Implicit Deep Latent Variable Models for Text Generation

Le Fang†, Chunyuan Li§, Jianfeng Gao§, Wen Dong†, Changyou Chen†
†University at Buffalo, §Microsoft Research, Redmond
{lefang, wendong, changyou}@buffalo.edu
{chunyl, jfgao}@microsoft.com

Abstract

Deep latent variable models (LVM) such as variational auto-encoder (VAE) have recently played an important role in text generation. One key factor is the exploitation of smooth latent structures to guide the generation. However, the representation power of VAEs is limited due to two reasons: (1) the Gaussian assumption is often made on the variational posteriors; and meanwhile (2) a notorious “posterior collapse” issue occurs. In this paper, we advocate sample-based representations of variational distributions for natural language, leading to implicit latent features, which can provide flexible representation power compared with Gaussian-based posteriors. We further develop an LVM to directly match the aggregated posterior to the prior. It can be viewed as a natural extension of VAEs with a regularization of maximizing mutual information, mitigating the “posterior collapse” issue. We demonstrate the effectiveness and versatility of our models in various text generation scenarios, including language modeling, unaligned style transfer, and dialog response generation. The source code to reproduce our experimental results is available on GitHub†.

1 Introduction

Deep latent variable models (LVM) such as variational auto-encoder (VAE) (Kingma and Welling, 2013; Rezende et al., 2014) are successfully applied for many natural language processing tasks, including language modeling (Bowman et al., 2015; Miao et al., 2016), dialogue response generation (Zhao et al., 2017b), controllable text generation (Hu et al., 2017) and neural machine translation (Shah and Barber, 2018) etc. One advantage of VAEs is the flexible distribution-based latent representation. It captures holistic properties of input, such as style, topic, and high-level linguistic/semantic features, which further guide the generation of diverse and relevant sentences.

However, the representation capacity of VAEs is restrictive due to two reasons. The first reason is rooted in the assumption of variational posteriors, which usually follow spherical Gaussian distributions with diagonal co-variance matrices. It has been shown that an approximation gap generally exists between the true posterior and the best possible variational posterior in a restricted family (Cremer et al., 2018). Consequently, the gap may militate against learning an optimal generative model, as its parameters may be always updated based on sub-optimal posteriors (Kim et al., 2018). The second reason is the so-called posterior collapse issue, which occurs when learning VAEs with an auto-regressive decoder (Bowman et al., 2015). It produces undesirable outcomes: the encoder yields meaningless posteriors that are very close to the prior, while the decoder tends to ignore the latent codes in generation (Bowman et al., 2015). Several attempts have been made to alleviate this issue (Bowman et al., 2015; Higgins et al., 2017; Zhao et al., 2017a; Fu et al., 2019; He et al., 2019).

These two seemingly unrelated issues are studied independently. In this paper, we argue that the posterior collapse issue is partially due to the restrictive Gaussian assumption, as it limits the optimization space of the encoder/decoder in a given distribution family. (i) To break the assumption, we propose to use sample-based representations for natural language, thus leading to implicit latent features. Such a representation is much more expressive than Gaussian-based posteriors. (ii) This implicit representation allows us to extend VAE and develop new LVM that further mitigate the posterior collapse issue. It represents all the sentences in the dataset as posterior samples in the latent space, and matches the aggregated posterior samples to the prior distribution. Consequently,
latent features are encouraged to cooperate and behave diversely to capture meaningful information for each sentence.

However, learning with implicit representations faces one challenge: it is intractable to evaluate the KL divergence term in the objectives. We overcome the issue by introducing a conjugate-dual form of the KL divergence (Rockafellar et al., 1966; Dai et al., 2018). It facilitates learning via training an auxiliary dual function. The effectiveness of our models is validated by producing consistently state-of-the-art results on a broad range of generation tasks, including language modeling, unsupervised style transfer, and dialog response generation.

2 Preliminaries

When applied to text generation, VAEs (Bowman et al., 2015) consist of two parts, a generative network (decoder) and an inference network (encoder). Given a training dataset \( D = \{ x_i \}_{i=1}^{|D|} \), where \( x_i = [x_{i1}, \ldots, x_{iT}] \) represents \( i \)th sentence of length \( T \). Starting from a prior distribution \( p(z) \), VAE generates a sentence \( x \) using the deep generative network \( p_\theta(x|z) \), where \( \theta \) is the network parameter. Therefore, the joint distribution \( p_\theta(x, z) \) is defined as \( p(z)p_\theta(x|z) \). The prior \( p(z) \) is typically assumed as a standard multivariate Gaussian. Due to the sequential nature of natural language, the decoder \( p_\theta(x|z) \) takes an auto-regressive form \( p_\theta(x|z) = \prod_{t=1}^T p_\theta(x_t|x_{<t}, z) \).

