Shallow VAEs with RealNVP Prior Can Perform as Well as Deep Hierarchical VAEs

Haowen Xu[1,2][0000−0003−2841−5788], Wenxiao Chen[1,2][0000−0001−8852−675X],
Jinlin Lai[1][0000−0003−3610−0970], Zhihan Li[1,2][0000−0002−6290−6575],
Youjian Zhao[1,2][0000−0001−9841−1796], and Dan Pei[1,2][0000−0002−5113−838X] ✉

[1] Department of Computer Science and Technology, Tsinghua University
[2] Beijing National Research Center for Information Science and Technology (BNRist)
{xhw15,chen-wx17,laijl16,lizhihan17}@mails.tsinghua.edu.cn
{zhaoyoujian,peidan}@tsinghua.edu.cn

Abstract. Learn the prior of VAE is a new approach to improve the evidence lower-bound. We show that using learned RealNVP prior and just one latent variable in VAE, we can achieve test NLL comparable to very deep state-of-the-art hierarchical VAE, outperforming many previous works with complex hierarchical VAE architectures. We provide the theoretical optimal decoder for Benoulli $p(\mathbf{x}|\mathbf{z})$. We demonstrate that, with learned RealNVP prior, β-VAE can have better rate-distortion curve than using fixed Gaussian prior.

1 Introduction

Variational auto-encoder (VAE) [12] is a powerful deep generative model, trained by variational inference, which demands the intractable true posterior to be approximated by a learned distribution, thus many different variational posteriors have been proposed [12, 16, 11].Alongside, some previous works further improved the variational lower-bound by learning the prior [9, 10, 17, 2].

Despite the achievements of these previous works on posteriors and priors, the state-of-the-art VAE models with continuous latent variables all rely on deep hierarchical latent variables[3], although some of them might have used complicated posteriors/priors as components in their architectures. Most latent variables in such deep hierarchical VAEs have no clear semantic meanings, just a technique for reaching good lower-bounds. We thus raise and answer a question: **with the help of learned priors, can shallow VAEs achieve performance comparable or better than deep hierarchical VAEs?** This question is important because a shallow VAE would be much more promising to scale to more complicated datasets than deep hierarchical VAEs. To answer this question, we conduct comprehensive experiments on several datasets with learned RealNVP priors and just one latent variable, which even shows advantage over some deep hierarchical VAEs with powerful posteriors. In summary, our contributions are:

[3] The term "hierarchical latent variables" refers to multiple layers of latent variables, while "one latent variable" refers to just one $\mathbf{z}$ in standard VAEs. "deep" refers to many hierarchical latent variables, while "shallow" refers to few latent variables.

- We conduct comprehensive experiments on four binarized datasets with four different network architectures. Our results show that VAE with RealNVP prior consistently outperforms standard VAE and RealNVP posterior.
- We are the first to show that using learned RealNVP prior with just one latent variable in VAE, it is possible to achieve test negative log-likelihoods (NLLs) comparable to very deep state-of-the-art hierarchical VAE on these four datasets, outperforming many previous works using complex hierarchical VAE equipped with rich priors/posteriors.
- We provide the theoretical optimal decoder for Bernoulli $p(\mathbf{x}|\mathbf{z})$.
- We demonstrate that, with learned RealNVP prior, β-VAE can have better rate-distortion curve [1] than using fixed Gaussian prior.

