}z$, denoted by $\int_\mathcal{Z} f_\lambda(z)\mathrm{d}z$. It is natural to determine Z by importance sampling $Z = \mathbb{E}_{p_\mathcal{N}(z)} \exp\{-\beta * D(G(z))\}$ as [Bauer and Mnih, 2019] did. By the theory of importance sampling, the variance of the estimation of Z is $\frac{1}{M} Var_{p_b}[\frac{f_\lambda}{p_b}]$ where M is the number of samples, and p_b is another distribution. Hence, the variance is smallest when $p_b = p_\lambda$ and it is larger when p_λ is farther from p_b. If we choose p_b as $p_\mathcal{N}$, when β is large, the variance will be large and it will influence the optimization and evaluation.

The optimal choice for p_b is p_λ itself but it is hard to sample from p_λ during training. We try to find a distribution which is near to p_λ and easy-sampling. As eq. (9) shows, when β approaches optimal, the discriminator can't distinguish the data generated by $p_\lambda(z)$ and $q_\phi(z)$. eq. (2) also shows that when $p_\lambda(z)$ is optimized for $\mathcal{L}(\theta, \phi, \lambda)$, it approaches to q_ϕ. Consequently, it is reasonable to choose q_ϕ as p_b.

However, as we mentioned before, $q_\phi(z)$ is intractable to compute the exact density. We introduce a bias estimation for $q_\phi(z)$, which will lead to the bias estimation for Z.

$$q_\phi(z) = \mathbb{E}_{p^*(x)}\, q_\phi(z|x) \approx \frac{1}{N}\sum_{i=1}^{N} q_\phi(z|x^{(i)}) \ge \frac{1}{N} q_\phi(z|x^{(j)})$$

where $x^{(j)}$ is one of real data, N is the size of training set. To reduce the error, $q_\phi(z|x^{(j)})$ should be one of the largest in summation. Therefore, we firstly choose $x^{(j)}$, then sample z from $q_\phi(z|x^{(j)})$ (by this way, $q_\phi(z|x^{(j)})$ will be large enough), and finally set $\frac{1}{N} q_\phi(z|x^{(j)})$ as a bias estimation for $q_\phi(z)$. When $p^*(x)$ consists of numerous data (*e.g.*in MNIST the input of model is sampled from real images), $p^*(x)$ is sampled from a finite set $\{e^{(1)}, \ldots, e^{(N)}\}$. We introduce another ELBO which use $q_\phi(z|e)$ instead of $q_\phi(z|x)$. By this way, $q_\phi(z)$ will be easily computed. The ELBO is:

$$\mathbb{E}_{p^*(x)} \ln p_\theta(x) \ge \mathbb{E}_{p^*(e)}\, \mathbb{E}_{p^*(x|e)} \ln \mathbb{E}_{q_\phi(z|e)} \frac{p_\theta(x|z)p_\theta(z)}{q_\phi(z|e)}$$

$$= \mathbb{E}_{p^*(e)}\, \mathbb{E}_{p^*(x|e)}\, \mathbb{E}_{q_\phi(z|e)} \ln \frac{p_\theta(x|z)p_\theta(z)}{q_\phi(z|e)} \tag{10}$$

$$= \mathbb{E}_{p^*(x)} \ln p^*(x) - \mathbb{E}_{p^*(e)}\, \mathbb{E}_{p^*(x|e)}\, KL(q_\phi(z|e), p_\theta(z|x))$$

where $p^*(x|e)$ means the sampling process from e, usually Bernouli distribution. eq. (10) is similar to the original ELBO eq. (1), and the above conclusion about learnable prior holds for eq. (10) by repeating above inference.

