A Batch Normalized Inference Network Keeps the KL Vanishing Away

Qile Zhu[1], Wei Bi[2], Xiaojiang Liu[2], Xiyao Ma[1], Xiaolin Li[3] and Dapeng Wu[1]
[1]University of Florida, [2]Tencent AI Lab, [3]AI Institute, Tongdun Technology
{valder,maxiy,dpwu}@ufl.edu
{victoriabi,kieranliu}@tencent.com
xiaolin.li@tongdun.net

Abstract

Variational Autoencoder (VAE) is widely used as a generative model to approximate a model's posterior on latent variables by combining the amortized variational inference and deep neural networks. However, when paired with strong autoregressive decoders, VAE often converges to a degenerated local optimum known as "posterior collapse". Previous approaches consider the Kullback–Leibler divergence (KL) individual for each datapoint. We propose to let the KL follow a distribution across the whole dataset, and analyze that it is sufficient to prevent posterior collapse by keeping the expectation of the KL's distribution positive. Then we propose Batch Normalized-VAE (BN-VAE), a simple but effective approach to set a lower bound of the expectation by regularizing the distribution of the approximate posterior's parameters. Without introducing any new model component or modifying the objective, our approach can avoid the posterior collapse effectively and efficiently. We further show that the proposed BN-VAE can be extended to conditional VAE (CVAE). Empirically, our approach surpasses strong autoregressive baselines on language modeling, text classification and dialogue generation, and rivals more complex approaches while keeping almost the same training time as VAE.

1 Introduction

Variational Autoencoder (VAE) (Kingma and Welling, 2014; Rezende et al., 2014)is one of the most popular generative framework to model complex distributions. Different from the Autoencoder (AE), VAE provides a distribution-based latent representation for the data, which encodes the input $\mathbf{x}$ into a probability distribution $\mathbf{z}$ and reconstructs the original input using samples from $\mathbf{z}$. When inference, VAE first samples the latent variable from the prior distribution and then feeds it into the decoder to generate an instance. VAE has been successfully applied in many NLP tasks, including topic modeling (Srivastava and Sutton, 2017; Miao et al., 2016; Zhu et al., 2018), language modeling (Bowman et al., 2016), text generation (Zhao et al., 2017b) and text classification (Xu et al., 2017).

An autoregressive decoder (e.g., a recurrent neural network) is a common choice to model the text data. However, when paired with strong autoregressive decoders such as LSTMs (Hochreiter and Schmidhuber, 1997) and trained under conventional training strategy, VAE suffers from a well-known problem named the *posterior collapse* or the KL *vanishing* problem. The decoder in VAE learns to reconstruct the data independent of the latent variable $\mathbf{z}$, and the KL vanishes to 0.

Many convincing solutions have been proposed to prevent posterior collapse. Among them, fixing the KL as a positive constant is an important direction (Davidson et al., 2018; Guu et al., 2018; van den Oord et al., 2017; Xu and Durrett, 2018; Tomczak and Welling, 2018; Kingma et al., 2016; Razavi et al., 2019). Some change the Gaussian prior with other distributions, e.g., a uniform prior (van den Oord et al., 2017; Zhao et al., 2018) or a von Mises-Fisher (vMf) distribution (Davidson et al., 2018; Guu et al., 2018; Xu and Durrett, 2018). However, these approaches force the same constant KL and lose the flexibility to allow various KLs for different data points (Razavi et al., 2019). Without changing the Gaussian prior, free-bits (Kingma et al., 2016) adds a threshold (free-bits) of the KL term in the ELBO object and stops the optimization of the KL part when its value is smaller than the threshold. Chen et al. (2017) point out that the objective of free-bits is non-smooth and suffers from the optimization challenges. δ-VAE (Razavi et al., 2019) sets the parameters in a specific range

*This work was done when Qile Zhu was an intern at Tencent AI Lab. Wei Bi is the corresponding author.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2636–2649
July 5 - 10, 2020.

to achieve a positive KL value for every latent dimension, which may limit the model performance.

Other work analyzes this problem form a view of optimization (Bowman et al., 2016; Zhao et al., 2017a; Chen et al., 2017; Alemi et al., 2018). Recently, He et al. (2019) observe that the inference network is lagging far behind the decoder during training. They propose to add additional training loops for the inference network only. Li et al. (2019) further propose to initialize the inference network with an encoder pretrained from an AE objective, then trains the VAE with the free-bits. However, these two methods are much slower than the original VAE.

The limitation of the constant KL and the high cost of additional training motivate us to seek an approach that allows flexible modeling for different data points while keeping as fast as the original VAE. In this paper, instead of considering the KL individually for each data point, we let it follow a distribution across the whole dataset. We demonstrate that keeping a positive expectation of the KL's distribution is sufficient to prevent posterior collapse in practice. By regularizing the distribution of the approximate posterior's parameters, a positive lower bound of this expectation could be ensured. Then we propose Batch Normalized-VAE (BN-VAE), a simple yet effective approach to achieving this goal, and discuss the connections between BN-VAE and previous enhanced VAE variants. We further extend BN-VAE to the conditional VAE (CVAE). Last, experimental results demonstrate the effectiveness of our approach on real applications, including language modeling, text classification and dialogue generation. Empirically, our approach surpasses strong autoregressive baselines and is competitive with more sophisticated approaches while keeping extremely higher efficiency. Code and data are available at https://github.com/valdersoul/bn-vae.

2 Background and Related Work

In this section, we first introduce the basic background of VAE, then we discuss the lagging problem (He et al., 2019). At last, we present more related work.

2.1 VAE Background

VAE (Kingma and Welling, 2014; Rezende et al., 2014) aims to learn a generative model $p(\mathbf{x}, \mathbf{z})$ to maximize the marginal likelihood $\log p(\mathbf{x})$ on a dataset. The marginal likelihood cannot be calculated directly due to an intractable integral over the latent variable $\mathbf{z}$. To solve this, VAE introduces a variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ which is parameterized by a complex neural network to approximate the true posterior. Then it turns out to optimize the ELBO of $\log p(\mathbf{x})$:

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

where ϕ represents the inference network and θ denotes the decoder. The above first term is the reconstruction loss, while the second one is the KL between the approximate posterior and the prior. The Gaussian distribution $\mathcal{N} \sim (0, I)$ is a usual choice for the prior, and the KL between the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ and the prior $p(\mathbf{z})$ can be computed as:

$$KL = \frac{1}{2}\sum_{i=1}^{n}(\mu_i^2 + \sigma_i^2 - \log \sigma_i^2 - 1), \tag{2}$$

where μ_i and σ_i is the mean and standard deviation of approximate posterior for the i_{th} latent dimension, respectively. When the decoder is autoregressive, it can recover the data independent of the latent $\mathbf{z}$ (Bowman et al., 2016). The optimization will encourage the approximate posterior to approach the prior which results to the zero value of the KL.

2.2 The Lagging Problem

Recently, He et al. (2019) analyze posterior collapse with the Gaussian prior from a view of training dynamics. The collapse is a local optimum of VAE when $q_\phi(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$ for all inputs. They further define two partial collapse states: *model collapse*, when $p_\theta(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$, and *inference collapse*, when $q_\phi(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$. They observe that the inference collapse always happens far before the model collapse due to the existence of autoregressive decoders. Different from the model posterior, the inference network lacks of guidance and easily collapses to the prior at the initial stage of training, and thus posterior collapse happens. Based on this understanding, they propose to aggressively optimize the inference network. However, this approach cost too much time compared with the original VAE. In our work, we also employ the Gaussian prior and thus suffer from the same lagging problem. Yet, our proposed approach does

not involve additional training efforts, which can effectively avoid the lagging problem (Section 3.3) and keep almost the same training efficiency as the original VAE (Section 5.1). More details can be found in Section 3.3.

