Fixing Gaussian Mixture VAEs for Interpretable Text Generation

Wenxian Shi∗
Peking University
wxsh@pku.edu.cn

Hao Zhou
ByteDance AI Lab
zhouhao.nlp@bytedance.com

Ning Miao
Peking University
miaoning@pku.edu.cn

Shenjian Zhao
ByteDance AI Lab
zhaoshenjian.01@bytedance.com

Lei Li
ByteDance AI Lab
lileilab@bytedance.com

Abstract

Variational auto-encoder (VAE) with Gaussian priors is effective in text generation. To improve the controllability and interpretability, we propose to use Gaussian mixture distribution as the prior for VAE (GMVAE), since it includes an extra discrete latent variable in addition to the continuous one. Unfortunately, training GMVAE using standard variational approximation often leads to the mode-collapse problem. We theoretically analyze the root cause — maximizing the evidence lower bound of GMVAE implicitly aggregates the means of multiple Gaussian priors. We propose Dispersed-GMVAE (DGMVAE), an improved model for text generation. It introduces two extra terms to alleviate mode-collapse and to induce a better structured latent space. Experimental results show that DGMVAE outperforms strong baselines in several language modeling and text generation benchmarks.

1 Introduction

Variational auto-encoders [Kingma and Welling, 2013; Rezende et al., 2014] VAEs) have been widely adopted in natural language generation [Bowman et al., 2016]. VAE employs a global latent variable to represent semantics, leading to diverse and coherent generated sentences. Vanilla VAE adopts a continuous latent variable following a multivariate Gaussian distribution with a diagonal covariance matrix. Recently, [Zhao et al., 2018b] propose to replace the continuous latent variable of VAE with a discrete one for better interpretation in generating dialog. The discrete latent variable could represent the dialog actions in their system, which gives promising results even in an unsupervised setting.

However, we argue that VAE only with a discrete latent variable is not sufficient for interpretable language generation. Compared with the continuous latent variable, the discrete one suffers from its relatively low model capacity. Because the discrete latent space only includes limited size of points, it is unable to convey as much information as the continuous latent space (infinite points).

In this paper, we propose to generate text using Gaussian mixture VAE (GMVAE). GMVAE has been effective in image modeling [Dilokthanakul et al., 2016; Jiang et al., 2017]. It enjoys the benefits of both discrete and continuous latent space where the discrete variable is easy to control. This is superior to semi-supervised VAEs [Kingma et al., 2014; Hu et al., 2017; Zhou and Neubig, 2017], containing two independent discrete and continuous latent variables.

However, vanilla GMVAE suffers from the mode-collapse problem in language generation, where the multiple Gaussian priors tend to concentrate during training and eventually degenerate into a

∗Work done while Wenxian Shi was a research intern in ByteDance AI Lab.
single Gaussian (Fig. 1). Essentially, each Gaussian component in latent space tends to have close means. GMVAE fails to capture the multi-modes structure of the data and cannot effectively utilize the discrete latent variable. For example, as illustrated in Fig. 1a, utterances requesting the weather and requesting an appointment will be mapped into the same mode due to the mode-collapse. The mode-collapse problem has also been observed in the image modeling task (Dilokthanakul et al., 2016) using GMVAE. However, the problem is more severe in the scenario of language generation.

In this paper, we propose the Dispersed-GMVAE (DGMVAE), which fixes the mode-collapse problem in GMVAE. First, we theoretically analyze that the mode-collapse problem is intrinsically caused by the evidence lower bound (ELBO) of GMVAE. Maximizing the ELBO of GMVAE implicitly aggregates the mean of the Gaussian mixture priors. We introduce two extra terms in our proposed DGMVAE to alleviate mode-collapse and to obtain a better structured latent space (Fig. 1b). Experimental results show that DGMVAE can avoid the mode-collapse problem effectively. Furthermore, DGMVAE achieves significantly better results in language modeling on Penn Treebanks (Marcus et al., 1993, PTB) and in interpretable text generation over two dialog benchmarks.

Our contributions include: a) we proposed to use GMVAE for interpretable text generation; b) we theoretically analyze the mode-collapse problem in GMVAE and address it by proposing DGMVAE; c) we empirically studied the performance of DGMVAE and show it obtains good results on various generation tasks.