Algorithm 1 Naive VAEPP training algorithm

Require: The gradient penalty algorithm R, the batch size b, the number of critic iterations per generator iteration n_c, the parameters for Adam Optimizers, τ.

while $\theta, \phi, \beta, \omega$ have not converged **do**
  for $k = 1, \dots n_c$ **do**
    for $i = 1, \dots, b$ **do**
      Sample real data $x \sim p^*$, $z \sim q_\phi(z|e)$, $\epsilon \sim p_{\mathcal{N}}$
      $Z^{(i)} \leftarrow \frac{1}{2}(\exp\{-\beta * D(G(\epsilon))\} + \frac{f_\lambda(z)}{\hat{q}_\phi(z)})$
      $\mathcal{L}^{(i)} \leftarrow \ln p_\theta(x|z) + \ln f_\lambda(z) - \ln q_\phi(z|e)$
    end for
    $\mathcal{L} \leftarrow \frac{1}{b}\sum_i^b \mathcal{L}^{(i)} - \ln(\frac{1}{b}\sum_i^b Z^{(i)})$
    $\theta, \phi, \beta \leftarrow$ Adam $(\nabla_{\theta,\phi,\beta}\mathcal{L}, \{\theta, \phi, \beta\}, \tau)$
  end for
  for $i = 1, \dots, b$ **do**
    Sample real data $e, x \sim p^*$, latent variable $z \sim p_{\mathcal{N}}$
    $\hat{x} = \mathbb{E}_{p_\theta(x|z)}[x]$, get gradient penalty $\zeta \leftarrow R(x, \hat{x})$
    $L^{(i)} \leftarrow D(\hat{x}) - D(x) + \zeta$
  end for
  $\omega \leftarrow$ Adam $(\nabla_\omega \frac{1}{b}\sum_i^b L^{(i)}, \omega, \tau)$
end while

Since $q_\phi(z|e)$ is known, above bias estimation of $q_\phi(z)$ is feasible by $\frac{1}{N}q_\phi(z|e^{(j)})$. The estimation using $q_\phi(z|x)$ is a special case that $p^*(x|e) = \delta(x - e)$, therefore, we will use $\frac{1}{N}q_\phi(z|e^{(j)})$ on all dataset. $\hat{q}_\phi(z)$ denotes the bias estimation of $q_\phi(z)$. Then, a bias estimation $\hat{Z}$ is given by:

$$Z = \mathbb{E}_{q_\phi(z)} \frac{f_\lambda(z)}{q_\phi(z)} \leq \mathbb{E}_{q_\phi(z)} \frac{f_\lambda(z)}{\hat{q}_\phi(z)} = \hat{Z} \tag{10}$$

Because β is optimized from small to large during training, we use both estimations for Z in training. After training, β is large and p_λ approach to q_ϕ by eq. (9), therefore we use eq. (10) for computing the final value of Z. The last thing we need to check is that the bias of estimation will not improve the log-likelihood in evaluation:

$$p_\theta(x) = \int_{\mathcal{Z}} \frac{1}{Z} f_\lambda(z) p_\theta(x|z) \geq \int_{\mathcal{Z}} \frac{1}{\hat{Z}} f_\lambda(z) p_\theta(x|z) = \hat{p}_\theta(x)$$

which means $\hat{p}_\theta(x)$ is a lower bound of model density $p_\theta(x)$.

Thanks to the stable and efficient gradient penalty regularizer term provided by WGAN-GP and WGAN-div, we enjoy stable and efficient training process. Training algorithm for VAEPP based on WGAN-GP is provided as algorithm 1.

4.3 Improvement of VAEPP

We have introduced the architecture and training algorithm for VAEPP, whose performance is better than vanilla VAE and some variants of VAE with learnable prior. However, we notice that in the training process, the optimization of ω may influence the optimization of θ, ϕ, β, *e.g.*the optimization for ω significantly worsen the loss function which have been optimized well. The reason is that the optimization for ω is independent to the optimization of θ, ϕ, β in algorithm 1. This independence is from the philosophy of GAN but it may suffer the performance of VAEPP (log-likelihood). Hence,

Algorithm 2 VAEPP training algorithm

Require: The gradient penalty algorithm R, the batch size b, the parameters for Adam Optimizers, τ.

while $\theta, \phi, \beta, \omega$ have not converged **do**
  for $i = 1, \dots, b$ **do**
    Sample real data $e, x \sim p^*$, $z \sim q_\phi(z|e)$, $\epsilon \sim p_{\mathcal{N}}$
    $\hat{x} = \mathbb{E}_{p_\theta(x|\epsilon)}[x]$, get gradient penalty $\zeta \leftarrow R(x, \hat{x})$
    $Z^{(i)} \leftarrow \frac{1}{2}(\exp\{-\beta * D(G(\epsilon))\} + \frac{f_\lambda(z)}{\hat{q}_\phi(z)})$
    $\mathcal{L}^{(i)} \leftarrow \ln p_\theta(x|z) + \ln f_\lambda(z) - \ln q_\phi(z|e) + \beta\zeta$
  end for
  $\mathcal{L} \leftarrow \frac{1}{b}\sum_i^b \mathcal{L}^{(i)} - \ln(\frac{1}{b}\sum_i^b Z^{(i)})$
  $\theta, \phi, \beta, \omega \leftarrow$ Adam $(\nabla_{\theta,\phi,\beta}\mathcal{L}, \{\theta, \phi, \beta, \omega\}, \tau)$
end while