2 Preliminaries

2.1 Variational Auto-encoder

Variational auto-encoder (VAE) [12] uses a latent variable $\mathbf{z}$ with prior $p_\lambda(\mathbf{z})$, and a conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$, to model the observed variable $\mathbf{x}$. $p_\theta(\mathbf{x}) = \int_{\mathcal{Z}} p_\theta(\mathbf{x}|\mathbf{z})\, p_\lambda(\mathbf{z})\, \mathrm{d}\mathbf{z}$, where $p_\theta(\mathbf{x}|\mathbf{z})$ is derived by a neural network with parameter θ. $\log p_\theta(\mathbf{x})$ is bounded below by evidence lower-bound (ELBO):

$$\log p_\theta(\mathbf{x}) \geq \mathcal{L}(\mathbf{x};\lambda,\theta,\phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\lambda(\mathbf{z})) \qquad (1)$$

where $q_\phi(\mathbf{z}|\mathbf{x})$ is the variational posterior to approximate $p_\theta(\mathbf{z}|\mathbf{x})$, derived by a neural network with parameter ϕ. Optimizing $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ *w.r.t.* empirical distribution $p^\star(\mathbf{x})$ can be achieved by maximizing the expected ELBO *w.r.t.* $p^\star(\mathbf{x})$:

$$\mathcal{L}(\lambda,\theta,\phi) = \mathbb{E}_{p^\star(\mathbf{x})}\,\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\lambda(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \qquad (2)$$

A hyper-parameter β can be added to $\mathcal{L}(\lambda,\theta,\phi)$, in order to control the trade-off between reconstruction loss and KL divergence, known as β-VAE [8, 1]:

$$\mathcal{L}_\beta(\lambda,\theta,\phi) = \mathbb{E}_{p^\star(\mathbf{x})}\,\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z}) + \beta\left(\log p_\lambda(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right)\right] \qquad (3)$$

[9] suggested an alternative decomposition of Eq. (2):

$$\mathcal{L}(\lambda,\theta,\phi) = \underbrace{\mathbb{E}_{p^\star(\mathbf{x})}\,\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]}_{①} - \underbrace{D_{\mathrm{KL}}(q_\phi(\mathbf{z})\|p_\lambda(\mathbf{z}))}_{②} - \underbrace{\mathbb{I}_\phi[Z;X]}_{③} \qquad (4)$$

where $\mathbb{I}_\phi[Z;X] = \iint q_\phi(\mathbf{z},\mathbf{x}) \log \frac{q_\phi(\mathbf{z},\mathbf{x})}{q_\phi(\mathbf{z})\,p^\star(\mathbf{x})} \mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{x}$ is the *mutual information.* Since $p_\lambda(\mathbf{z})$ is only in ②, ELBO can be enlarged if $p_\lambda(\mathbf{z})$ is trained to match $q_\phi(\mathbf{z})$.

2.2 RealNVP Prior

As a universal density estimator, RealNVP [6] can be readily adopted to derive a learnable prior $p_\lambda(\mathbf{z})$ from a simple prior $p_\xi(\mathbf{w})$ (*e.g.*, unit Gaussian) as follows:

$$p_\lambda(\mathbf{z}) = p_\xi(\mathbf{w}) \left|\det\left(\frac{\partial f_\lambda(\mathbf{z})}{\partial \mathbf{z}}\right)\right|, \quad \mathbf{z} = f_\lambda^{-1}(\mathbf{w}) \qquad (5)$$

where $\det\left(\partial f_\lambda(\mathbf{z})/\partial \mathbf{z}\right)$ is the Jacobian determinant of $f_\lambda(\mathbf{z}) = (f_K \circ \cdots \circ f_1)(\mathbf{z})$, with each f_k being invertible. [6] introduced the *affine coupling layer* as f_k, while [13] further introduced *actnorm* and *invertible 1x1 convolution.*