2 Related Work

VAEs for Language Generation. Variational auto-encoders are proposed by Kingma and Welling (2013, VAEs) and Rezende et al. (2014), and applied by Bowman et al. (2016) for natural language generation. VAEs are extended by many following works in various specific language generation tasks, such as dialog generation (Serban et al., 2017; Wen et al., 2017; Zhao et al., 2017b, 2018b), summarization (Li et al., 2017a) and other natural language generation tasks (Miao et al., 2016; Zhang et al., 2016; Semeniuta et al., 2017; Gupta et al., 2018; Xu and Durrett, 2018).

Additionally, Wen et al. (2017) and Zhao et al. (2018b) propose to replace the continuous latent variable with a discrete one for interpretable sentence generation. Kingma et al. (2014) propose the semi-VAE for semi-supervised learning. This model is then adopted by Hu et al. (2017); Zhou and Neubig (2017) for style-transfer and labeled sequence transduction, respectively. Different from GMVAE, continuous and discrete latent variables in semi-VAE are independent.

Gaussian Mixture VAEs. Using Gaussian mixture models as priors in VAEs is not new. Gaussian mixture variational auto-encoder has been used in the unsupervised clustering (Dilokthanakul et al., 2016; Jiang et al., 2017), obtaining promising results. Wang et al. (2019) used GMM as priors for topic-guided text generation. GMVAE used in this paper is similar to Jiang et al. (2017). However, we apply GMVAE for interpretable text generation and propose the DGMVAE to address the mode-collapse problem according to our theoretical analysis.
KL Collapse vs. Mode Collapse. The vanilla VAE models usually suffer from the KL collapse problem in language generation, in which the KL regularization term will quickly collapse to 0. A line of following work (Bowman et al., 2016; Zhao et al., 2016b; 2018b; Higgins et al., 2017) is proposed to avoid the KL collapse problem. More specifically, mode collapse is related to mixture models, in which multiple modes vanish and collapse into a single mode. Mode collapse is also caused by the KL term, but the essential cause is different. In this paper, we focus on addressing the mode-collapse problem.

3 Proposed Approach

Probabilistic graphical models of VAE and its variants are shown in Fig. 2. Vanilla VAE (Kingma and Welling, 2013) only includes a continuous latent variable; discrete VAE models such as DI-VAE (Zhao et al., 2018b) adopt a discrete latent variable for interpretability; semi-VAE (Kingma et al., 2014) employ independent discrete and continuous latent variables; Gaussian mixture VAE (GM-VAE) (Dilokthanakul et al., 2016; Jiang et al., 2017) use dependent discrete and continuous latent variables for better interpretable generation performance.

In the following, we will first describe the vanilla GMVAE in Sec. 3.1. We will give a theoretical analysis of the mode-collapse issue in GMVAE. Based on the theoretical insights, we propose DGMVAE to fix the issue.

3.1 Gaussian Mixture VAE

GMVAE is a probabilistic generative model that adopts the Gaussian mixture models (Bishop, 2006) as its prior. GMVAE employs a discrete latent variable $c$ and a continuous latent variable $z$, with $z$ dependent on $c$. In this model, the marginal likelihood of a sentence $x$ is:

$$p(x) = \int \sum_c p(z, c)p_\theta(x|z)dz,$$

where $\theta$ is the parameters of generation model which generates $x$ from $z$, $p(z, c)$ is the Gaussian mixture prior distribution and can be computed by $p(z, c) = p(c)p(z|c)$. Intuitively, $c$ represents the components of mixture Gaussian and $p(c)$ could be assumed as an uniform distribution; while $p(z|c)$ is a multivariate Gaussian distribution of the corresponding component.

Testing. During testing, a mixture Gaussian component $c$ is first chosen according to the prior distribution $p(c)$. Then the continuous variable $z$ is sampled from the chosen Gaussian prior $p(z|c)$. As in Bowman et al. (2016), a generation network takes $z$ as input and generate the sentence $x$ through a decoder $p_\theta(x|z)$.

Training. Optimizing and inference for Eq. 1 is difficult. Following previous work of Kingma and Welling (2013) and Rezende et al. (2014), we use a variational posterior distribution $q_\phi(z, c|x)$ with parameters $\phi$ to approximate the real posterior distribution $p(z, c|x)$. With the mean field approximation (King et al., 2003), $q_\phi(z, c|x)$ can be factorized as: $q_\phi(z, c|x) = q_\phi(z|x)q_\phi(c|x)$.