it is necessary to combine this two optimization into one to improve the performance and stability of VAEPP. Our solution is to use SGVB with gradient penalty regularizer to train VAEPP, *i.e.*$\max_{\theta,\phi,\beta} \max_{Lip(D)\leq 1} \mathcal{L}(\theta, \phi, \beta, \omega)$.

What will happen if $\mathcal{L}(\theta, \phi, \beta, \omega)$ is directly trained by SGVB? The behavior of θ, ϕ, β is same as algorithm 1 since the optimization for them is not changed. We only need to show the optimization $\max_{Lip(D)\leq 1} \mathcal{L}(\theta, \phi, \beta, \omega)$. We firstly show an inequality of $\ln Z$:

$$\ln Z = \ln \mathbb{E}_{p_{\mathcal{N}}(z)} e^{-\beta * D(G(z))} \geq \mathbb{E}_{p_{\mathcal{N}}(z)}[-\beta * D(G(z))]$$

Then $\max_{Lip(D)\leq 1} \mathcal{L}(\theta, \phi, \beta, \omega)$ indeed find a suboptimal solution for $W^1(p^*, p_\theta)$ (sign $\simeq$ means that optimizations at left and right are equivalent):

$$\begin{aligned} \max_{Lip(D)\leq 1} \mathcal{L} &\simeq \max_{Lip(D)\leq 1} \{-\mathbb{E}_{q_\phi(z)}\, \beta * D(G(z)) - \ln Z\} \\ &\leq \beta \max_{Lip(D)\leq 1} \{\mathbb{E}_{p_{\mathcal{N}}(z)}\, D(G(z)) - E_{q_\phi(z)} D(G(z))\} \\ &= \beta W^1(p^\dagger, p_r) \approx \beta W^1(p^\dagger, p^*) \end{aligned} \tag{11}$$

where p_r denotes $p_r(x) = \mathbb{E}_{q_\phi(z)}\, p_\theta(x|z)$, consisting of reconstructed data. The last approximation sign is from the fact that $p_r \to p^*$ after a few epoch in the training of VAE.

eq. (11) indicates that it is possible to obtain a suboptimal solution for D by directly optimizing $\mathcal{L}$, and the gradient penalty term should be multiplied by β. By this way, the optimizations for ω and θ, ϕ, β is combined into one, which is provides as algorithm 2.

4.4 Sampling

It is not easy to sample z from $p_\lambda(z)$ since the formula of $p_\lambda(z)$ is complicated. Accept/Reject Sampling (ARS) is also not useful for p_λ because ARS requires that $p_\lambda(z)/p_{\mathcal{N}}(z)$ is bounded by a constant M (It means β is limited to a very small value), such that a sample could be sampled in M times.

Langevin dynamics may be a useful sampling method because it only requires that $\nabla_z \log p_\lambda(z)$ is computable and the initial z_0 has an enough high density [Song and Ermon, 2019]. Moreover, MEG [Kumar *et al.*, 2019] have implemented a Metropolis-Adjusted Langevin Algorithm (MALA) for sampling where the formula of density is similar to p_λ and also contains a discriminator term. But the selection of initial z_0 whose density is high enough is still a problem.