3 The Optimal Decoder for Bernoulli $p(\mathrm{x}|\mathrm{z})$

Proposition 1. *Given a finite number of discrete training data, i.e.,* $p^\star(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}-\mathbf{x}^{(i)})$*, if* $p_\theta(\mathbf{x}|\mathbf{z}) = \mathrm{Bernoulli}(\boldsymbol{\mu}_\theta(\mathbf{z}))$*, where the Bernoulli mean* $\boldsymbol{\mu}_\theta(\mathbf{z})$ *is produced by the decoder, and* $0 < \mu_\theta^k(\mathbf{z}) < 1$ *for each of its k-th dimensional output, then the optimal decoder* $\boldsymbol{\mu}_\theta(\mathbf{z})$ *is:*

$$\boldsymbol{\mu}_\theta(\mathbf{z}) = \sum_i w_i(\mathbf{z})\,\mathbf{x}^{(i)}, \quad \textit{where } w_i(\mathbf{z}) = \frac{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}{\sum_j q_\phi(\mathbf{z}|\mathbf{x}^{(j)})} \ \textit{ and } \sum_i w_i(\mathbf{z}) = 1 \quad (6)$$

Proof. See [18] due to page limitations.

4 Experiments

4.1 Setup

Datasets We use statically and dynamically binarized MNIST (denoted as *StaticMNIST* and *MNIST* in our paper), FashionMNIST and Omniglot.

Models We perform systematically controlled experiments, using the following models: (1) **DenseVAE**, with dense layers; (2) **ConvVAE**, with convolutional layers; (3) **ResnetVAE**, with ResNet layers; and (4) **PixelVAE** [7], with several PixelCNN layers on top of the ResnetVAE decoder. For RealNVP [6], we use K blocks of invertible mappings (K is called *flow depth* hereafter), while each block contains an *invertible dense*, a dense *coupling layer*, and an *actnorm* [13]. The dimensionality of $\mathbf{z}$ are 40 for StaticMNIST and MNIST, while 64 for FashionMNIST and Omniglot.

Training and evaluation Unless specified, all experiments are repeated for 3 times to report metric means. We perform early-stopping using negative log-likelihood (NLL), to prevent over-fitting on StaticMNIST and on all datasets with PixelVAE. We use 1,000 samples to compute various metrics on test set.

4.2 Quantitative Results

In Tables 1 and 2, we compare ResnetVAE and PixelVAE with RealNVP prior to other approaches on StaticMNIST and MNIST. Due to page limitations, results on Omniglot and FashionMNIST are omitted, but they have a similar trend. All models except ours and that of [10] used at least 2 latent variables. Notice that, although [10] also adopted RealNVP prior, we have better test NLLs than their work, as well as solid analysis on our experimental results.

Our ResnetVAE with RealNVP prior is second only to BIVA among all models without PixelCNN decoder, and ranks the first among all models with PixelCNN

Table 1. Test NLL on StaticMNIST. "†" indicates a hierarchical model with 2 latent variables, while "‡" indicates at least 3 latent variables. $K = 50$ in our models.

Model	NLL
Models without PixelCNN decoder	
ConvHVAE + Lars prior† [2]	81.70
ConvHVAE + VampPrior† [17]	81.09
ResConv + RealNVP prior [10]	81.44
VAE + IAF‡ [11]	79.88
BIVA‡ [14]	**78.59**
Our ConvVAE + RNVP $p(z)$	80.09
Our ResnetVAE + RNVP $p(z)$	79.84
Models with PixelCNN decoder	
VLAE‡ [4]	79.03
PixelHVAE + VampPrior† [17]	79.78
Our PixelVAE + RNVP $p(z)$	**79.01**

Table 2. Test NLL on MNIST. "†" and "‡" has the same meaning as Table 1.

Model	NLL
Models without PixelCNN decoder	
ConvHVAE + Lars prior† [2]	80.30
ConvHVAE + VampPrior† [17]	79.75
VAE + IAF‡ [11]	79.10
BIVA‡ [14]	**78.41**
Our ConvVAE + RNVP $p(z)$	78.61
Our ResnetVAE + RNVP $p(z)$	78.49
Models with PixelCNN decoder	
VLAE‡ [4]	78.53
PixelVAE† [7]	79.02
PixelHVAE + VampPrior† [17]	78.45
Our PixelVAE + RNVP $p(z)$	**78.12**

decoder. On MNIST, the NLL of our model is very close to BIVA, while the latter used 6 latent variables and very complicated architecture. Meanwhile, our ConvVAE with RealNVP prior has lower test NLL than ConvHVAE with *Lars prior* and *VampPrior*. Since the architecture of ConvVAE is simpler than ConvHVAE (which has 2 latent variables), it is likely that our improvement comes from the RealNVP prior rather than the different architecture.