The goal of model training is to maximize the variational distribution \( q_\phi(z|x) \), which is the net-

\[ \mathcal{L}_R = \mathbb{E}_{x \sim p} \left[ \log p_\theta(x \mid z) \right] + \text{KL} \left( q_\phi(z \mid x) \parallel p(z) \right) \]

\[ \mathcal{L}_G = -\mathbb{E}_{x \sim p} \text{KL} \left( q_\phi(z \mid x) \parallel p(x) \right) \]

Note that \( \mathcal{L}_1 \) and \( \mathcal{L}_2 \) provide two different views for the VAE objective:

- \( \mathcal{L}_1 \) consists of a reconstruction error term \( \mathcal{L}_E \) and a KL divergence regularization term \( \mathcal{L}_R \). With a strong auto-regressive decoder \( p_\theta(x|z) \), the objective tends to degenerate all encoding distribution \( q_\phi(z|x) \) to the prior, causing \( \mathcal{L}_R \to 0 \), i.e., the posterior collapse issue.

- \( \mathcal{L}_2 \) indicates that VAE requests a flexible encoding distribution family to minimize the approximation gap \( \mathcal{L}_G \) between the true posterior and the best possible encoding distribution. This motivates us to perform more flexible posterior inference with implicit representations.

3 The Proposed Models

We introduce a sample-based latent representation for natural language, and develop two models that leverage its advantages. (1) When replacing the Gaussian variational distributions with sample-based distributions in VAEs, we derive implicit VAE (iVAE). (2) We further extend VAE to maximize mutual information between latent representations and observed sentences, leading to a variant termed as iVAEMI.

3.1 Implicit VAE

**Implicit Representations** Instead of assuming an explicit density form such as Gaussian, we define a sampling mechanism to represent \( q_\phi(z|x) \) as a set of sample \( \{z_{x,i}\}_{i=1}^M \), through the encoder as

\[ z_{x,i} = G_\phi(x, \epsilon_i), \epsilon_i \sim q(\epsilon) \]

where the \( i \)-th sample is drawn from a neural network \( G_\phi \) that takes \( (x, \epsilon_i) \) as input; \( q(\epsilon) \) is a simple distribution such as standard Gaussian. It is difficult to naively combine the random noise \( \epsilon \) with the sentence \( x \) (a sequence of discrete tokens) as the input of \( G_\phi \). Our solution is to concatenate noise \( \epsilon_i \) with hidden representations \( h \) of \( x \). \( h \) is generated using a LSTM encoder, as illustrated in Figure 1.

**Dual form of KL-divergence** Though theoretically promising, the implicit representations in (6) render difficulties in optimizing the KL term \( \mathcal{L}_R \) in (3), as the functional form is no longer tractable with implicit \( q_\phi(z|x) \). We resort to evaluating its dual form based on Fenchel duality theorem (Rockafellar et al., 1966; Dai et al., 2018):

\[ \text{KL} \left( q_\phi(z \mid x) \parallel p(z) \right) = \max_{\nu} \mathbb{E}_{z \sim q_\phi(z \mid x)} \nu \psi(x, z) - \mathbb{E}_{z \sim p(z)} \exp(\nu \psi(x, z)) \]
3.2 Mutual Information Regularized iVAE

It is noted the inherent deficiency of the original VAE objective in (3): the KL divergence regularization term matches each posterior distribution independently to the same prior. This is prone to posterior collapse in text generation, due to a strong auto-regressive decoder \(p_{\theta}(x|z)\). When sequentially generating \(x_t\), the model learns to solely rely on the ground-truth \([x_1, \cdots, x_{t-1}]\), and ignore the dependency from \(z\) (Fu et al., 2019). It results in the learned variational posteriors \(q_{\phi}(z|x)\) to exactly match \(p(z)\), without discriminating data \(x\).

To better regularize the latent space, we propose to replace \(L_{VAE} = \mathbb{E}_{x \sim D} [-KL(q_{\phi}(z|x) \parallel p(z))]\) in (3), with the following KL divergence:

\[
-L_{KL}(q_{\phi}(z) \parallel p(z)),
\]

where \(q_{\phi}(z) = \int q(z|x)q_{\phi}(z|x)dx\) is the aggregated posterior, \(q(z)\) is the empirical data distribution for the training dataset \(D\). The integral is estimated by ancestral sampling in practice, \textit{i.e.} we first sample \(x\) from dataset and then sample \(z \sim q_{\phi}(z|x)\).