Model	MNIST	CIFAR
With autoregressive		
PixelCNN	81.30	3.14
DRAW	80.97	3.58
IAFVAE	79.88	3.11
PixelVAE++	78.00	2.90
PixelRNN	79.20	3.00
VLAE	79.03	2.95
PixelHVAE with VampPrior	78.45	
Without autoregressive		
Implicit Optimal Priors	83.21	
Discrete VAE	81.01	
LARS	80.30	
VampPrior	79.75	
Naive VAEPP	76.49	3.15
VAEPP	76.37	2.91
VAEPP+Flow	(77.11)	2.84

Table 1: Test log-likelihood on MNIST and Bits/dim on CIFAR-10. Bits/dim means $-\log p_\theta(x|z)/(3072 * \ln(2))$. Most of VAEs based on learnable prior focus on MNIST and don't try CIFAR-10. The data is from [Maaløe *et al.*, 2019]. VAEPP+Flow means VAEPP with a normalization flow on encoder, to enhance the ability of encoder. Additional, we compare VAE based on $q_\phi(z|x)$ and $q_\phi(z|e)$ on MNIST, whose NLL are 81.10 and 83.30 respectively. It validates that using $q_\phi(z|e)$ is not the reason of improvement but the powerful presentation ability of Pull-back Prior. VAEPP achieves the state of art on MNIST, and is competitive to the models with autoregressive component.

Following the philosophy of VAEPP, *i.e.*using the technique of GAN to assist VAE, it is natural to use GAN to model the distribution $q_\phi(z)$, and use samples of GAN as the initial point of MALA, which has high enough density in $p_\lambda(z)$. The sampling of VAEPP consists of 3 part: generate initial z_0 by a GAN, then generate $z \sim p_\lambda(z)$ by Langevin dynamics, and finally generate x by decoder.

5 Experiments

VAEPP is evaluated in vast common datasets including MNIST, Fashion-MNIST [Xiao *et al.*, 2017], Omniglot [Lake *et al.*, 2015], CIFAR-10 [Krizhevsky *et al.*, 2009] and CelebA [Liu *et al.*, 2015] with metrics log-likelihood, FID [Heusel *et al.*, 2017] and IS [Salimans *et al.*, 2016] to show the performance of VAEPP. Moreover, we try to help VAE to solve OoD problem by the additional information of discriminator.

5.1 Log-likelihood

We evaluate and compare the performance of VAEPP after training by algorithm 1 and algorithm 2 on CIFAR10 when the garadient penalty algorithm is selected from 3 strategy: WGAN-GP, WGAN-div-1 (sampling the linear combination of two real or two fake data points), WGAN-div-2 (sampling both real or fake data points) as shown in table 3. Our conclusion is that algorithm 2 outperforms algorithm 1 under all of settings in CIFAR-10 dataset. We then evaluate them on other datasets, MNIST, CelebA. This conclusion also holds for them. This validate our proposition in section 4.3.

Model	Static MNIST	Fashion	Omniglot
Naive VAEPP			
VAEPP			
VAEPP+Flow			

Table 2: Test log-likelihood on Static MNIST, Fashion-MNIST and Omniglot. VAEPP+Flow get better loss in Static MNIST than VAEPP but it suffers from overfitting, which leads to worse test log-likelihood.

GP Strategy	Naive VAEPP	VAEPP
WGAN-GP	3.15	2.95
WGAN-div-1	3.20	2.91
WGAN-div-2	4.47	2.99

Table 3: Comparation between VAEPP and Improved VAEPP when gradient penalty strategy varies on CIFAR-10 with dim $\mathcal{Z} = 1024$. For any gradient penalty strategy in the table, VAEPP outperforms Naive VAEPP, which validates the our intuition of design of algorithm 2. We select WGAN-div-1 as our default gradient penalty strategy since it achieves best performance in VAEPP.

We compare our algorithms with other log-likelihood based model on MNIST, CIFAR-10 and CelebA as shown in table 1, table 2. Because the improvement of auto-regressive components is significant, we separate models by whether use auto-regressive component as [Maaløe *et al.*, 2019] did. Improved VAEPP outperforms most of the models without autoregressive component and is competitive to the models with autoregressive component. The reason of why VAEPP don't use auto-regressive component is that VAEPP is time-consuming in training, evaluation and sampling due to the huge structure (need additional discriminator) and Langevin dynamics. It is not easy to apply auto-regressive component on VAEPP considering auto-regressive component is also time-consuming. On the other hand, we expect that the pure improvement on learnable prior could improve the performance of VAE rather than the careful design on encoder or decoder, since it is clearer and easier to develop in theory. Therefore, how to apply autoregressive component on VAEPP is a valuable and challenging practical work. We leave it as a future work.