The posterior $q_\phi(z|x)$ is assumed as a multivariate Gaussian distribution, whose mean $\mu_\phi(x)$ and variance $\sigma^2_\phi(x)$ are obtained through a neural network (recognition network). $q_\phi(c|x)$ is calculated according to:

$$q_\phi(c = i|x) = \frac{q_\phi(x|c = i)q(c = i)}{\sum_j q_\phi(x|c = j)q(c = j)},$$

in which $q_\phi(x|c = i)$ is the probability of generating the mapped vector of $x$ in latent space by $i$th Gaussian component, and $q(c)$ could be taken as prior $p(c)$. In practice, $\mu_\phi(x)$ is taken as the deterministic mapping of $x$ in the latent space, which does not damage the conditional independence in mean field approximation.

Instead of optimizing the marginal likelihood in Eq. 1, we maximize a evidence lower bound (ELBO). The ELBO can be decomposed as the summation of a reconstruction term and regularization terms.
We abbreviate \( \mu \) ELBO

We further investigate the \( z \)

According to the theoretical insights in Sec. 3.2, we propose to include two extra terms in our objective function to analyze mode-collapse. To this end, we present two theorems, which indicate that the regularization terms of GMV AE’s \( \mathcal{R}_c \) and \( \mathcal{R}_z \), are responsible for the mode collapse problem. We only give explanations and remarks for each theorem, with the details included in the supplementary materials.

We abbreviate \( \mu \phi (c) \), the posterior mean of \( z \) given \( x \), as \( \mu \phi \), and \{\( \mu \_i \}\}_{i=1,2,...,K}, the set of means of Gaussian components, as \( M \). The trace of variance matrix of mean in \( M \) is denoted as \( \text{Var}_M \).

**Theorem 1.** Maximizing the \( \mathcal{R}_c \) pushes a close upper bound of \( \text{Var}_M \), \( S\mu\phi \), to decrease. Here \( S\mu\phi = \sum_k (\mu\phi - \mu_k)^T (\mu\phi - \mu_k) \) is the squared sum of distance between \( \mu_k \) and \( \mu\phi \).

By performing some algebraic operations, we find that the inner product of \( \left[ \frac{\partial \text{Var}_M}{\partial \mu_i} \right]_{i=1,2,...,K} \) and \( \left[ \frac{\partial S\mu\phi}{\partial \mu_i} \right]_{i=1,2,...,K} \) is always non-negative, which means the directions of their gradients are opposite. So, performing gradient ascent on ELBO will make \( S\mu\phi \) smaller. As a result, \( \text{Var}_M \) is limited by the decreasing bound \( \frac{S\mu\phi}{K} \).

We abbreviate \( \text{Var}_{q_\phi(c|x)} \mu_c \) as the trace of variance matrix of mean under the distribution of \( q_\phi(c|x) \), \( \text{Var}_{q_\phi(c|x)} \mu_c = \text{E}_{q_\phi(c|x)} |\mu_c|^2 - \text{E}_{q_\phi(c|x)} |\mu_c| \), assuming the standard deviation of all Gaussian components equal, i.e., \( \sigma_1 = \sigma_2 = ... = \sigma_K = \sigma \).

**Theorem 2.** \( \mathcal{R}_z \) contains a negative regularization term of \( \text{Var}_{q_\phi(c|x)} \mu_c \).

\( \mathcal{R}_z \) could be re-written as

\[
\mathbb{E}_{q_\phi(c|x)} \sum_c \mu_c \phi(c|x) \log \frac{p(z|c)}{q_\phi(z|x)} = -\text{KL}(q_\phi(z|x)||\hat{p}(z|x))) - \frac{1}{2\sigma^2} \text{Var}_{q_\phi(c|x)} \mu_c,
\]

where \( \hat{p}(z|x) = \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2\sigma^2} (z - \mathbb{E}_{q_\phi(c|x)} \mu_c)^2 \right) \) is a multivariate Gaussian distribution, whose mean is the expectation of the mean of priors under distribution \( q_\phi(c|x) \).

Hence, maximizing \( \mathcal{R}_z \) implicitly minimize \( \text{Var}_{q_\phi(c|x)} \mu_c \), which may also lead to mode collapse in GMVAE.