In (11), variational posterior is regularized as a whole \(q_{\phi}(z)\), encouraging posterior samples from different sentences to cooperate to satisfy the objective. It implies a solution that each sentence is represented as a local region in the latent space, the aggregated representation of all sentences match the prior; This avoids the degenerated solution from (3) that the feature representation of individual sentence spans over the whole space.

Connection to mutual information The proposed latent variable model coincides with (Zhao et al., 2018, 2017a) where mutual information is introduced into the optimization, based on the following decomposition result (Please see detailed...
proof in Appendix A):
\[-\text{KL}(q_\phi(z) \parallel p(z)) = I(x, z) \]
\[-\mathbb{E}_x \text{KL}(q_\phi(z|x) \parallel p(z)),\]
where $I(x, z)$ is the mutual information between $z$ and $x$ under the joint distribution $q_\phi(x, z) = q(x)q_\phi(z|x)$.
Therefore, the objective in (11) also maximizes the mutual information between individual sentences and their latent features. We term the new LVM objective as iVAE MI:
\[
\mathcal{L}_{\text{VAE MI}} = \mathbb{E}_{x \sim D} \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \text{KL}(q_\phi(z) \parallel p(z))
\]  
(12)

**Training scheme** Note that the aggregated posterior $q_\phi(z)$ is also a sample-based distribution. Similarly, we evaluate (12) through its dual form:
\[
\text{KL}(q_\phi(z) \parallel p(z)) = \max_\nu \mathbb{E}_{z \sim q_\phi(z)} \log \nu_\phi(z) - \mathbb{E}_{z \sim p(z)} \log \nu_\phi(z).
\]  
(13)
Therefore, iVAE MI in (12) can be written as:
\[
\mathcal{L}_{\text{VAE MI}} = \mathbb{E}_{x \sim D} \mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) - \mathbb{E}_{z \sim q_\phi(z)} \log \nu_\phi(z) + \mathbb{E}_{z \sim p(z)} \log \nu_\phi(z),
\]  
(14)
where the auxiliary network $\nu_\phi(z)$ is parameterized as a neural network. Different from iVAE, $\nu_\phi(z)$ in iVAE MI only takes posterior samples as input. The training algorithm is similar to iVAE in Section 3.1, except a different auxiliary network $\nu_\phi(z)$. In Appendix B, we show the full algorithm of iVAE MI.

We illustrate the proposed methods in Figure 1. Note that both iVAE and iVAE MI share the same model architecture, except a different auxiliary network $\nu$.

4 Related Work

4.1 Solutions to posterior collapse

Several attempts have been made to alleviate the posterior collapse issue. The KL annealing scheme in language VAEs has been first used in (Bowman et al., 2015). An effective cyclical KL annealing schedule is used in (Fu et al., 2019), where the KL annealing process is repeated multiple times. KL term weighting scheme is also adopted in $\beta$-VAE (Higgins et al., 2017) for disentanglement. On model architecture side, dilated CNN was considered to replace auto-regressive LSTMs for decoding (Yang et al., 2017). The bag-of-word auxiliary loss was proposed to improve the dependence on latent representations in generation (Zhao et al., 2017b). More recently, lagging inference proposes to aggressively update encoder multiple times before a single decoder update (He et al., 2019). Semi-amortized VAE refines variational parameters from an amortized encoder per instance with stochastic variational inference (Kim et al., 2018).

All these efforts utilize the Gaussian-based forms for posterior inference. Our paper is among the first ones to attribute posterior collapse issue to the restrictive Gaussian assumption, and advocate more flexible sample-based representations.

4.2 Implicit Feature Learning

Sample-based distributions, as well as implicit features, have been widely used in representation learning (Donahue et al., 2017; Li et al., 2017a). Vanilla autoencoders learn point masses of latent features rather than their distributions. Adversarial variational Bayes introduces an auxiliary discriminator network like GANs (Goodfellow et al., 2014; Makhzani et al., 2015) to learn almost arbitrarily distributed latent variables (Mescheder et al., 2017; Pu et al., 2017b). We explore the similar spirit in the natural language processing (NLP) domain. Amortized MCMC and particle based methods are introduced for LVM learning in (Li et al., 2017d; Pu et al., 2017a; Chen et al., 2018). Coupled variational Bayes (Dai et al., 2018) emphasizes an optimization embedding, i.e., a flow of particles, in a general setting of non-parametric variational inference. It also utilizes similar dual form with auxiliary function $\nu_\phi(x, z)$ to evaluate KL divergence. Adversarially regularized autoencoders (Makhzani et al., 2015; Kim et al., 2017) use similar objectives with iVAEs, in the form of a reconstruction error plus a specific regularization evaluated with implicit samples. Mutual information has also been considered into regularization in (Zhao et al., 2018, 2017a) to obtain more informative representations.