2.3 Related Work

To prevent posterior collapse, we have mentioned many work about changing the prior in the introduction. Besides these approaches, some work modifies the original training objective directly. For example, Bowman et al. (2016) introduce an annealing strategy, where they slightly increase the weight of KL from 0 to 1 during the warm-up period. β-VAE (Higgins et al., 2017) treats the KL weight as a hyperparameter to constrain the minimum value of the KL. Alemi et al. (2017), on the other hand, set a fixed KL weight to control the mutual information between $\mathbf{z}$ and $\mathbf{x}$. Tolstikhin et al. (2018) leverage the wasserstein distance to replace the KL. Zhao et al. (2017a) replace the KL with maximum mean discrepancy. Fang et al. (2019) introduce sample-based representations which lead to implicit latent features with an auxiliary network.

Some change the training strategy. Kim et al. (2018) address the amortization gap (Cremer et al., 2018) in VAE and propose Semi-Amortized VAE to compose the inference network with additional mean-field updates. Fu et al. (2019) propose a cyclical annealing schedule, which repeats the process of increasing β multiple times.

There are various other approaches to solve the posterior collapse. For example, some researchers choose to weaken the decoder by replacing the LSTM decoder with convolution neural networks without autoregressive modeling (Semeniuta et al., 2017; Yang et al., 2017). Chen et al. (2017) input a lossy representation of data to the autoregressive decoder and enforce $\mathbf{z}$ to capture the information about the original input. Inheriting this idea, some following work add direct connections between $\mathbf{z}$ and $\mathbf{x}$ (Zhao et al., 2017b; Dieng et al., 2019). Ma et al. (2019) introduce an additional regularization to learn diverse latent representation. δ-VAE (Razavi et al., 2019) and free-bits (Kingma et al., 2016) set a minimum number of KL for each latent dimension to prevent the posterior collapse.

Srivastava and Sutton (2017, 2018) find that using ADAM (Kingma and Ba, 2014) with a high learning rate to train VAE may cause the gradients to diverge early. Their explanation for the diverging behavior lies in the exponential curvature of the gradient from the inference network which produces the variance part of the approximate posterior. Then they apply batch normalization to the variance part to solve this problem. We use the simple SGD without momentum to train our model. Moreover, we apply batch normalization to the mean part of the inference network to keep the expectation of the KL's distribution positive, which is different from their work. We also find that Sønderby et al. (2016) utilize batch normalization in all fully connected layers with nonlinear activation functions to improve the model performance. Different from it, our approach directly applies batch normalization to the parameters of the approximate posterior, which is the output of the inference network.

3 Batch-Normalized VAE

In this section, we first derive the expectation of the KL's distribution and show that it is enough to avoid posterior collapse by keeping the expectation of the KL's distribution positive. Then we propose our regularization method on the parameters of the approximate posterior to ensure a positive lower bound of this expectation. We further discuss the difference between our approach and previous work.

3.1 Expectation of the KL's Distribution

Given an $\boldsymbol{x} \in \mathcal{X}$, the inference network parametrizes a n-dimension diagonal Gaussian distribution with its mean $\mu = f_\mu(\boldsymbol{x})$ and diagonal covariance $\boldsymbol{\Sigma} = diag(f_\Sigma(\boldsymbol{x}))$, where f_μ and f_Σ are two neural networks. In practice, the ELBO is computed through a Monte Carlo estimation from b samples. The KL in Eq. 2 is then computed over b samples from $\mathcal{X}$:

$$
\begin{aligned}
KL &= \frac{1}{2b}\sum_{j=1}^{b}\sum_{i=1}^{n}(\mu_{ij}^2 + \sigma_{ij}^2 - \log\sigma_{ij}^2 - 1) \\
&= \frac{1}{2}\sum_{i=1}^{n}(\frac{\sum_{j=1}^{b}\mu_{ij}^2}{b} + \frac{\sum_{j=1}^{b}\sigma_{ij}^2}{b} \\
&\quad - \frac{\sum_{j=1}^{b}\log\sigma_{ij}^2}{b} - 1). \qquad (3)
\end{aligned}
$$

When b gets larger, the above empirical value will approach the mean of the KL across the whole dataset.

To make use of this observation, we assume that μ_i and $\log\sigma_i^2$ for each latent dimension i follow

a certain distribution with a fixed mean and variance across the dataset respectively. The distribution may vary between different latent dimensions. In this way, the KL turns to a distribution of μ_i's and $\log \sigma_i^2$'s. From Eq. 3, we can see that $\sum_{j=1}^{b} \mu_{ij}^2/b$ is the sample mean of μ_i^2, which converges to $\mathrm{E}[\mu_i^2] = \mathrm{Var}[\mu_i] + \mathrm{E}^2[\mu_i]$. Similarly, $\sum_{j=1}^{b} \sigma_{ij}^2/b$ converges to $\mathrm{E}[\sigma_i^2]$, and $\sum_{j=1}^{b} \log \sigma_{ij}^2/b$ to $\mathrm{E}[\log \sigma_i^2]$. Thus, we can derive the expectation of the KL's distribution as:

$$\begin{aligned}\mathrm{E}[KL] &= \frac{1}{2}\sum_{i=1}^{n}(\mathrm{Var}[\mu_i] + \mathrm{E}^2[\mu_i] \\ &\quad + \mathrm{E}[\sigma_i^2] - \mathrm{E}[\log \sigma_i^2] - 1) \\ &\geq \frac{1}{2}\sum_{i=1}^{n}(\mathrm{Var}[\mu_i] + \mathrm{E}^2[\mu_i]), \end{aligned} \tag{4}$$

where $\mathrm{E}[\sigma_i^2 - \log \sigma_i^2] \geq 1$ since the minimum of $e^x - x$ is 1. If we can guarantee a positive lower bound of $\mathrm{E}[KL]$, we can then effectively prevent the posterior collapse.

Based on Eq. 4, the lower bound is only dependent on the number of latent dimensions n and μ_i's mean and variance. This motivates our idea that with proper regularization on the distributions of μ_i's to ensure a positive lower bound of $\mathrm{E}[KL]$.

3.2 Normalizing Parameters of the Posterior

The remaining key problem is to construct proper distributions of μ_i's that can result in a positive lower bound of $\mathrm{E}[KL]$ in Eq. 4. Here, we propose a simple and efficient approach to accomplish this by applying a fixed batch normalization on the output of the inference network (μ_i). Batch Normalization (BN) (Ioffe and Szegedy, 2015) is a widely used regularization technique in deep learning. It normalizes the output of neurons and makes the optimization landscape significantly smoother (Santurkar et al., 2018). Different from other tasks that apply BN in the hidden layers and seek fast and stable training, here we leverage BN as a tool to transform μ_i into a distribution with a fixed mean and variance. Mathematically, the regularized μ_i is written by:

$$\hat{\mu}_i = \gamma \frac{\mu_i - \mu_{\mathcal{B}i}}{\sigma_{\mathcal{B}i}} + \beta, \tag{5}$$

where μ_i and $\hat{\mu}_i$ are means of the approximate posterior before and after BN. $\mu_{\mathcal{B}i}$ and $\sigma_{\mathcal{B}i}$ denote the mean and standard deviations of μ_i. They are biased estimated within a batch of samples for each dimension indecently. γ and β are the scale and shift parameter. Instead of using a learnable γ in Eq. 5, we use a fixed BN which freezes the scale γ. In this way, the distribution of μ_i has the mean of β and the variance of γ^2. β is a learnable parameter that makes the distribution more flexible.