Tables 1 and 2 show that using RealNVP prior with just one latent variable, it is possible to achieve NLLs comparable to very deep state-of-the-art VAE (BIVA), ourperforming many previous works (including works on priors, and works of complicated hierarchical VAE equipped with rich posteriors like VAE + IAF). **This discovery shows that shallow VAEs with learned prior and a small number of latent variables is a promising direction.**

4.3 Qualitative Results

Figure 1 samples images from ResnetVAE with/without RealNVP prior. Compared to standard ResnetVAE, ResnetVAE with RealNVP prior produces fewer digits that are hard to interpret. The last column of each 6x6 grid shows the training set images, most similar to the second-to-last column in pixel-wise L2 distance. There are differences between the last two columns, indicating our model is not just memorizing the training data.

4.4 Ablation Study

RealNVP prior leads to substantially lower NLLs than standard VAE and RealNVP posterior Table 3 shows the NLLs of DenseVAE, ResnetVAE

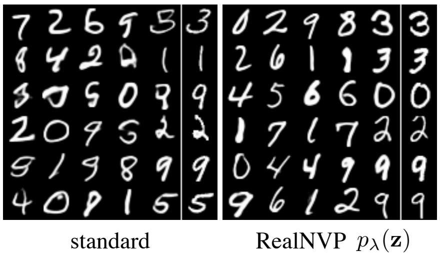

Fig. 1. Sample means from $p_\lambda(\mathbf{z})$ of ResnetVAE with: (left) Gaussian prior; (right) RealNVP prior. The last column of each 6x6 grid shows the training set images, most similar to the second-to-last column in pixel-wise L2 distance.

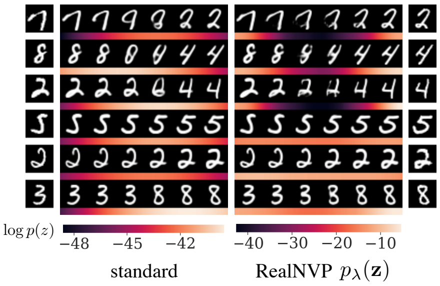

Fig. 2. Interpolations of $\mathbf{z}$ from ResnetVAE, between the centers of $q_\phi(\mathbf{z}|\mathbf{x})$ of two training points, and heatmaps of $\log p_\lambda(\mathbf{z})$. The left- and right-most columns are the training points.

Table 3. Average test NLL (lower is better) of different models, with Gaussian prior & Gaussian posterior ("normal"), Gaussian prior & RealNVP posterior ("RNVP $q(z|x)$"), and RealNVP prior & Gaussian posterior ("RNVP $p(z)$"). $K = 20$.

	DenseVAE			**ResnetVAE**			**PixelVAE**		
Datasets	normal	RNVP $q(z\|x)$	RNVP $p(z)$	normal	RNVP $q(z\|x)$	RNVP $p(z)$	normal	RNVP $q(z\|x)$	RNVP $p(z)$
StaticMNIST	88.84	86.07	**84.87**	82.95	80.97	**79.99**	79.47	79.09	**78.92**
MNIST	84.48	82.53	**80.43**	81.07	79.53	**78.58**	78.64	78.41	**78.15**
FashionMNIST	228.60	227.79	**226.11**	226.17	225.02	**224.09**	224.22	223.81	**223.40**
Omniglot	106.42	102.97	**102.19**	96.99	94.30	**93.61**	89.83	89.69	**89.61**

and PixelVAE with $K = 20$. We see that RealNVP prior consistently outperforms standard VAE and RealNVP posterior in test NLL, with as large improvement as about 2 nats (compared to standard ResnetVAE) or 1 nat (compared to ResnetVAE with RealNVP posterior) on ResnetVAE, and even larger improvement on DenseVAE. The improvement is not so significant on PixelVAE, likely because less information is encoded in the latent variable of PixelVAE [7].