Most previous works focus on image domain. It is largely unexplored in NLP. Further, the auto-regressive decoder renders an additional challenge when applying implicit latent representations. Adversarial training with samples can be empirically unstable, and slow even applying recent stabilization techniques in GANs (Arjovsky et al., 2017; Gulrajani et al., 2017). To the best of our knowledge, this paper presents the first to effectively apply implicit feature representations, to NLP.
5 Experiments

In this section, the effectiveness of our methods is validated by largely producing state-of-the-art metrics on a broad range of text generation tasks under various scenarios.

5.1 Language Modeling

Datasets. We consider three public datasets, the Penn Treebank (PTB) (Marcus et al., 1993; Bowman et al., 2015), Yahoo, and Yelp corpora (Yang et al., 2015; He et al., 2019). PTB is a relatively small dataset with sentences of varying lengths, whereas Yahoo and Yelp contain larger amounts of long sentences. Detailed statistics of these datasets are shown in Table 1.

Settings. We implement both encoder and decoder as one-layer LSTM. As illustrated in Figure 1, concatenation of the last hidden state of the encoder LSTM and a Gaussian noise is fed into a MLP to draw a posterior sample. When decoding, a latent sample is fed into a MLP to produce the initial hidden state of the decoder LSTM. A KL-annealing scheme is also used (Bowman et al., 2015). We list more detailed hyper-parameters and architectures in Appendix C.1.2.

Baselines. We compare with several state-of-the-art VAE language modeling methods, including (1) VAEs with a monotonically KL-annealing schedule (Bowman et al., 2015); (2) β-VAE (Higgins et al., 2017), VAEs with a small penalty on KL term scaled by β; (3) SA-VAE (Kim et al., 2018), mixing instance-specific variational inference with amortized inference; (4) Cyclical VAE (Fu et al., 2019) that periodically anneals the KL term; (5) Lagging VAE, which aggressively updates encoder multiple times before a single decoder update.

Evaluation metrics. Two categories of metrics are used to study VAEs for language modeling:

- To characterize the modeling ability of the observed sentences, we use the negative ELBO as the sum of reconstruction loss and KL term, as well as perplexity (PPL).
- Compared with traditional neural language models, VAEs has its unique advantages in feature learning. To measure the quality of learned features, we consider (1) KL: \(\text{KL}(q_\phi(z|x) \parallel p(z))\); (2) Mutual information (MI) \(I(x, z)\) under the joint distribution \(q_\phi(x, z)\); (3) Number of active units (AU) of latent representation. The activity of a latent dimension \(z\) is measured as \(A_z = \text{Cov}_x (E_{z \sim q(z|x)} [z])\), which is defined as active if \(A_z > 0.01\).

The evaluation of implicit LVMs is unexplored in language models, as there is no analytical forms for the KL term. We consider to evaluate both \(\text{KL}(q_\phi(z) \parallel p(z))\) and \(\text{KL}(q_\phi(z|x) \parallel p(z))\) by training a fully connected \(\nu\) network in Eq. (7) and (13). To avoid the inconsistency between \(\nu(x, z)\) and \(\nu(z)\) networks due to training, we train them using the same data and optimizer in every iteration. We evaluate each distribution \(q_\phi(z|x)\) with 128 code samples per \(x\).

We report the results in Table 2. A better language model would pursue a lower negative ELBO (also lower reconstruction errors, lower PPL), and make sufficient use of the latent space (i.e., maintain relatively high KL term, higher mutual infor-

### Table 1: Statistics of datasets for language modeling.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>PTB</th>
<th>Yahoo</th>
<th>Yelp</th>
</tr>
</thead>
<tbody>
<tr>
<td>num. sents.</td>
<td>42,000</td>
<td>100,000</td>
<td>100,000</td>
</tr>
<tr>
<td>min. len.</td>
<td>2</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>max. len.</td>
<td>78</td>
<td>201</td>
<td>201</td>
</tr>
<tr>
<td>avg. len.</td>
<td>21.9</td>
<td>79.9</td>
<td>96.7</td>
</tr>
</tbody>
</table>