Now, we derive the lower bound of $\mathrm{E}[KL]$ by using the fixed BN. With the fixed mean β and variance γ^2 for μ_i in hand, we get a new lower bound as below:

$$\begin{aligned}\mathrm{E}[KL] &\geq \frac{1}{2}\sum_{i}^{n}(\mathrm{Var}[\mu_i] + \mathrm{E}^2[\mu_i]) \\ &= \frac{n \cdot (\gamma^2 + \beta^2)}{2}. \end{aligned} \tag{6}$$

To this end, we can easily control the lower bound of $\mathrm{E}[KL]$ by setting γ. Algorithm 1 shows the training process.

Algorithm 1 BN-VAE training.

Initialize $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$.
for $i = 1, 2, \cdots$ Until Convergence **do**
  Sample a mini-batch $\mathbf{x}$.
  μ, $\log \sigma^2 = f_\phi(\mathbf{x})$.
  $\mu' = BN_{\gamma,\beta}(\mu)$.
  Sample $\mathbf{z} \sim \mathcal{N}(\mu', \sigma^2)$ and reconstruct $\mathbf{x}$ from $f_\theta(\mathbf{z})$.
  Compute gradients $\mathbf{g}_{\phi,\theta} \leftarrow \nabla_{\phi,\theta}\mathcal{L}(\mathbf{x}; \phi, \theta)$.
  Update ϕ, θ using $\mathbf{g}_{\phi,\theta}$.
end for

3.3 Connections with Previous Approaches

Constructing a positive KL: Both free-bits (Kingma et al., 2016) and δ-VAE (Razavi et al., 2019) set a threshold on the KL value. Free-bits changes the KL term in the ELBO to a hinge loss term: $\sum_i^n \max(\lambda, KL(q_\phi(z_i|x)||p(z_i)))$. Another version of free-bits is to apply the threshold to the entire sum directly instead of the individual value. Training with the free-bits objective, the model will stop to drive down the KL value when it is already below λ. However, Chen et al. (2017) point out that the objective of free-bits is non-smooth and suffers from the optimization challenges. Our approach does not face the optimization problem since we use the original ELBO objective.

δ-VAE sets a target rate of δ for each latent dimension by constraining the mean and variance of

the approximate posterior:

$$\sigma_q = \sigma_q^l + (\sigma_q^u - \sigma_q^l)\frac{1}{1+e^{-q_\phi(x)}}, \quad (7)$$

$$\mu = 2\delta + 1 + \ln(\sigma_q^2) - \sigma_q^2 + \max(0, \mu_\phi(\mathbf{x})), \quad (8)$$

where $[\sigma^l, \sigma^u]$ are the feasible interval for σ_q by solving $\ln(\sigma_q^2) - \sigma_q^2 + 2\delta + 1 \geq 0$. Although δ-VAE can ensure a minimum value for the KL, it limits the model performance due to that the parameters are constrained in the interval. Our approach only constrains the distributions of μ, which is more flexible than δ-VAE. Experiments further show that our approach surpass both free-bits and δ-VAE.

Reducing inference lag: As we focus on the setting of the conventional Gaussian prior, the lagging problem mentioned in Section 2.2 is crucial. To this point, it is beneficial to analyze an alternate form of the ELBO:

$$\mathcal{L} = \log p_\theta(\mathbf{x}) - KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})). \quad (9)$$

With this view, the only goal of the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ is to match the model posterior $p_\theta(\mathbf{z}|\mathbf{x})$. We examine the performance of our approach to reduce inference lag using the same synthetic experiment in He et al. (2019). Details can be found in Section 1 of the Appendix. The synthetic experiment indicates that our approach with the regularization is beneficial to rebalance the optimization between inference and generation, and finally overcomes posterior collapse. We also prefer a large γ due to that a small γ will push the approximate posterior to the prior. More details on the synthetic experiment can be found in the Appendix.

4 Extension to CVAE

Given an observation $\mathbf{x}$ and its output $\mathbf{y}$, CVAE (Sohn et al., 2015; Zhao et al., 2017b) models the conditional distribution $p(\mathbf{y}|\mathbf{x})$. The variational lower bound of the conditional log-likelihood is:

$$\begin{aligned}\mathcal{L} &= \mathrm{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}[\log p_\kappa(\mathbf{y}|\mathbf{x},\mathbf{z})] \\ &\quad - KL(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x})) \\ &\leq \log p(\mathbf{y}|\mathbf{x}). \end{aligned} \quad (10)$$

Different from VAE, the prior $p_\theta(\mathbf{z}|\mathbf{x})$ in CVAE is not fixed, which is also parametrized by a neural network. It is possible to apply another BN on the mean of the prior with a different γ so that the expectation of the KL becomes a constant. However, this lower bound is uncontrollable due to the density of $\mu_1 + \mu_2$ is the convolution of their densities, which is intractable. [1]

To overcome this issue, we propose to constrain the prior with a fixed distribution. We achieve it by adding another KL between the prior and a known Gaussian distribution $r(\mathbf{z})$, i.e. $KL(p_\theta(\mathbf{z}|\mathbf{x})||r(\mathbf{z}))$. Instead of optimizing the ELBO in Eq. 10, we optimize a lower bound of the ELBO for CVAE:

$$\mathcal{L}' = \mathcal{L} - KL(p_\theta(\mathbf{z}|\mathbf{x})||r(\mathbf{z})) \leq \mathcal{L}. \quad (11)$$

The KL term in the new bound is the sum of $KL(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x}))$ and $KL(p_\theta(z|x)||r(z))$, which can be computed as:

$$\begin{aligned}KL = \frac{1}{2}\sum_{i=1}^{n}(&\frac{\sigma_{qi}^2 + (\mu_{qi} - \mu_{pi})^2}{\sigma_{pi}^2} \\ &+ \sigma_{pi}^2 + \mu_{pi}^2 - log\sigma_{qi}^2 - 1), \end{aligned} \quad (12)$$

where σ_q, μ_q and σ_p, μ_p are the parameters of q_ϕ and p_θ respectively. n denotes the hidden size. The KL term vanishes to 0 when and only when q_ϕ and p_θ collapse to $r(\mathbf{z})$, which is the normal distribution. As we explained in Section 3.2, KL won't be 0 when we apply BN in q_ϕ. We then prove that when q_ϕ collapses to p_θ, the KL term is not the minima (details in Section 2 of the Appendix) so that $KL(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x}))$ won't be 0. In this way, we can avoid the posterior collapse in CVAE. Algorithm 2 shows the training details.

Algorithm 2 BN-CVAE training.