Using RealNVP prior only has better NLL than using both RealNVP prior and posterior, or using RealNVP posterior only, with the same total number of RealNVP layers, as shown in Table 4.

Active units Table 5 counts the *active units* [3] of different ResnetVAEs, which quantifies the number of latent dimensions used for encoding information from input data. We can see that, both RealNVP prior and posterior can make all units of a ResnetVAE to be active (which is in sharp contrast to standard VAE). This in conjunction with Tables 3 and 4 indicates that, the good regularization effect, "a learned RealNVP prior can lead to more active units than a fixed prior" [17, 2], is not the main cause of the huge improvement in NLLs, especially for the improvement of RealNVP prior over RealNVP posterior.

Table 4. Test NLL of ResnetVAE on MNIST, with RealNVP posterior ("$q(z|x)$"), RealNVP prior ("$p(z)$"), and RealNVP prior & posterior ("both"). Flow depth K is $2K_0$ for the posterior or the prior in "$q(z|x)$" and "$p(z)$", while K_0 for both the posterior and the prior in "both".

	K_0			
ResnetVAE &	1	5	10	20
$q(z\|x)$, $K = 2K_0$	80.29	79.68	79.53	79.49
both, $K = K_0$	79.85	79.01	78.71	78.56
$p(z)$, $K = 2K_0$	**79.58**	**78.75**	**78.58**	**78.51**

Table 5. Average number of *active units* of ResnetVAE, with standard prior & posterior ("normal"), RealNVP posterior ("RNVP $q(z|x)$"), and RealNVP prior ("RNVP $p(z)$").

	ResnetVAE		
Dataset	normal	RNVP $q(z\|x)$	RNVP $p(z)$
StaticMNIST	30	**40**	**40**
MNIST	25.3	**40**	**40**
FashionMNIST	27	**64**	**64**
Omniglot	59.3	**64**	**64**

Table 6. Average test ELBO ("*elbo*"), reconstruction loss ("*recons*"), $\mathbb{E}_{p^\star(\mathbf{x})} D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\lambda(\mathbf{z}))$ ("*kl*"), and $\mathbb{E}_{p^\star(\mathbf{x})} D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x}))$ ("$kl_{z|x}$") of ResnetVAE with different priors.

	standard				**RealNVP** $p(z)$			
Dataset	*elbo*	*recons*	*kl*	$kl_{z\|x}$	*elbo*	*recons*	*kl*	$kl_{z\|x}$
StaticMNIST	-87.61	-60.09	**27.52**	4.67	**-82.85**	**-54.32**	28.54	**2.87**
MNIST	-84.62	-58.70	**25.92**	3.55	**-80.34**	**-53.64**	26.70	**1.76**
FashionMNIST	-228.91	-208.94	**19.96**	2.74	**-225.97**	**-204.66**	21.31	**1.88**
Omniglot	-104.87	-66.98	**37.89**	7.88	**-99.60**	**-61.21**	38.39	**5.99**

4.5 Reconstruction Loss and Posterior Overlapping

Better reconstruction loss In Table 6, *ELBO* and *reconstruction loss* ("*recons*", which is ① in Eq. (4)) of ResnetVAE with RealNVP prior are substantially higher than standard ResnetVAE, just as the trend of test log-likelihood (LL) in Table 3. On the contrary, $\mathbb{E}_{p^\star(\mathbf{x})} D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\lambda(\mathbf{z}))$ ("*kl*", which is ② + ③) happens to be larger. Since ELBO equals to ① − (② + ③), this suggests that in our experiments, the improvement in ELBO (and also NLL) of ResnetVAE with RealNVP prior all comes from the improved reconstruction loss.