### Table 2: Language modeling on three datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>-ELBO ↓</th>
<th>PPL ↓</th>
<th>KL ↑</th>
<th>MI ↑</th>
<th>AU ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE</td>
<td>PTB</td>
<td>357.9</td>
<td>40.56</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>β(0.5)-VAE</td>
<td>PTB</td>
<td>358.2</td>
<td>40.49</td>
<td>4.2</td>
<td>2.0</td>
<td>4.0</td>
</tr>
<tr>
<td>SA-VAE</td>
<td>PTB</td>
<td>355.9</td>
<td>39.73</td>
<td>2.8</td>
<td>1.7</td>
<td>8.0</td>
</tr>
<tr>
<td>iVAE</td>
<td>PTB</td>
<td>348.2</td>
<td>36.70</td>
<td>7.6</td>
<td>4.6</td>
<td>32.0</td>
</tr>
<tr>
<td>iVAE_Mi</td>
<td>PTB</td>
<td>348.7</td>
<td>36.88</td>
<td>11.6</td>
<td>11.0</td>
<td>32.0</td>
</tr>
<tr>
<td>VAE</td>
<td>Yahoo</td>
<td>357.9</td>
<td>40.56</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>β(0.4)-VAE</td>
<td>Yahoo</td>
<td>358.2</td>
<td>40.49</td>
<td>4.2</td>
<td>2.0</td>
<td>4.0</td>
</tr>
<tr>
<td>SA-VAE</td>
<td>Yahoo</td>
<td>355.9</td>
<td>39.73</td>
<td>2.8</td>
<td>1.7</td>
<td>8.0</td>
</tr>
<tr>
<td>iVAE</td>
<td>Yahoo</td>
<td>348.2</td>
<td>36.70</td>
<td>7.6</td>
<td>4.6</td>
<td>32.0</td>
</tr>
<tr>
<td>iVAE_Mi</td>
<td>Yahoo</td>
<td>348.7</td>
<td>36.88</td>
<td>11.6</td>
<td>11.0</td>
<td>32.0</td>
</tr>
<tr>
<td>VAE</td>
<td>Yelp</td>
<td>357.9</td>
<td>40.56</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>β(0.4)-VAE</td>
<td>Yelp</td>
<td>358.2</td>
<td>40.49</td>
<td>4.2</td>
<td>2.0</td>
<td>4.0</td>
</tr>
<tr>
<td>SA-VAE</td>
<td>Yelp</td>
<td>355.9</td>
<td>39.73</td>
<td>2.8</td>
<td>1.7</td>
<td>8.0</td>
</tr>
<tr>
<td>iVAE</td>
<td>Yelp</td>
<td>348.2</td>
<td>36.70</td>
<td>7.6</td>
<td>4.6</td>
<td>32.0</td>
</tr>
<tr>
<td>iVAE_Mi</td>
<td>Yelp</td>
<td>348.7</td>
<td>36.88</td>
<td>11.6</td>
<td>11.0</td>
<td>32.0</td>
</tr>
<tr>
<td>Dataset</td>
<td>PTB</td>
<td>Yahoo</td>
<td>Yelp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-----</td>
<td>-------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAE/β-1</td>
<td>1.0</td>
<td>1.3</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SA-VAE</td>
<td>5.5</td>
<td>7.1</td>
<td>9.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lag-VAE</td>
<td>-</td>
<td>-</td>
<td>2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iVAEs</td>
<td>1.4</td>
<td>1.8</td>
<td>1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Total training time in hours: absolute time and relative time versus VAE.

<table>
<thead>
<tr>
<th>Model</th>
<th>Forward↓</th>
<th>Reverse↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE</td>
<td>18494</td>
<td>10149</td>
</tr>
<tr>
<td>Cyc-VAE</td>
<td>3390</td>
<td>5587</td>
</tr>
<tr>
<td>AE</td>
<td>672</td>
<td>2589</td>
</tr>
<tr>
<td>β(0)-VAE</td>
<td>625</td>
<td>1897</td>
</tr>
<tr>
<td>β(0.5)-VAE</td>
<td>939</td>
<td>4078</td>
</tr>
<tr>
<td>SA-VAE</td>
<td>341</td>
<td>10651</td>
</tr>
<tr>
<td>iVAE</td>
<td>116</td>
<td>1520</td>
</tr>
<tr>
<td>iVAEMI</td>
<td>134</td>
<td>1137</td>
</tr>
</tbody>
</table>

Table 5: Forward and reverse PPL on PTB.

| t =0 | in new york the company declined comment |
| t =0.1 | in new york the company declined comment |
| t =0.2 | in new york the transaction was suspended |
| t =0.3 | in the securities company said yesterday |
| t =0.4 | in other board the transaction had disclosed |
| t =0.5 | other of those has been available |
| t =0.6 | both of companies have been unchanged |
| t =0.7 | both men have received a plan to restructure |
| t =0.8 | and to reduce that it owns |
| t =0.9 | and to continue to make prices |
| t =1  | and they plan to buy more today |

Table 4: Interpolating latent representation.