1: Initialize ϕ, θ and κ.
2: **for** $i = 1, 2, \cdots$ Until Convergence **do**
3: Sample a mini-batch $\mathbf{x}, \mathbf{y}$.
4: μ_q, $\log\sigma_q^2 = f_\phi(\mathbf{x},\mathbf{y})$ and μ_p, $\log\sigma_p^2 = f_\theta(\mathbf{x})$.
5: $\mu_q' = BN_{\gamma,\beta}(\mu_q)$.
6: Sample $\mathbf{z} \sim \mathcal{N}(\mu_q', \sigma_q^2)$ and reconstruct $\mathbf{y}$ from $f_\kappa(\mathbf{z},\mathbf{x})$.
7: Compute gradients $\mathbf{g}_{\phi,\theta,\kappa} \leftarrow \nabla_{\phi,\theta,\kappa}\mathcal{L}'$.
8: Update ϕ, θ, κ using $\mathbf{g}_{\phi,\theta,\kappa}$.
9: **end for**

[1] We perform empirical study on this method and find that the neural network can always find a small KL value in this situation.

	Yahoo				Yelp			
Model	**NLL**	**KL**	**MI**	**AU**	**NLL**	**KL**	**MI**	**AU**
Without a pretrained AE encoder								
CNN-VAE	≤332.1	10.0	-	-	≤359.1	7.6	-	-
LSTM-LM	328	-	-	-	351.1	-	-	-
VAE	328.6	0.0	0.0	0.0	357.9	0.0	0.0	0.0
β-VAE (0.4)	328.7	6.3	2.8	8.0	358.2	4.2	2.0	4.2
cyclic *	330.6	2.1	2.0	2.3	359.5	2.0	1.9	4.1
Skip-VAE *	328.5	2.3	1.3	8.1	357.6	1.9	1.0	7.4
SA-VAE	327.2	5.2	2.7	9.8	**355.9**	2.8	1.7	8.4
Agg-VAE	**326.7**	5.7	2.9	15.0	**355.9**	3.8	2.4	11.3
FB (4)	331.0	4.1	3.8	3.0	359.2	4.0	1.9	32.0
FB (5)	330.6	5.7	2.0	3.0	359.8	4.9	1.3	32.0
δ-VAE (0.1) *	330.7	3.2	0.0	0.0	359.8	3.2	0.0	0.0
vMF-VAE (13) *	327.4	2.0	-	32.0	357.5	2.0	-	32.0
BN-VAE (0.6) *	**326.7**	6.2	5.6	32.0	356.5	6.5	5.4	32.0
BN-VAE (0.7) *	327.4	8.8	7.4	32.0	**355.9**	9.1	7.4	32.0
With a pretrained AE encoder								
cyclic *	333.1	25.8	9.1	32.0	361.5	20.5	9.3	32.0
FB (4) *	**326.2**	8.1	6.8	32.0	356.0	7.6	6.6	32.0
δ-VAE (0.15) *	331.0	5.6	1.1	11.2	359.4	5.2	0.5	5.9
vMF-VAE (13) *	328.4	2.0	-	32.0	357.0	2.0	-	32.0
BN-VAE (0.6) *	326.7	6.4	5.8	32.0	**355.5**	6.6	5.9	32.0
BN-VAE (0.7) *	326.5	9.1	7.6	32.0	355.7	9.1	7.5	32.0

Table 1: Results on Yahoo and Yelp datasets. We report mean values across 5 different random runs. * indicates the results are from our experiments, while others are from He et al. (2019); Li et al. (2019). We only show the best performance of every model for each dataset. More results on various parameters can be found in the Appendix.

5 Experiments

5.1 VAE for Language Modeling

Setup: We test our approach on two benchmark datasets: Yelp and Yahoo corpora (Yang et al., 2017). We use a Gaussian prior $\mathcal{N}(0, I)$, and the approximate posterior is a diagonal Gaussian. Following previous work (Burda et al., 2016; He et al., 2019), we report the estimated negative log likelihood (NLL) from 500 importance weighted samples, which can provide a tighter lower bound compared to the ELBO and shares the same information with the perplexity (PPL). Besides the NLL, we also report the KL, the mutual information (MI) I_q (Alemi et al., 2017) and the number of activate units (AU) (Burda et al., 2016) in the latent space. The I_q can be calculated as:

$$I_q = \mathrm{E}_{p_d(\mathbf{x})}[KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] - KL(q_\phi(\mathbf{z})||p(\mathbf{z})), \quad (13)$$

where $p_d(\mathbf{x})$ is the empirical distribution. The aggregated posterior $q_\phi(\mathbf{z}) = \mathrm{E}_{p_d(\mathbf{x})}[q_\phi(\mathbf{z}|\mathbf{x})]$ and $KL(q_\phi(\mathbf{z})||p(\mathbf{z}))$ can be approximated with Monte Carlo estimations. The AU is measured as $A_z = Cov(\mathrm{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}[\mathbf{z}])$. We set the threshold of 0.01, which means if $A_{zi} > 0.01$, the unit i is active.

Configurations: We use a 512-dimension word embedding layer for both datasets. For the encoder and the decoder, a single layer LSTM with 1024 hidden size is used. We use $\mathbf{z}$ to generate the initial state of the encoder following Kim et al. (2018); He et al. (2019); Li et al. (2019). To optimize the objective, we use mini-batch SGD with 32 samples per batch. We use one NVIDIA Tesla v100 for the experiments. For all experiments, we use the linear annealing strategy that increases the KL weight from 0 to 1 in the first 10 epochs if possible.

Compared methods: We compare our model with several strong baselines and methods that hold the previous state-of-the-art performance on text modeling benchmarks.

- Baselines, including neural autoregressive models (the LSTM language model).
- Methods with weakening the decoder: CNN-VAE (Yang et al., 2017).
- Methods with a modified model structure: Skip-VAE (Dieng et al., 2019).
- Methods with a modified training objective:
  - VAE with annealing (Bowman et al., 2016).
  - β-VAE (Higgins et al., 2017).
  - Cyclic annealing (Fu et al., 2019), we use the default cyclic schedule.
- Methods with a lower bound for KL values:
  - Free-bits (FB) (Kingma et al., 2016).
  - δ-VAE (Razavi et al., 2019).
  - vMF-VAE (Xu and Durrett, 2018)
- Methods with a modified training strategy.

Model	Yahoo		Yelp	
	Hours	Ratio	Hours	Ratio
VAE	3.83	1.00	4.50	1.00
SA-VAE	52.99	12.80	59.37	12.64
Agg VAE	11.76	2.84	21.44	4.56
AE+FB	7.70	2.01	9.22	2.05
BN-VAE	3.98	1.04	4.60	1.02

Table 2: Comparison of training time to convergence. We report both the absolute hours and relative speed.

- Semi-amortized VAE (SA-VAE) (Kim et al., 2018).
- VAE with an aggressive training (Agg-VAE) (He et al., 2019).
- FB with a pretrained inference network (AE+FB) (Fu et al., 2019)

Main results: Table 1 shows the results. We further split the results into two different settings, one for models with a pretrained inference network and one without it. Our approach achieves the best NLL in the setting without a pretrained inference network on both datasets and is competitive in the setting with a pretrained encoder. Moreover, we can observe that:

• δ-VAE does not perform well in both settings, which shows that constraining the parameters in a small interval is harmful to the model. In vMF-VAE, data points share the same KL value. Our approach is flexible and gets better performance.

• Although Agg-VAE and SA-VAE both get good performance, they require additional updates on the inference network and cost more training efforts, which are validated in the next part.