Smaller standard deviation of Gaussian posterior with RealNVP prior In Fig. 3, we plot the histograms of per-dimensional stds of $q_\phi(\mathbf{z}|\mathbf{x})$, as well as the distances and *normalized distances* (which is roughly distance/std) between each closest pair of $q_\phi(\mathbf{z}|\mathbf{x})$ (see Appendix A for formulations). The stds of $q_\phi(\mathbf{z}|\mathbf{x})$ with RealNVP prior are substantially smaller, and the *normalized distances* are larger. Larger *normalized distances* indicate less density of $q_\phi(\mathbf{z}|\mathbf{x})$ to be overlapping. We discussed one possible theoretical reason of this phenomenon in [18], on the basis of our Proposition 1.

Appropriate overlapping among $q_\phi(\mathbf{z}|\mathbf{x})$ with learned prior To demonstrate that the stds of $q_\phi(\mathbf{z}|\mathbf{x})$ with RealNVP prior are reduced according to the dissimilarity between $\mathbf{x}$ rather than being reduced equally (*i.e.*, $q_\phi(\mathbf{z}|\mathbf{x})$ exhibits

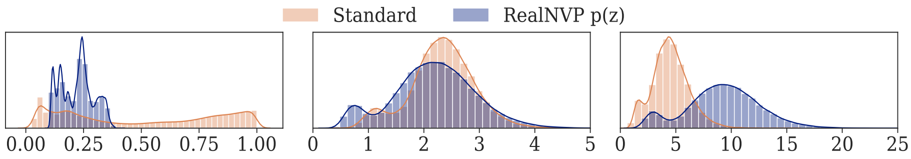

Fig. 3. Histograms of: (left) per-dimensional stds of $q_\phi(\mathbf{z}|\mathbf{x})$; (middle) distances between closest pairs of $q_\phi(\mathbf{z}|\mathbf{x})$; and (right) *normalized distances*. See Appendix A.

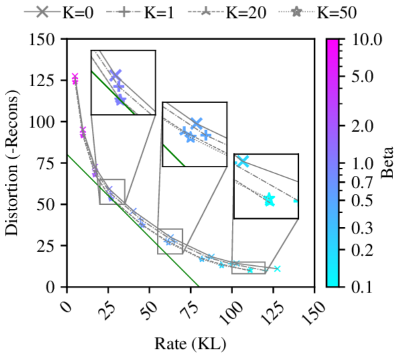

Fig. 4. Rate ($D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}))$ and distortion (*i.e.*, the negate of *reconstruction loss*) of β-ResnetVAE trained with different β and prior flow depth K.

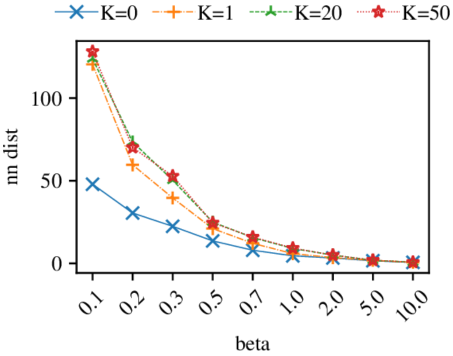

Fig. 5. Average *normalized distance* of β-ResnetVAE trained with different β and prior flow depth K.