### 3.3 Dispersed-GMVAE

In this section, we propose the Dispersed-GMVAE (DGMVAE), which is a simple yet effective way to avoid the mode-collapse problem.

According to the theoretical insights in Sec. 3.2, we propose to include two extra terms in our objective to balance the mode collapse from ELBO. We propose a new objective \( \mathcal{L}(\theta) \) for \( x \) sampled from the dataset \( D \):

\[
\mathcal{L}(\theta) = \mathbb{E}_x \text{ELBO} + \mathcal{L}_{\text{mi}} + \beta \cdot \mathcal{L}_{\text{var}},
\]

\[
\mathcal{L}_{\text{mi}} = \mathbb{E}_{q_\phi(c)} \log q_\phi(c) - \mathbb{E}_x \mathbb{E}_{q_\phi(c|x)} \log q_\phi(c|x), \quad \mathcal{L}_{\text{var}} = \frac{1}{2\sigma^2} \mathbb{E}_x \text{Var}_{q_\phi(c|x)} \mu_c,
\]

(4)
where \( q_\phi(c) \) is the posterior marginal distribution estimated by \( E_x q_\phi(c|x) \). Empirically, it is obtained by averaging \( q_\phi(c|x) \) within the mini-batch.

DGMVAE add an extra variance regularization term \( L_{\text{var}} \) and a mutual information term \( L_{\text{mi}} \) in its objective. Intuitively, they serve different roles. \( L_{\text{var}} \) with a hyper-parameter \( \beta \) is proposed to regularize the concentration trends of Gaussian mixture components. We can tune \( \beta \) to make a trade-off between variance and concentration degree.

We also include a mutual information term in Eq. 4:

\[
L_{\text{mi}} = \mathcal{H}(c) - \mathcal{H}(c|x).
\]

As introduced by previous works (Chen et al., 2016; Zhao et al., 2017a, 2018b), \( L_{\text{mi}} \) could enhance the interpretability and alleviate the KL-collapse. Our ablation study in the experiments shows that \( L_{\text{var}} \) and \( L_{\text{mi}} \) are both necessary to obtain good empirical performances of DGMVAE.

The final objective of DGMVAE could be:

\[
E_x E_{q_\phi(z|x)} \log p_\theta(x|z) - KL(q_\phi(c)||p(c)) - E_x [KL(q_\phi(z|x)||\hat{p}(z|x))] - \beta' E_x \text{Var}_{q_\phi(c|x)} \mu_c.
\]

in which \( \beta' = \frac{1-\beta}{2\sigma_z^2} \). More details on how to obtain the final objective function from Eq. 3 and Eq. 4 could be found in supplementary materials.

Except for the learning objective, DGMVAE has similar architecture as GMVAE. It consists of an RNN encoder for learning posterior and an RNN decoder for generation.

**Encoder.** Recurrent neural networks such as GRU (Chung et al., 2014) as recognition networks encode sentences into compact hidden states. The mean \( \mu_c \) and variance \( \sigma_z^2 \) of the posterior distribution \( q_\phi(z|x) \) (assumed as a multivariate diagonal Gaussian) are obtained from the last hidden states through two affine transformations.

**Decoder.** In the decoding phase, we first sample a \( z \) from the GMM priors (in testing) or from posterior (in training) by the reparameterization trick (Kingma and Welling, 2013). The sentences will be generated on a recurrent neural language model fashion (generation networks), with the \( z \) as the initialized hidden state. We use multiple independent discrete latent variables following Zhao et al. (2018b).

**Interpretable Dialog Generation.** We follow the same approach of DI-VAE (Zhao et al., 2018b) for interpretable dialog generation. The approach could be extended to other scenarios of interpretable generations, but we only validate our DGMVAE on dialog for comparing with Zhao et al. (2018b).

Specifically, in dialog generation, we generate response \( r \) given the dialog context \( y \). A DGMVAE model is pre-trained in all utterances of the training set to capture the interpretable facts (discrete latent variable \( c \)) such as dialog actions or intentions. In training, a hierarchical recurrent encoder-decoder model (HRED) with attention (Sordoni et al., 2015; Serban et al., 2016) \( p_\theta(r|z, y) \) is trained to generate the response. Here \( z \) is obtained from the pre-trained recognition network \( q_\phi(z|r) \) of DGMVAE and then fed into the decoder. A policy network \( p_\pi(c|r) \) is trained jointly to predict \( c \) sampled from \( q_\phi(c|r) \) in order to predict \( c \) in the testing stage.