• Cyclic annealing with a pretrained inference network achieves the highest KL, but it may not be a good generative model.

• Paired with a pretrained inference network, all methods except cyclic annealing can someway boost the performance. This phenomenon indicates that the lagging problem (He et al., 2019) is important in VAE training. When leveraging the pretrained inference network, our approach achieves the smallest performance gap compared with other methods. In other words, our approach can alleviate the lagging problem efficiently.

Training time: Table 2 shows the training time (until convergence) and the relative ratio of the basic VAE, our approach and the other best three models in Table 1. SA-VAE is about 12 times slower than our approach due to the local update for each data point. Agg-VAE is 2-4 times slower than ours because it requires additional training for the inference network. AE+FB needs to train an autoencoder before the VAE. However, our approach is fast since we only add one-layer batch normalization, and thus the training cost is almost the same as the basic VAE. More results about the training behavior can be found in Section 3 of the Appendix.

#label	100	500	1k	2k	10k
AE	81.1	86.2	90.3	89.4	94.1
VAE	66.1	82.6	88.4	89.6	94.5
δ-VAE	61.8	61.9	62.6	62.9	93.8
Agg-VAE	80.9	85.9	88.8	90.6	93.7
cyclic	62.4	75.5	80.3	88.7	94.2
FB (9)	79.8	84.4	88.8	91.12	94.7
AE+FB (6)	87.6	90.2	92.0	93.4	94.9
BN-VAE (0.7)	**88.8**	**91.6**	**92.5**	**94.1**	**95.4**

Table 3: Accuracy on Yelp.

Model	CVAE	CVAE (BOW)	BN-VAE
PPL	36.40	24.49	30.67
KL	0.15	9.30	5.18
BLEU-4	10.23	8.56	8.64
A-bow Prec	95.87	96.89	96.64
A-bow Recall	90.93	93.95	94.43
E-bow Prec	86.26	83.55	84.69
E-bow Recall	77.91	81.13	81.75

Table 4: Comparison on dialogue generation.

Performance on a downstream task - Text classification: The goal of VAE is to learn a good representation of the data for downstream tasks. Here, we evaluate the quality of latent representations by training a one-layer linear classifier based on the mean of the posterior distribution. We use a downsampled version of the Yelp sentiment dataset (Shen et al., 2017). Li et al. (2019) further sampled various labeled data to train the classifier. To compare with them fairly, we use the same samples in Li et al. (2019). Results are shown in Table 3. Our approach achieves the best accuracy in all the settings. For 10k training samples, all the methods get a good result. However, when only using 100 training samples, different methods vary a lot in accuracy. The text classification task shows that our approach can learn a good latent representation even without a pretrained inference network.

5.2 CVAE for Dialogue Generation

Setup: For dialogue generation, we test our approach in the setting of CVAE. Following previous work (Zhao et al., 2017b), we use the Switchboard (SW) Corpus (Godfrey and Holliman, 1997), which contains 2400 two-sided telephone conversations.

Model	Fluency			Relevance			Informativeness		
	Avg	#Accept	#High	Avg	#Accept	#High	Avg	#Accept	#High
CVAE	2.11 (0.58)	87%	23%	1.90 (0.49)	82%	8%	1.39 (0.59)	34%	5%
CVAE (BOW)	2.08 (0.73)	84%	23%	1.86 (0.58)	75%	11%	**1.54** (0.65)	46%	8%
BN-CVAE	**2.16** (0.71)	88%	27%	**1.92** (0.67)	80%	12%	**1.54** (0.67)	43%	10%

Table 5: Human evaluation results. Numbers in parentheses is the corresponding variance on 200 test samples.

Topic: ETHICS IN GOVERNMENT		
Context: have trouble drawing lines as to what's illegal and what's not		
Target (statement): well i mean the other problem is that they're always up for		
CVAE	CVAE (BOW)	BN-CVAE
1. yeah	1. yeah	1. it's not a country
2. yeah	2. oh yeah they're not	2. it is the same thing that's what i think is about the state is a state
3. yeah	3. no it's not too bad	3. yeah it's

Table 6: Sampled generated responses. Only the last sentence in the context is shown here.

We use a bidirectional GRU with hidden size 300 to encode each utterance and then a one-layer GRU with hidden size 600 to encode previous k-1 utterances as the context. The response decoder is a one-layer GRU with hidden size 400. The latent representation $\mathbf{z}$ has a size of 200. We use the evaluation metrics from Zhao et al. (2017b): (1) Smoothed Sentence-level BLEU (Chen and Cherry, 2014); (2) Cosine Distance of Bag-of-word Embedding, which is a simple method to obtain sentence embeddings. We use the pretrained Glove embedding (Pennington et al., 2014) and denote the average method as A-*bow* and the extreme method as E-*bow*. Higher values indicate more plausible responses. We compared our approach with CVAE and CVAE with bag-of-words (BOW) loss (Zhao et al., 2017b), which requires the decoder in the generation network to predict the bag-of-words in the response $\mathbf{y}$ based on $\mathbf{z}$.

Automatic evaluation: Table 4 shows the results of these three approaches. From the KL values, we find that CVAE suffers from posterior collapse while CVAE (BOW) and our approach avoid it effectively. For BLEU-4, we observe the same phenomenon in the previous work (Fu et al., 2019; Zhao et al., 2017b) that CVAE is slightly better than the others. This is because CVAE tends to generate the most likely and safe responses repeatedly with the collapsed posterior. As for precision, these three models do not differ much. However, CVAE (BOW) and our BN-VAE outperform CVAE in recall with a large margin. This indicates that BN-VAE can also produce diverse responses with good quality like CVAE (BOW).

Human evaluation: We conduct the human evaluation by asking five annotators from a commercial annotation company to grade 200 sampled conversations from the aspect of fluency, relevance and informativeness on a scale of 1-3 (see Section 4 of the Appendix for more details on the criteria). We also report the proportion of acceptable/high scores (≥ 2 and $= 3$) on each metric. Table 5 shows the annotation results. Overall, our approach beats the other two compared methods in relevance and fluency with more informative responses. Also, our approach has the largest proportion of responses whose scores are *High*. This indicates that our model can produce more meaningful and relevant responses than the other two.

Case study: Table 6 shows the sampled responses generated by the three methods (more can be found in the Appendix). By maintaining a reasonable KL, responses generated by our approach are more relevant to the query with better diversity compared to the other two. We test the three methods in the simplest setting of dialogue generation. Note that the focus of this work is to improve the CVAE itself by avoiding its KL vanishing problem but not to hack the state-of-the-art dialogue generation performance. To further improve the quality of generated responses, we can enhance our approach by incorporating knowledge such as dialogue acts (Zhao et al., 2017b), external facts (Ghazvininejad et al., 2018) and personal profiles (Zhang et al., 2018).

6 Conclusions and Future Work

In this paper, we tackle the posterior collapse problem when VAE is paired with autoregressive decoders. Instead of considering the KL individually, we make it follow a distribution D_{KL} and show that keeping the expectation of D_{KL} positive is sufficient to prevent posterior collapse. We propose Batch Normalized VAE (BN-VAE), a simple but effective approach to set a lower bound of D_{KL}

by regularization the approximate posterior's parameters. Our approach can also avoid the recently proposed lagging problem efficiently without additional training efforts. We show that our approach can be easily extended to CVAE. We test our approach on three real applications, language modeling, text classification and dialogue generation. Experiments show that our approach outperforms strong baselines and is competitive with more complex methods which keeping substantially faster.