To valid that it is better to use $q_\phi(z)$ to evaluate Z than $p_\mathcal{N}(z)$ in section 4.2, we calculate the $KL(q_\phi(z)||p_\lambda(z))$ and $KL(p_\mathcal{N}(z)||p_\lambda(z))$ on CIFAR-10 and MNIST. The former is less than the difference (180.3 on CIFAR-10) between reconstruction term and ELBO [Hoffman and Johnson, 2016], and the latter one can be evaluated directly (1011.30 on CIFAR-10). Consequently, $q_\phi(z)$ is closer to $p_\lambda(z)$ than $p_\mathcal{N}(z)$.

To ensure the variance of estimation $\hat{Z}$ is small enough, the $q_\phi(z|e)$ is selected as truncated normal distribution (drop the sample whose magnitude is more than 2 standard deviation form the mean) instead of normal distribution. In eq. (10), $\hat{q}_\phi(z)$ is the denominator and estimated by $\frac{1}{N} q_\phi(z|e^{(j)})$. If $q_\phi(z|e^{(j)})$ is selected as normal distribution, the probability

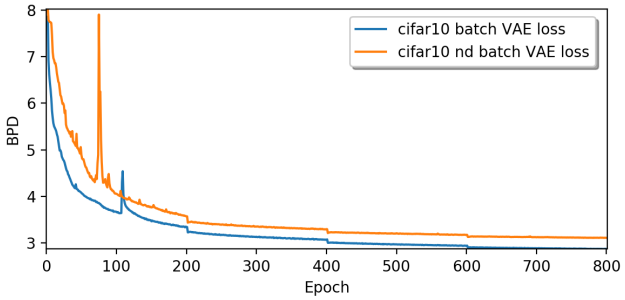

Figure 1: Training process of Naive VAEPP and VAEPP on CIFAR-10. As can been, Naive VAEPP is more unstable and nearly crash at 80 epoch. Although VAEPP have a gap at 100 epoch, but it is acceptable. From global view, the training loss of VAEPP is more smooth than Naive VAEPP and is better than Naive VAEPP over almost all training process, which validates the motivation in section 4.3. The little gap at per 200 epoch is due to the learning rate is half at every 200 epoch.

Model	MNIST	Fashion	CIFAR	CelebA
Best GAN	~ 10	~ 32	~ 70	~ 49
VAE+Flow	54.8.	62.1	81.2	65.7
WAE-MMD	115.0	101.7	80.9	62.9
2-StageVAE	12.6	29.3	72.9	44.4
VAEPP	12.6	29.3	71.0	53.4

Table 4: FID comparasons to GAN-based models and other VAEs. Best GAN indicates the best FID on each dataset across all GAN models when trained using settings suggested by original authors. The data of Best GAN and other VAEs is from [Dai and Wipf, 2019]. As suggested by [Dai and Wipf, 2019], the variance of $p_\theta(x|z)$ of VAEPP is independent on z and $\dim \mathcal{Z}$ is small (128).

of z in tail will be larger when the number of samples become larger, which will lead to huge variance to the estimation $\hat{Z}$. With 10^9 samples, the variance of these two method are 0.809260 (standard) and 0.000967 (truncated) in MNIST.

To valid that VAEPP is more stable and efficient than Naive VAEPP in section 4.3, we draw the training loss of VAEPP and Naive VAEPP on CIFAR-10, shown in fig (TODO).

5.2 Quality of Sampling

The quality of samples of VAE is worse than GAN, and it is indeed a reason that we involve the techniques of GAN to improve the VAE model. We use Wasserstein distance to infer Pull-back Prior and GAN to sample the initial z_0 for Langevin dynamics. These techniques will help VAEPP improve the quality of samples. The samples of VAEPP gets good FID and IS, competitive to GANs and 2-Stage VAE (which is the SOTA of VAE in FID), as shown in table 4.