Latent space interpolation. One favorable property of VAEs (Bowman et al., 2015; Zhao et al., 2018) is to provide smooth latent representations that capture sentence semantics. We demonstrate this by interpolating two latent features, each of which represents a unique sentence. Table 4 shows the generated examples. We take two sentences $x_1$ and $x_2$, and obtain their latent features as the sample-averaging results for $z_1$ and $z_2$, respectively, from the implicit encoder, and then greedily decode conditioned on the interpolated feature $z_t = z_1 \cdot (1 - t) + z_2 \cdot t$ with $t$ increased from 0 to 1 by a step size of 0.1. It generates sentences with smooth semantic evolution.

Improved Decoder. One might wonder whether the improvements come from simply having a more flexible encoder during evaluation, rather than from utilizing high-quality latent features, and learning a better decoder. We use PTB to confirm our findings. We draw samples from the prior $p(z)$, and greedily decode them using the trained decoder. The quality of the generated text is evaluated by an external library “KenLM Language Model Toolkit” (Heafield et al., 2013) with two metrics (Kim et al., 2017): (1) Forward PPL: the fluency of the generated text based on language models derived from the PTB training corpus; (2) Reverse PPL: the fluency of PTB corpus based on language model derived from the generated text, which measures the extent to which the generations are representative of the PTB underlying language model. For both the PPL numbers, the lower the better. We use $n=5$ for n-gram language models in “KenLM”.

As shown in Table 5, implicit LVMs outperform others in both PPLs, which confirms that the implicit representation can lead to better decoders. The vanilla VAE model performs the worst. This is expected, as the posterior collapse issue results in poor utilization of a latent space. Besides, we can see that iVAEMI generates comparably fluent but more diverse text than pure iVAE, from the lower reverse PPL values. This is reasonable, due to the ability of iVAEMI to encourage diverse latent samples per sentence with the aggregated regularization in the latent space.

5.2 Unaligned style transfer

We next consider the task of unaligned style transfer, which represents a scenario to generate text with desired specifications. The goal is to control one style aspect of a sentence that is independent of its content. We consider non-parallel corpora of sentences, where the sentences in two corpora have the same content distribution but with different styles, and no paired sentence is provided.
**Model Extension.** The success of this task depends on the exploitation of the distributional equivalence of content to learn sentiment independent content code and decoding it to a different style. To ensure such independence, we extend iVAEMI by adding a sentiment classifier loss to its objective (14), similar to the previous style transfer methods (Shen et al., 2017; Kim et al., 2017). Let $y$ be the style attribute, $x_p$ and $x_n$ (with corresponding features $z_p$, $z_n$) be sentences with positive and negative sentiments, respectively. The style classifier loss $L_{\text{class}}(z_p, z_n)$ is the cross entropy loss of a binary classifier.

The classifier and encoder are trained adversarially: (1) the classifier is trained to distinguish latent features with different sentiments; (2) the encoder is trained to fool the classifier in order to remove distinctions of content features from sentences. In practice, the classifier is implemented as an MLP. We implement two separate decoder LSTMs for clean sentiment decoding: one for positive $p(x|z, y = 1)$, and one for negative sentiment $p(x|z, y = 0)$. The prior $p(z)$ is also implemented as an implicit distribution, via transforming noise from a standard Gaussian through a MLP. Appendix C.2.2 lists more details.

**Baseline.** We compare iVAEMI with a state-of-the-art unaligned sentiment transferer: the adversarially regularized autoencoder (ARAE) (Kim et al., 2017), which directly regularizes the latent space with Wasserstein distance measure.

**Datasets.** Following (Shen et al., 2017), the Yelp restaurant reviews dataset is processed from the original Yelp dataset in language modeling. Reviews with user rating above three are considered positive, and those below three are considered negative. The pre-processing allows sentiment analysis on sentence level with feasible sentiment, ending up with shorter sentences with each at most 15 words than those in language modeling. Finally, we get two sets of unaligned reviews: 250K negative sentences, and 350K positive ones. Other dataset details are shown in Appendix C.2.1.