We leverage the Gaussian prior as the example to introduce our method in this work. The key to our approach to be applicable is that we can get a formula for the expectation of the KL. However, it is hard to get the same formula for some more strong or sophisticated priors, e.g., the Dirichlet prior. For these distributions, we can approximate them by the Gaussian distributions (such as in Srivastava and Sutton (2017)). In this way, we can batch normalize the corresponding parameters. Further study in this direction may be interesting.

References

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. 2018. Fixing a broken elbo. In *ICML*.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. Deep variational information bottleneck. In *ICLR*.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *CONLL*.

Yuri Burda, Roger B. Grosse, and Ruslan R. Salakhutdinov. 2016. Importance weighted autoencoders. In *ICLR*.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*.

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2017. Variational lossy autoencoder. In *ICLR*.

Chris Cremer, Xuechen Li, and David Duvenaud. 2018. Inference suboptimality in variational autoencoders. In *ICML*.

Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. 2018. Hyperspherical variational auto-encoders. In *UAI*.

Adji B Dieng, Yoon Kim, Alexander M Rush, and David M Blei. 2019. Avoiding latent variable collapse with generative skip models. In *AISTATS*.

Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. 2019. Implicit deep latent variable models for text generation. In *EMNLP-IJCNLP*.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In *NAACL*.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *AAAI*.

J Godfrey and E Holliman. 1997. Switchboard-1 release 2: Linguistic data consortium. In *SWITCHBOARD: A User's Manual*.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. In *Transactions of the Association of Computational Linguistics*. MIT Press.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging inference networks and posterior collapse in variational autoencoders. In *ICLR*.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. In *Neural computation*. MIT Press.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Yoon Kim, Sam Wiseman, Andrew C. Miller, David A Sontag, and Alexander M. Rush. 2018. Semi-amortized variational autoencoders. In *ICML*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *ICLR*.

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

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. Improved variational inference with inverse autoregressive flow. In *NeurIPS*.

Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. 2019. A surprisingly effective fix for deep latent variable modeling of text. In *EMNLP-IJCNLP*.

Xuezhe Ma, Chunting Zhou, and Eduard Hovy. 2019. MAE: Mutual posterior-divergence regularization for variational autoencoders. In *ICLR*.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *ICML*.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. 2017. Neural discrete representation learning. In *NeurIPS*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Ali Razavi, Aaron van den Oord, Ben Poole, and Oriol Vinyals. 2019. Preventing posterior collapse with delta-VAEs. In *ICLR*.

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

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. 2018. How does batch normalization help optimization? In *NeurIPS*.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. In *EMNLP*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *NeurIPS*.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *NeurIPS*.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. 2016. Ladder variational autoencoders. In *NeurIPS*.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *ICLR*.

Akash Srivastava and Charles Sutton. 2018. Variational inference in pachinko allocation machines. In *arXiv preprint arXiv:1804.07944*.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2018. Wasserstein autoencoders. In *ICLR*.

Jakub M. Tomczak and Max Welling. 2018. Vae with a vampprior. In *AISTATS*.

Jiacheng Xu and Greg Durrett. 2018. Spherical latent spaces for stable variational autoencoders. In *EMNLP*.

Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. 2017. Variational autoencoder for semi-supervised text classification. In *AAAI*.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *ICML*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *ACL*.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2017a. Infovae: Information maximizing variational autoencoders. In *arXiv preprint arXiv:1706.02262*.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. 2018. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *ACL*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017b. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*.

Qile Zhu, Zheng Feng, and Xiaolin Li. 2018. Graphbtm: Graph enhanced autoencoded variational inference for biterm topic model. In *EMNLP*.

A Appendix

A.1 Experiments on Synthetic Data

We follow the Agg-VAE and construct the synthetic data to validate whether our approach can avoid the lagging problem. VAE used in this synthetic task has a LSTM encoder and a LSTM decoder. We use a scalar latent variable because we need to compute $\mu_{x,\theta}$ which is approximated by discretization of $p_\theta(z|x)$. To visualize the training progress, we sample 500 data points from the validation set and show them on the mean space.

We plot the mean value of the approximate posterior and the model posterior during training for the basic VAE and BN-VAE. As shown the first column in Fig. 1, all points have the zero mean of the model posterior (the x-axis), which indicates that $\mathbf{z}$ and $\mathbf{x}$ are independent at the beginning of training. For the basic VAE, points start to spread in the x-axis during training while sharing almost the same y value, since the model posterior $p_\theta(\mathbf{z}|\mathbf{x})$ is well learned with the help of the autoregressive decoder. However, the inference posterior $q_\phi(\mathbf{z}|\mathbf{x})$ is lagging behind $p_\theta(\mathbf{z}|\mathbf{x})$ and collapses to the prior in the end. Our regularization approximated by BN, on the other hand, pushes the inference posterior $q_\phi(\mathbf{z}|\mathbf{x})$ away from the prior ($p(\mathbf{z})$) at the initial training stage, and forces $q_\phi(\mathbf{z}|\mathbf{x})$ to catch up with $p_\theta(\mathbf{z}|\mathbf{x})$ to minimize $KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))$ in Eq. 9. As in the second row of Fig. 1, points spread in both directions and towards the diagonal.

We also report the results on different γ's with different batch sizes (32 in Fig. 1). Fig. 2 shows the training dynamics. Both settings of γ avoid posterior collapse efficiently. A larger γ produces more diverse μ's which spread on the diagonal. However, a small γ results in a small variance for the distribution of μ, thus μ's in the bottom row are closer to the original (mean of the distribution). When γ is 0, posterior collapse happens. Different batch sizes do not diff a lot, so 32 is a decent choice. An intuitive improvement of our method is to automatically learn different γ for different latent dimensions, which we leave for future work.

A.2 Proof in CVAE

The KL can be computed as:

$$KL = \frac{1}{2}\sum_{i=1}^{n}(\frac{\sigma_{qi}^2 + (\mu_{qi} - \mu_{pi})^2}{\sigma_{pi}^2} + \sigma_{pi}^2 + \mu_{pi}^2 - log\sigma_{qi}^2 - 1). \quad (14)$$

We need to prove that KL will not achieve the minimum number when μ_{pi} equals to μ_{qi} and σ_{pi} equals σ_{qi}. We take hidden size as 1 for example. The binary function about μ_{pi} and σ_{pi} is:

$$f_{\mu_{pi},\sigma_{pi}} = (\frac{\sigma_{qi}^2 + (\mu_{qi} - \mu_{pi})^2}{\sigma_{pi}^2} + \sigma_{pi}^2 + \mu_{pi}^2 - log\sigma_{qi}^2 - 1), \quad (15)$$

the maxima and minima of $f_{\mu_{pi},\sigma_{pi}}$ must be the stationary point of $f_{\mu_{pi},\sigma_{pi}}$ due to its continuity. The stationary point is:

$$\frac{\partial f}{\partial \mu_{pi}} = \frac{2(\mu_{pi} - \mu_{qi})}{\sigma_{pi}^2} + 2\mu_{pi} \quad (16)$$

$$\frac{\partial f}{\partial \sigma_{pi}} = \frac{-2(\sigma_{qi}^2 + (\mu_{qi} - \mu_{pi})^2)}{\sigma_{pi}^3} + 2\sigma_{pi}. \quad (17)$$

When $\mu_{pi} = \mu_{qi}$ and $\sigma_{pi} = \sigma_{qi}$, both partial derivative is not 0. So it is not the stationary point of f, then it won't be the minima.