### 4 Experiments

In this section, we empirically test the generation quality and interpretable ability of our proposed model on standard benchmarks, compared with a line of baselines.

#### 4.1 Setup

We conduct experiments following Zhao et al. (2018b). For generation quality, we use the Penn Treebanks (Marcus et al., 1993, PTB) pre-processed by Mikolov (Mikolov et al., 2010) as the benchmark. For interpretability, we use the Daily Dialogs (Li et al., 2017b; DD) and the Stanford Multi-Domain Dialog (Eric et al., 2017; SMD) datasets. DD is a chat-oriented dataset containing 13,118 multi-turn dialogs, annotated with dialog actions and emotions. SMD contains 3,031 human-Woz, task-oriented dialogs collected from 3 different domains (navigation, weather and scheduling).

We compare our model with the following baselines: 1) RNNLM, language model (Mikolov et al., 2010) implemented by LSTM (Hochreiter and Schmidhuber, 1997) and 2) AE, auto-encoders (Vincent 2010; 2)

2 https://github.com/tensorflow/models/blob/master/tutorials/rnn/ptb/ptb_word_lm.py
et al., 2010) without latent space regularization; 3) DAE, auto-encoders with discrete latent space; 4) VAE, the vanilla VAE [Kingma and Welling, 2013] with only continuous latent variable and normal distribution prior; 5) DVAE, VAE with discrete latent variables; 6) DI-VAE, a DVAE variant (Zhao et al., 2018b) with an extra mutual information term; 7) semi-VAE, semi-supervised VAE model proposed by Kingma et al. (2014) with independent discrete and continuous latent variables; 8) GMVAE, vanilla GMVAE models as introduced in 3.1. Results of these baselines are obtained by our implementation except DI-VAE. Gumbel-softmax (Jang et al., 2016) is used for reparameterization in VAE variants with discrete latent variable.

The encoder and decoder in all models are implemented with single-layer GRU (Chung et al., 2014), with the hidden size as 512. The dimension of discrete latent variables is set to 10 for PTB and 5 for DD and SMD, while the number of discrete latent variables is set to 20, 3 and 3. The dimension of continuous latent space is 100 for PTB, 15 for DD and 48 for SMD. $\beta$ is set to 0.9 for DGMVAE. KL annealing with logistic weight function is adopted for all VAE variants. All hyper-parameters including $\beta$ are chosen according to the objective (language generation task) or BLEU scores (dialog generation task) in the validation set. Details of hyper-parameters are included in the supplementary.

4.2 Effects of DGMVAE on Mode-Collapse

We illustrate the effectiveness of DGMVAE to alleviate the mode-collapse problem. Fig. 3 gives a visualization. We train GMVAE and DGMVAE in utterances on the DD dataset, and randomly sample 300 points from test data at 2,000 and 10,000 training steps, respectively. The dimension of latent space is set to 2 for visualization. As in Fig. 3, the mean and variance of GMM priors are indicated by grey points and circles, respectively. The means of posteriors are marked as colored points (points with different discrete latent variables are associated with different colors).

It can be seen that, after 10,000 training steps, the vanilla GMVAE degenerates into uni-Gaussian VAE, with the same mean values of all Gaussian components (Fig. 3a and 3b). DGMVAE gives quite promising results as shown in Fig. 3g and 3h, in which different components of the GMM are dispersed and cluster data points into multiple modes well. In order to verify the effects of the additional two terms of DGMVAE, we incorporate $L_{mi}$ and $L_{var}$ incrementally. GMVAE + $L_{mi}$ indeed helps alleviate the mode-collapse, however, the posterior points are quite concentrated to the priors. This indicates the latent space in such case is not smooth enough. GMVAE + $L_{var}$ can also avoid the mode-collapse problem, but it does not cluster points with the same discrete labels together well.

---

We get results of DI-VAE with their released codes. Our reported results differs as in their paper because we find they perform tokenization twice in their codes.
With the presence of both continuous and discrete latent variables, DGMVAE AE enjoys its higher model capacity and gives the best reconstruction performance (BLEU, PPL), superior to other VAE variants. Although semi-VAE also includes discrete and continuous latent variables, it fails to make use of both of them because of the independent hypothesis. As shown in Tab. 1 both the two terms improve the performance. The DGMVAE with both terms achieves the best results.