**Evaluation Metrics.** (1) Acc: the accuracy of transferring sentences into another sentiment measured by an automatic classifier: the “fasttext” library (Joulin et al., 2017); (2) BLEU: the consistency between the transferred text and the original; (3) PPL: the reconstruction perplexity of original sentences without altering sentiment; (4) RPPL: the reverse perplexity that evaluates the training corpus based on language model derived from the generated text, which measures the extent to which the generations are representative of the training corpus; (5) Flu: human evaluated index on the fluency of transferred sentences when read alone (1-5, 5 being most fluent as natural language); (6) Sim: the human evaluated similarity between the original and the transferred sentences in terms of their contents (1-5, 5 being most similar). Note that the similarity measure doesn’t care sentiment but only the topic covered by sentences. For human evaluation, we show 1000 randomly selected pairs of original and transferred sentences to crowdsourcing readers, and ask them to evaluate the "Flu" and "Sim" metrics stated above. Each measure is averaged among crowdsourcing readers.

As shown in Table 7, iVAEMI outperforms ARAE in metrics except Acc, showing that iVAEMI captures informative representations, generates consistently opposite sentences with similar grammatical structure and reserved semantic meaning. Both methods perform successful sentiment transfer as shown by Acc values. iVAEMI achieves a little lower Acc due to much more content retaining, even word retaining, of the source sentences.

### Table 6: Sentiment transfer on Yelp. (Up: From negative to positive, Down: From positive to negative.)

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc↑</th>
<th>BLEU↑</th>
<th>PPL↓</th>
<th>RPPL↓</th>
<th>Flu↑</th>
<th>Sim↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARAE</td>
<td>95</td>
<td>32.5</td>
<td>6.8</td>
<td>395</td>
<td>3.6</td>
<td>3.5</td>
</tr>
<tr>
<td>iVAEMI</td>
<td>92</td>
<td>36.7</td>
<td>6.2</td>
<td>285</td>
<td>3.8</td>
<td>3.9</td>
</tr>
</tbody>
</table>

Table 7: Sentiment Transfer on Yelp.
We consider the open-domain dialog response generation task, where we need to generate a natural language response given a dialog history. It is crucial to learn a meaningful latent feature representation of the dialog history in order to generate a consistent, relevant, and contentful response that is likely to drive the conversation (Gao et al., 2019).

Datasets. We consider two mainstream datasets in recent studies (Zhao et al., 2017b, 2018; Fu et al., 2019; Gu et al., 2018): Switchboard (Godfrey and Holliman, 1997) and Dailydialog (Li et al., 2017c). Switchboard contains 2,400 two-way telephone conversations under 70 specified topics. Dailydialog has 13,118 daily conversations for an English learner. We process each utterance as the response of previous 10 context utterances for each speaker. The datasets are separated into training, validation, and test sets as convention: 2316:60:62 for Switchboard and 10:1:1 for Dailydialog, respectively.

Model Extension. We adapt iVAEMI by integrating the context embedding $c$ into all model components. The prior $p(z|c)$ is defined as an implicit mapping between context embedding $c$ and prior samples, which is not pre-fixed but learned together with the variational posterior for more modeling flexibility. The encoder $q(z|x, c)$, auxiliary dual function $\nu_{q}(z, c)$ and decoder $p(x|c, z)$ depend on context embedding $c$ as well. Both encoder and decoder are implemented as GRUs. The utterance encoder is a bidirectional GRU with 300 hidden units in each direction. The context encoder and decoder are both GRUs with 300 hidden units. Appendix C.3.1 presents more training details.

Baseline. We benchmark representative baselines and state-of-the-art approaches, include: SeqGAN, a GAN based model for sequence generation (Li et al., 2017b); CVAE baseline (Zhao et al., 2017b); dialogue WAE, a conditional Wasserstein auto-encoder for response generation (Gu et al., 2018).

Table 6 presents some examples. In each box, we show the source sentence, the transferred sentence by ARAE and iVAEMI, respectively. We observe that ARAE usually generates new sentences that miss the content of the source, while iVAEMI shows better content-preserving.

5.3 Dialog response generation

We consider the open-domain dialog response generation task, where we need to generate a natural language response given a dialog history. It is crucial to learn a meaningful latent feature representation of the dialog history in order to generate a consistent, relevant, and contentful response that is likely to drive the conversation (Gao et al., 2019).

Datasets. We consider two mainstream datasets in recent studies (Zhao et al., 2017b, 2018; Fu et al., 2019; Gu et al., 2018): Switchboard (Godfrey and Holliman, 1997) and Dailydialog (Li et al., 2017c). Switchboard contains 2,400 two-way telephone conversations under 70 specified topics. Dailydialog has 13,118 daily conversations for an English learner. We process each utterance as the response of previous 10 context utterances for each speaker. The datasets are separated into training, validation, and test sets as convention: 2316:60:62 for Switchboard and 10:1:1 for Dailydialog, respectively.