A.3 Language Modeling

We investigate the training procedure for different models. We plot the MI I_q, D_{KL} in the ELBO and the distance between the approximated posterior and the prior, $D_{KL}(q_\phi(z)||p(z))$. As in Eq. 4 in the main paper, D_{KL} in the ELBO is the sum of the other two. Fig. 3 shows these three values throughout the training. Although D_{KL} is the upper bound of the mutual information, we notice that the gap is usually large. In the initial training stage, D_{KL} increases in the basic VAE with annealing, while its MI remains small. With the weight decreases, the method finally suffers from posterior collapse. In contrast, our approach can obtain a high MI with a small D_{KL} value like aggressive VAE. The full results on language modeling are in Table 8.

A.4 CVAE for dialogue generation

Human evaluation: We evaluate the generated responses from three aspects: relevance, fluency and informativeness. Here we introduce the criteria of the evaluation as shown in Table 7. We sample 200 conversations from the test set. For each conversation, we sample three generated responses from each model, totally 600 responses.

Case study: We report 4 examples generated from these three models, shown in Table 9. CVAE (BOW) and our approach both can generate diverse responses. However, responses from ours are more related to the context compared with the other two.

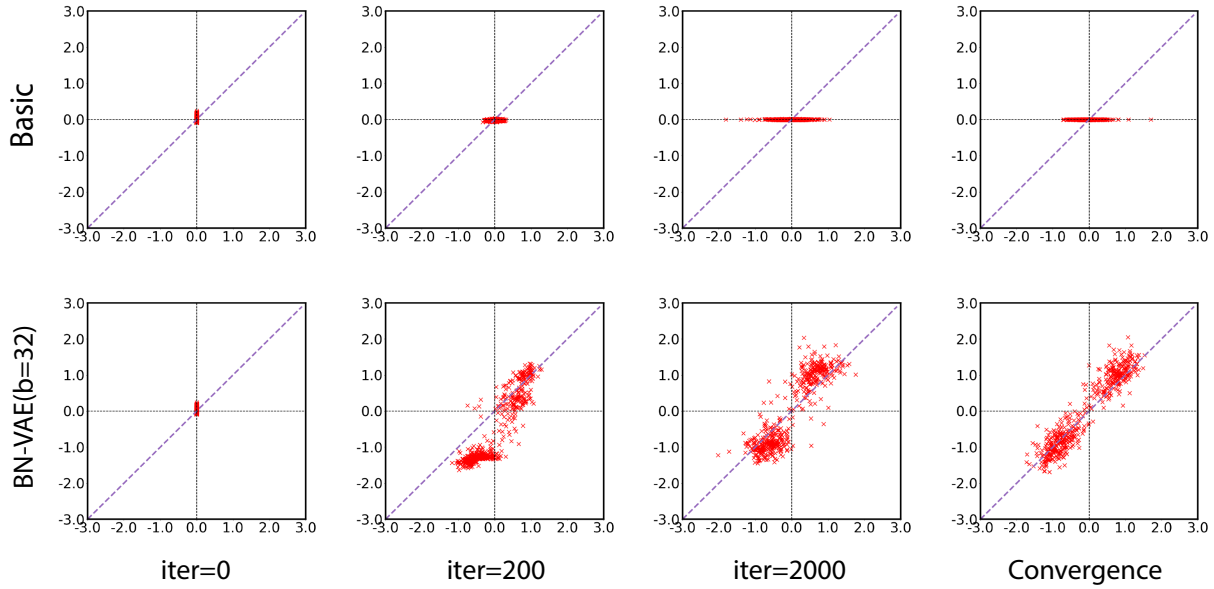

Figure 1: Visualization of 500 sampled data from the synthetic dataset during the training. The x-axis is $\mu_{x,\theta}$, the approximate model posterior mean. The y-axis is $\mu_{x,\phi}$, which represents the inference posterior mean. b is batch size and γ is 1 in BN.

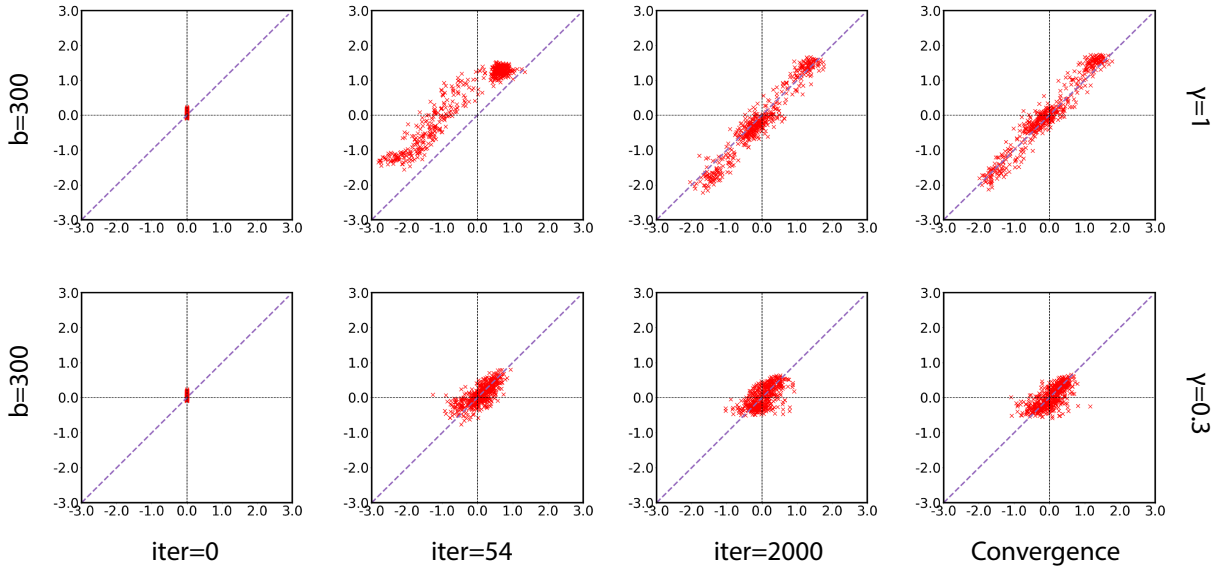

Figure 2: Visualization of our BN-VAE on different γ for synthetic data.

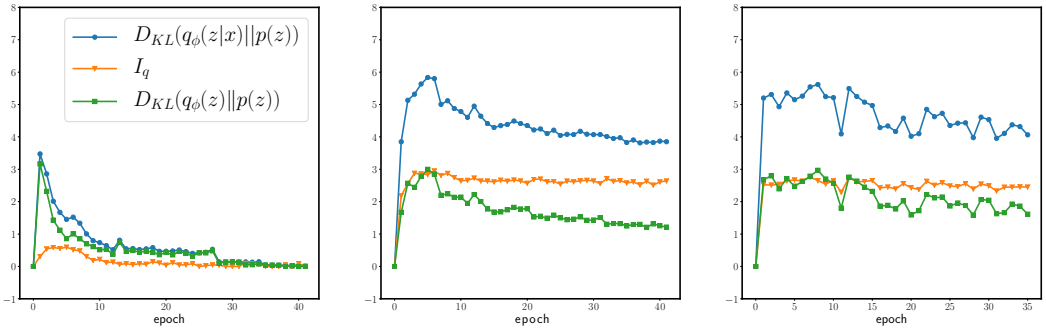

Figure 3: Training behavior on Yelp. Left/Middle/Right: VAE/Agg-VAE/BN-VAE (all models are with annealing).