Besides the reconstruction, we also find that DGMVAE significantly outperforms related work in generating high-quality sentences. rPPL is a powerful metric for measuring the fluency and diversity; DGMVAE obtains the lowest rPPL, which is significantly better than all other models. The lowest wKL also shows that word distribution in DGMVAE generations is most consistent with the training word frequencies calculated in generated data and training data shows the quality of generation. Perplexity of samples generated from posterior in test set measures the fluency of reconstruction. These metrics are evaluated on the test set of PTB, except rPPL and wKL, which are calculated on sentences generated by sampling from these models’ prior distribution (sampling a random vector for AE).

Besides, the values of the regularization terms are also included in order to give some indications of the mode-collapse and KL collapse. We list the KL divergence of continuous latent variables (KL(c)) and discrete latent variables (KL(d)), which are averaged by the number of discrete variables. The variance of GMM mean (VM) and mutual information (MI) terms are shown as well.

We first present the ablation study to show whether the two extra terms (Sec. 3.3) both contribute to the performance gains. As shown in Tab. 1 VM term helps to alleviate the mode-collapse, according to the higher variance of mean. MI term helps to increase the information encoded by discrete latent variables, according to higher mutual information. As shown in the last three rows of Tab. 1 both the two terms improve the performance. The DGMVAE with both terms achieves the best results.

With the presence of both continuous and discrete latent variables, DGMVAE enjoys its higher model capacity and gives the best reconstruction performance (BLEU, PPL), superior to other VAE variants. Although semi-VAE also includes discrete and continuous latent variables, it fails to make use of both of them because of the independent hypothesis. As shown in Tab. 1 either discrete or continuous latent variable collapses in semi-VAEs. AE could reproduce input sentences well, but it fails to generate diverse sentences.

Besides the reconstruction, we also find that DGMVAE significantly outperforms related work in generating high-quality sentences. rPPL is a powerful metric for measuring the fluency and diversity; DGMVAE obtains the lowest rPPL, which is significantly better than all other models. The lowest wKL also shows that word distribution in DGMVAE generations is most consistent with the training set.

### Table 1: Language generation results on PTB. $\beta = 0.9$ for DGMVAE. The larger (or lower*), the better.

| Model                  | Evaluation Results | Regularization Terms |
|------------------------|--------------------|-----------------------|
|                        | rPPL↑ | BLEU↑ | wKL↑ | PPL↑ | KL(z) | KL(d) | VM | MI |
| Test Set               |       |       |      |      |       |       |    |    |
| RNNLM [Mikolov et al. [2010]] |       |       |      |      |       |       |    |    |
| AE [Vincent et al. [2010]] | 730.81 | 16.88 | 0.58 | 31.90 |       |       |    |    |
| VAE [Kingma and Welling [2013]] | 922.71 | 3.73  | 0.76 | 91.95 | 6.62  |       |    |    |
| DVAE                   | 997.71 | 3.93  | 0.58 | 88.55 |       |       |    |    |
| DL-VAE [Zhang et al. [2018]] | 453.53 | 3.61  | 0.58 | 100.56 | 1.74  | 1.22  |    |    |
| semi-VAE [Kingma et al. [2014]] | 574.42 | 4.19  | 0.69 | 93.72 | 0.13  | 1.26  |    |    |
| GMVAE                  | 712.34 | 4.87  | 0.73 | 92.95 | 0.49  | 0.14  | 1.34 |
| DGMVAE − $L_{mi}$     | 925.66 | 4.17  | 0.80 | 90.26 | 7.13  | 0.02  | 0.38 | 0.016 |
| DGMVAE − $L_{mi}$     | 331.80 | 6.34  | 0.45 | 61.77 | 13.03 | 0.10  | 9.93 | 1.30  |
| DGMVAE − $L_{mi}$     | 560.56 | 5.64  | 0.62 | 71.12 | 3.87  | 0.31  | 24.84 | 0.28  |
| DGMVAE                 | 244.30 | 8.45  | 0.35 | 49.60 | 6.41  | 0.10  | 21.42 | 1.19  |

*Sample size here is 40,000, the same as PTB training set.