As common sense, it is hard to achieve best performance in FID, IS and log-likelihood by same setting. We observe this fact that when dimension of latent space is increasing, the trend of FID, IS is different to the trend of log-likelihood, as shown in fig. 2. As diagnosis in [Dai and Wipf, 2019], the variance of $p_\theta(x|z)$ is chosen as a learnable scalar, and the dimension of latent space is selected as a number that little

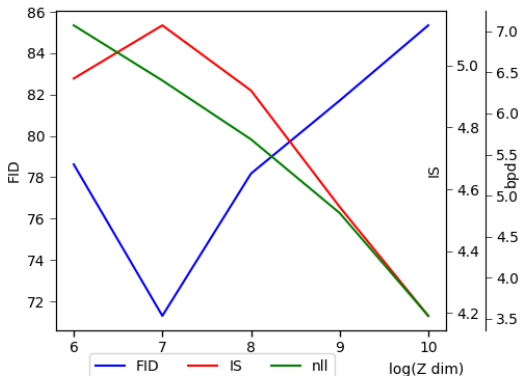

Figure 2: Comparation of VAEPP with learnable scalar variance of $p_\theta(x|z)$, as the dimension of latent space varies on CIFAR-10, with metrics BPD, FID and IS. FID and BPD is better when it is smaller and IS is better when it is larger. When dimension of latent space is greater than 128, the quality of sampling becomes worse and BPD becomes better as $\dim \mathcal{Z}$ increases. It validates the proposition that $\dim \mathcal{Z}$ should be selected as a minimal number of active latent dimensions in [Dai and Wipf, 2019]. Meanwhile, the reconstruction term is optimized more due to the latent space could remains more information and the $KL(q_\phi(z)||p_\lambda(z))$ could stay invariant (learnable prior minimizes it) as $\dim \mathcal{Z}$ is larger. It also shows that FID, IS is not always same as BPD, maybe greatly different.

larger than the dimension of real data manifold, $\dim \mathcal{Z} = 128$, as our experimental result.

To valid the eq. (9), we calculate the $\mathbb{E}_{p_\lambda(z)}[D(G(z))]$ (discriminator on generated samples) and $\mathbb{E}_{q_\phi(z)}[D(G(z))]$ (discriminator on reconstructed samples). They are -12.3467 and -12.2947 respectively on CIFAR-10. The discriminator on real samples are -12.2962, which is nearly same as discriminator on reconstructed samples.

To valid the assumption in section 3.2 holds in actual experiment, we calculate $|\mathbb{E}_{p_\theta(x|z)} D(x) - D(G(z))|$ on $q_\phi(z)$, $p_\mathcal{N}(z)$ and $p_\lambda(z)$, which are (), () and () respectively on CIFAR-10.

6 Conclusion

We propose a novel learnable prior Pull-back Prior for VAE model with solid inference, which rise a novel analysis direction DMA to construct new priors. Based on the inference of Pull-back Prior, we propose the Naive training method for VAE with Pull-back Prior and an improvement of such training method. They are evaluated on vast common dataset, and shows powerful performance in log-likelihood and quality of sampling. We believe that this paper could lead VAE models into a new stage, with clear formula, general framework and powerful performance.

References

[Arjovsky *et al.*, 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[Bauer and Mnih, 2019] Matthias Bauer and Andriy Mnih. Resampled priors for variational autoencoders. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 66–75, 2019.

[Dai and Wipf, 2019] Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.

[Dinh *et al.*, 2016] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[Gelfand *et al.*, 2000] Izrail Moiseevitch Gelfand, Richard A Silverman, et al. *Calculus of variations*. Courier Corporation, 2000.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

[Heusel *et al.*, 2017] 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 *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[Hoffman and Johnson, 2016] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016.

[Kingma and Dhariwal, 2018] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[Kingma *et al.*, 2016] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[Kumar *et al.*, 2019] Rithesh Kumar, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019.

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

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[Maaløe *et al.*, 2019] Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. *arXiv preprint arXiv:1902.02102*, 2019.

[Rezende *et al.*, 2014] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[Salimans *et al.*, 2016] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[Song and Ermon, 2019] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.

[Takahashi *et al.*, 2019] Hiroshi Takahashi, Tomoharu Iwata, Yuki Yamanaka, Masanori Yamada, and Satoshi Yagi. Variational autoencoder with implicit optimal priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5066–5073, 2019.

[Tomczak and Welling, 2016] Jakub M Tomczak and Max Welling. Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

[Tomczak and Welling, 2018] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223, 2018.

[Van den Oord *et al.*, 2016] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

[Wu *et al.*, 2018] Jiqing Wu, Zhiwu Huang, Janine Thoma, Dinesh Acharya, and Luc Van Gool. Wasserstein divergence for gans. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 653–668, 2018.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.