Table 7: Human evaluation criteria.

	Fluency	Relevance	Informativeness
1 Point	1. Hard to understand 2. Too many syntax mistakes	Not related to the query at all	1. Generic responses. 2. Repeated query.
2 Points	1. Several syntax mistakes but still understandable 2. short responses, e.g., Generic responses	1. Response and query are in the same domain/topic but are not directly related 2. Generic responses	between 1 and 3.
3 Points	Only few syntax mistakes with a moderate length	closely related to the query	1. Creative responses. 2. Contain new information about the query.

	Yahoo				Yelp			
Model	**NLL**	**KL**	**MI**	**AU**	**NLL**	**KL**	**MI**	**AU**
CNN-VAE	$\leq$332.1	10.0	-	-	$\leq$359.1	7.6	-	-
LSTM-LM	328	-	-	-	351.1	-	-	-
VAE	328.6	0.0	0.0	0.0	357.9	0.0	0.0	0.0
β-VAE (0.2)	332.2	19.1	3.3	20.4	360.7	11.7	3.0	10.0
β-VAE (0.4)	328.7	6.3	2.8	8.0	358.2	4.2	2.0	4.2
β-VAE (0.6)	328.5	0.3	0.0	1.0	357.9	0.2	0.1	3.8
β-VAE (0.8)	328.8	0.0	0.0	0.0	358.1	0.0	0.0	0.0
cyclic *	330.6	2.1	2.0	2.3	359.5	2.0	1.9	4.1
Skip-VAE *	328.5	2.3	1.3	8.1	357.6	1.9	1.0	7.4
SA-VAE	327.2	5.2	2.7	9.8	**355.9**	2.8	1.7	8.4
Agg-VAE	**326.7**	5.7	2.9	15.0	**355.9**	3.8	2.4	11.3
FB (4)	331.0	4.1	3.8	3.0	359.2	4.0	1.9	32.0
FB (5)	330.6	5.7	2.0	3.0	359.8	4.9	1.3	32.0
δ-VAE (0.1) *	330.7	3.2	0.0	0.0	359.8	3.2	0.0	0.0
δ-VAE (0.15) *	331.6	4.8	0.0	0.0	360.4	4.8	0.0	0.0
δ-VAE (0.2) *	332.2	6.4	0.0	0.0	361.5	6.4	0.0	0.0
δ-VAE (0.25) *	333.5	8.0	0.0	0.0	362.5	8.0	0.0	0.0
vMF-VAE (13) *	327.4	2.0	-	32.0	357.5	2.0	-	32.0
vMF-VAE (16) *	328.5	3.0	-	32.0	367.8	3.0	-	32.0
vMF-VAE (20) *	329.4	4.0	–	32.0	358.0	4.0	-	32.0
vMF-VAE (23) *	328.7	5.0	-	32.0	357.3	5.0	-	32.0
vMF-VAE (25) *	330.1	6.0	-	32.0	357.8	6.0	-	32.0
vMF-VAE (30) *	329.5	7.0	-	32.0	357.8	7.0	-	32.0
BN-VAE (0.3) *	328.1	1.6	1.4	32.0	356.7	1.7	1.4	32.0
BN-VAE (0.4) *	327.7	2.7	2.2	32.0	356.2	3.1	2.5	32.0
BN-VAE (0.5) *	327.4	4.2	3.3	32.0	356.4	4.4	3.8	32.0
BN-VAE (0.6) *	**326.7**	6.2	5.6	32.0	356.5	6.5	5.4	32.0
BN-VAE (0.7) *	327.4	8.8	7.4	32.0	**355.9**	9.1	7.4	32.0
Pretrained encoder								
+cyclic *	333.1	25.8	9.1	32.0	361.5	20.5	9.3	32.0
+FB (2) *	327.2	4.3	3.8	32.0	356.6	4.6	4.2	32.0
+FB (3) *	327.1	4.5	3.9	32.0	356.3	5.8	5.2	32.0
+FB (4) *	**326.2**	8.1	6.8	32.0	356.0	7.6	6.6	32.0
+FB (5) *	326.6	8.9	7.3	32.0	356.5	9.0	7.4	32.0
+FB (6) *	326.6	10.8	8.1	32.0	356.5	12.0	8.6	32.0
+FB (7) *	326.6	12.1	8.5	32.0	356.8	13.4	8.9	32.0
+FB (8) *	326.7	13.6	8.9	32.0	357.5	15.8	9.2	32.0
+δ-VAE (0.15) *	331.0	5.6	1.1	11.2	359.4	5.2	0.5	5.9
vMF-VAE (13) *	328.4	2.0	-	32.0	357.0	2.0	-	32.0
+BN-VAE (0.6) *	326.7	6.4	5.8	32.0	**355.5**	6.6	5.9	32.0
+BN-VAE (0.7) *	326.5	9.1	7.6	32.0	355.7	9.1	7.5	32.0

Table 8: Results on Yahoo and Yelp datasets. We report mean values across 5 different random runs. * indicates the results are from our experiments, while others are from previous report.

Table 9: Sampled generated responses. Only the last sentence in the context is shown here.

Topic: ETHICS IN GOVERNMENT		
Context: have trouble drawing lines as to what's illegal and what's not		
Target (statement): well i mean the other problem is that they'are always up for		
CVAE	CVAE (BOW)	BN-CVAE
1. yeah	1. yeah	1. it's not a country
2. yeah	2. oh yeah, they're not	2. it is the same thing that's what i think is about the state is a state
3. yeah	3. no it's not too bad	3. yeah it's
Topic:VACATION SPOTS		
Context: well i ' ll talk to you later		
Target (conventional-closing) : okay now do you push the buttons now		
CVAE	CVAE (BOW)	BN-CVAE
1. okay bye - bye	1. okay so we'll go ahead and start	1. okay bye - bye
2. bye	2. so i guess it depends on how much you are you	2. nice talking to you too
3. okay bye - bye	3. it's	3. all right take care bye - bye
Topic: RECYCLING		
Context: are they doing a lot of recycling out in georgia		
Target (statement-non-opinion) : well at my workplace they are we have places for aluminum cans and we have everybody's been unk a separate trash can for		
CVAE	CVAE (BOW)	BN-CVAE
1. yeah	1. well that's good for a while i'll tell you that you're not doing anything at ti and then you're in a small town	1. well we do recycle newspapers
2. yeah	2. oh i know i've got a lot of trees and trees and stuff and	2. yes i do too
3. yeah	3. yeah it's like you know people that want to be unk and they're not going to bother you to make a mess	3. well we're at a point where we're going to be a landfill space
Topic: UNIVERSAL HEALTH INS		
Context: some of the good obviously that nobody has to worry about health care		
Target (statement-non-opinion) : and i guess i'll have to help with grandchildren one of these days i hope		
CVAE	CVAE (BOW)	BN-CVAE
1. um - hum	1. okay well see we don't have any choice of any of those	1. well i hope that we should have a balanced budget
2. uh - huh	2.um - hum	2. uh - huh
3. uh - huh	3. yeah	3. well that's a good idea