"appropriate overlapping"), we plot the interpolations of $\mathbf{z}$ between the centers of $q_\phi(\mathbf{z}|\mathbf{x})$ of two training points, and $\log p_\lambda(\mathbf{z})$ of these interpolations in Fig. 2, We visualize $p_\lambda(\mathbf{z})$, because it is trained to match $q_\phi(\mathbf{z})$, and can be computed much more reliable than $q_\phi(\mathbf{z})$; and because the density of $q_\phi(\mathbf{z})$ between $\mathbf{z}$ corresponding to two $\mathbf{x}$ points can be an indicator of how $q_\phi(\mathbf{z}|\mathbf{x})$ overlap between them. The RealNVP $p_\lambda(\mathbf{z})$ scores the interpolations of $\mathbf{z}$ between the centers of $q_\phi(\mathbf{z}|\mathbf{x})$ of two training points, giving low likelihoods to hard-to-interpret interpolations between two dissimilar $\mathbf{x}$ (the first three rows), while giving high likelihoods to good interpolations between two similar $\mathbf{x}$ (the last three rows). In contrast, the unit Gaussian prior assigns high likelihoods to all interpolations, even to hard-to-interpret ones. This suggests that the posterior overlapping is "more appropriate" with RealNVP prior than with unit Gaussian prior.

Learned prior influences the trade-off between reconstruction loss and KL divergence We plot the rate-distortion curve (RD curve) [1] of β-ResnetVAE trained with different β and flow depth K in Fig. 4. Rate is $D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}))$, while distortion is negative reconstruction loss. Each connected curve with the same shape of points in Fig. 4 correspond to the models with the same K, but different β. We can see that the curves of $K = 1$ is closer to the boundary formed by the green line and the x & y axes than $K = 0$, while $K = 20$ & 50 are even closer. According to [1], points on the RD curve being closer to the boundary suggests that the corresponding models are closer to the theoretical optimal models on a particular dataset, when traded between reconstruction loss and KL divergence. Given this, we conclude that learned prior can lead to a "better" trade-off from the perspective of RD curve.

We also plotted the average *normalized distance* of β-ResnetVAE trained with different β and flow depth K in Fig. 5. Learned prior can encourage less posterior overlapping than unit Gaussian prior for various β, not only for $\beta = 1$.

5 Related work

Learned priors, as a natural choice for the conditional priors of intermediate variables, have long been unintentionally used in hierarchical VAEs [11, 14]. A few works were proposed to enrich the prior, *e.g.*, Gaussian mixture priors [5], and auto-regressive priors [7, 4], without the awareness of its relationship with the *aggregated posterior*, until [9]. Since then, attempts have been made in matching the prior to *aggregated posterior*, by using RealNVP [10], variational mixture of posteriors [17], and learned accept/reject sampling [2]. However, none of these works recognized the improved reconstruction loss induced by learned prior. Moreover, they did not show that learned prior with just one latent variable can achieve comparable results to those of many deep hierarchical VAEs.

The trade-off between reconstruction loss and KL divergence was discussed in the context of β-VAE [8, 1, 15], however, they did not further discuss the impact of a learned prior on this trade-off. [15] also discussed the posterior overlapping, but only within the β-VAE framework, thus was only able to control the degree of overlapping globally, without considering the local dissimilarity between $\mathbf{x}$.

6 Conclusion

In this paper, using learned RealNVP prior with just one latent variable in VAE, we managed to achieve test NLLs comparable to very deep state-of-the-art hierarchical VAE, outperforming many previous works of complex hierarchical VAEs equipped with rich priors/posteriors. We provide the theoretical optimal decoder for Benoulli $p(\mathbf{x}|\mathbf{z})$. We showed that with learned RealNVP prior, β-VAE can have better rate-distortion curve [1] than with fixed Gaussian prior. We believe this paper is an important step towards shallow VAEs with learned prior and a small number of latent variables, which potentially can be more scalable to large datasets than those deep hierarchical VAEs.