Model Extension. We adapt iVAEMI by integrating the context embedding $c$ into all model components. The prior $p(z|c)$ is defined as an implicit mapping between context embedding $c$ and prior samples, which is not pre-fixed but learned together with the variational posterior for more modeling flexibility. The encoder $q(z|x, c)$, auxiliary dual function $\nu_{q}(z, c)$ and decoder $p(x|c, z)$ depend on context embedding $c$ as well. Both encoder and decoder are implemented as GRUs. The utterance encoder is a bidirectional GRU with 300 hidden units in each direction. The context encoder and decoder are both GRUs with 300 hidden units. Appendix C.3.1 presents more training details.

Baseline. We benchmark representative baselines and state-of-the-art approaches, include: SeqGAN, a GAN based model for sequence generation (Li et al., 2017b); CVAE baseline (Zhao et al., 2017b); dialogue WAE, a conditional Wasserstein auto-encoder for response generation (Gu et al., 2018).

Table 6: Dialog response generation on two datasets.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>SeqGAN</th>
<th>CVAE</th>
<th>WAE</th>
<th>iVAEMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU-R↑</td>
<td>0.282</td>
<td>0.295</td>
<td>0.394</td>
<td>0.427</td>
</tr>
<tr>
<td>BLEU-P↑</td>
<td>0.282</td>
<td>0.258</td>
<td>0.254</td>
<td>0.254</td>
</tr>
<tr>
<td>BLEU-F1↑</td>
<td>0.282</td>
<td>0.275</td>
<td>0.309</td>
<td>0.319</td>
</tr>
<tr>
<td>BOW-A↑</td>
<td>0.817</td>
<td>0.856</td>
<td>0.897</td>
<td>0.930</td>
</tr>
<tr>
<td>BOW-E↑</td>
<td>0.515</td>
<td>0.562</td>
<td>0.627</td>
<td>0.670</td>
</tr>
<tr>
<td>BOW-G↑</td>
<td>0.748</td>
<td>0.846</td>
<td>0.887</td>
<td>0.900</td>
</tr>
<tr>
<td>Intra-dist1↑</td>
<td>0.705</td>
<td>0.803</td>
<td>0.713</td>
<td>0.828</td>
</tr>
<tr>
<td>Intra-dist2↑</td>
<td>0.521</td>
<td>0.415</td>
<td>0.651</td>
<td>0.692</td>
</tr>
<tr>
<td>Inter-dist1↑</td>
<td>0.070</td>
<td>0.112</td>
<td>0.245</td>
<td>0.391</td>
</tr>
<tr>
<td>Inter-dist2↑</td>
<td>0.052</td>
<td>0.102</td>
<td>0.413</td>
<td>0.668</td>
</tr>
</tbody>
</table>

Table 8: Dialog response generation on two datasets.

Evaluation Metrics. We adopt several widely used numerical metrics to systematically evaluate the response generation, including BLEU score (Papineni et al., 2002), BOW Embedding (Liu et al., 2016) and Distinct (Li et al., 2015), as used in (Gu et al., 2018). For each testing context, we sample 10 responses from each model.

- **BLEU** measures how much a generated response contains n-gram overlaps with the references. We compute 4-gram BLEU. For each test context, we sample 10 responses from the models and compute their BLEU scores. BLEU precision and recall is defined as the average and maximum scores, respectively (Zhao et al., 2017b).
- **BOW embedding** is the cosine similarity of bag-of-words embeddings between the generated responses and the reference. We compute three different BOW embedding: greedy, average, and extreme.
- **Distinct** evaluates the diversity of the generated responses: dist-n is defined as the ratio of unique n-grams (n=1,2) over all n-grams in the generated responses. We evaluate diversities for both within each sampled response and among all responses as intra-dist and inter-dist, respectively.
Tables 8 show the performance comparison. iVAEMI achieves consistent improvement on a majority of the metrics. Especially, the BOW embeddings and Distinct get significantly improvement, which implies that iVAEMI produces both meaningful and diverse latent representations.

6 Conclusion

We present two types of implicit deep latent variable models, iVAE and iVAEMI. Core to these two models is the sample-based representation of the latent features in LVM, in replacement of traditional Gaussian-based distributions. Extensive experiments show that the proposed implicit LVM models consistently outperform the vanilla VAEs on three tasks, including language modeling, style transfer and dialog response generation.

Acknowledgments

We acknowledge Yufan Zhou for helpful discussions and the reviewers for their comments.

References


Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. 2017. Adversarial feature learning. ICLR.