A Formulation of closest pairs of $q_\phi(\mathbf{z}|\mathbf{x})$ and others

$q_\phi(\mathbf{z}|\mathbf{x}^{(j)})$ is the *closest neighbor* of $q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$ if $j = \arg\min_{j \neq i} \|\boldsymbol{\mu}_\phi(\mathbf{x}^{(j)}) - \boldsymbol{\mu}_\phi(\mathbf{x}^{(i)})\|$. Such pairs of $q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$ and $q_\phi(\mathbf{z}|\mathbf{x}^{(j)})$ are called *closest pairs of* $q_\phi(\mathbf{z}|\mathbf{x})$. The *distance* d_{ij} and the *normalized distance* $\widetilde{d_{ij}}$ of a closest pair $q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$ and $q_\phi(\mathbf{z}|\mathbf{x}^{(j)})$ are defined as $\mathbf{d}_{ij} = \boldsymbol{\mu}_\phi(\mathbf{x}^{(j)}) - \boldsymbol{\mu}_\phi(\mathbf{x}^{(i)})$, $d_{ij} = \|\mathbf{d}_{ij}\|$, and $\widetilde{d_{ij}} =$

$\frac{2d_{ij}}{\mathrm{Std}[i;j]+\mathrm{Std}[j;i]}$ Roughly speaking, the *normalized distance* $\widetilde{d_{ij}}$ can be viewed as "distance/std" along the direction of $\mathbf{d}_{ij}$, which indicates the scale of the "hole" between $q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$ and $q_\phi(\mathbf{z}|\mathbf{x}^{(j)})$.

Acknowledgments

This work has been supported by National Key R&D Program of China 2019YFB1802504 and the Beijing National Research Center for Information Science and Technology (BNRist) key projects.

References

1. Alemi, A., et al.: Fixing a Broken ELBO. In: ICML. pp. 159–168 (Jul 2018)
2. Bauer, M., Mnih, A.: Resampled priors for variational autoencoders. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 66–75 (2019)
3. Burda, Y., Grosse, R.B., Salakhutdinov, R.: Importance weighted autoencoders. In: ICLR 2016, Conference Track Proceedings (2016)
4. Chen, X., et al.: Variational lossy autoencoder. In: ICLR (2017)
5. Dilokthanakul, N., et al.: Deep unsupervised clustering with gaussian mixture variational autoencoders. arXiv preprint arXiv:1611.02648 (2016)
6. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real NVP. In: 5th International Conference on Learning Representations, ICLR (2017)
7. Gulrajani, I., et al.: Pixelvae: A latent variable model for natural images. In: ICLR 2017, Conference Track Proceedings (2017)
8. Higgins, I., et al.: Beta-vae: Learning basic visual concepts with a constrained variational framework. In: ICLR 2017. vol. 3 (2017)
9. Hoffman, M.D., Johnson, M.J.: Elbo surgery: Yet another way to carve up the variational evidence lower bound. In: Workshop in Advances in Approximate Bayesian Inference, NIPS (2016)
10. Huang, C.W., et al.: Learnable Explicit Density for Continuous Latent Space and Variational Inference. arXiv:1710.02248 [cs, stat] (Oct 2017)
11. Kingma, D.P., et al.: Improved variational inference with inverse autoregressive flow. In: NIPS 2016. pp. 4743–4751 (2016)
12. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: ICLR (2014)
13. Kingma, D.P., Dhariwal, P.: Glow: Generative Flow with Invertible 1x1 Convolutions. In: NIPS 2018, pp. 10215–10224. (2018)
14. Maaløe, et al.: BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling. arXiv:1902.02102 [cs, stat] (Feb 2019)
15. Mathieu, E., Rainforth, T., Siddharth, N., Teh, Y.W.: Disentangling disentanglement in variational autoencoders. In: ICML 2019, pp. 4402–4412 (2019)
16. Rezende, D., Mohamed, S.: Variational Inference with Normalizing Flows. In: ICML-15. pp. 1530–1538 (2015)
17. Tomczak, J., Welling, M.: VAE with a VampPrior. In: International Conference on Artificial Intelligence and Statistics. pp. 1214–1223 (2018)
18. Xu, H., et al.: On the necessity and effectiveness of learning the prior of variational auto-encoder. `http://arxiv.org/abs/1905.13452`