*For models with VM term, it is the KL divergence between posterior $q_\phi(z|x)$ and the expected prior $\hat{p}(z|x)$.

*For models with MI term, it is the KL divergence between marginal posterior $q_\phi(c)$ and prior $p(c)$.

*VM is calculated on the posterior and taking average over all samples.
Table 2: Results of interpretable language generation on DD. Mutual information (MI), BLEU and homogeneity with actions (act) and emotions (em) are shown. The larger ↑, the better.

| Model                  | MI   | BLEU | act | em |
|------------------------|------|------|-----|----|
| DI-VAE                 | 1.28 | 3.05 | 0.18| 0.09|
| semi-VAE               | 0.03 | 4.06 | 0.02| 0.08|
| semi-VAE + Lmi         | 1.21 | 3.69 | 0.21| 0.14|
| GMVAE                  | 0.00 | 2.03 | 0.08| 0.02|
| DGMVAE − Lvar          | 1.41 | 2.98 | 0.19| 0.09|
| DGMVAE − Lmi           | 0.53 | 7.63 | 0.11| 0.09|
| DGMVAE                 | 1.32 | 7.39 | 0.23| 0.16|

Table 3: Dialog evaluation results on SMD. Four automatic metrics: BLEU, average (Ave.), extrema (Ext.) and greedy (Grd.) word embedding based similarity are shown. Response quality and consistency within the same c are scored by human.

| Model                  | BLEU | Ave. | Ext. | Grd. |
|------------------------|------|------|------|------|
| DI-VAE                 | 7.06 | 76.17| 43.98| 60.92|
| DGMVAE                 | 10.16| 78.93| 48.14| 64.87|

Table 4: Example actions (Act) and corresponding utterances (Utt) discovered by DGMVAE on SMD. The action name is annotated by experts.

| Act | Inform-route/address |
|-----|----------------------|
| Utt | There is a Safeway 4 miles away. |
|     | There are no hospitals within 2 miles. |
|     | There is Jing Jing and PF Changs. |

| Act | Request-weather |
|-----|----------------|
| Utt | What is the weather today? |
|     | What is the weather like in the city? |
|     | What’s the weather forecast in New York? |

Table 5: Dialog cases on SMD, which are generated by sampling different c from policy network. The label of sampled c are listed in parentheses with the annotated action name.

| Context                  | Predict                  |
|--------------------------|--------------------------|
| Sys: Taking you to Chevron. | (1-1-3, thanks) Thank you car, let’s go there! |
|                          | (1-0-2, request-address) What is the address? |
| User: Make an appointment for the doctor. | (3-2-4, set-reminder) Setting a reminder for your doctor’s appointment on the 12th at 3pm. |
|                          | (3-0-4, request-time) What time would you like to be schedule your doctor’s appointment? |

4.4 Interpretable Generation Results

Following Zhao et al. (2018b), we include the experiments of interpretable language generation on DD and dialog generation on SMD, respectively.

Because utterances in DD are annotated with Action and Emotion labels, we evaluate the ability of DGMVAE to capture these latent attributes on DD. We take the index i with the largest posterior probability \( q(\phi = i|x) \) as latent action labels. Following Zhao et al. (2018b), we use homogeneity as the metric to evaluate the consistency between golden action and emotion labels with labels obtained from DGMVAE. The number of our labels is 125. Results of homogeneity of action (act) and emotion (em) together with MI term and BLEU are shown in Tab. 2. It shows that DGMVAE outperforms other VAEs in reconstruction and gives the best homogeneity on both the action and emotion.

We also evaluate the ability of interpretable dialog generation of DGMVAE on SMD. Both automatic evaluation and human evaluation are conducted. BLEU and three word embedding-based topic similarity (Serban et al., 2017): Embedding Average, Embedding Extrema and Embedding Greedy (Mitchell and Lapata, 2008; Forgues et al., 2014; Rus and Lintean, 2012) are used to evaluate the quality of responses. In addition, three human evaluators were asked to score the quality (from 0 to 3) of 159 responses generated by DI-VAE and DGMVAE. Because SMD does not offer human annotated action labels of dialog utterances, we follow Zhao et al. (2018b) to label dialog actions by human experts for each discrete latent variable \( c \), according to their sampled utterances. Another 3 annotators are asked to evaluate the consistency between the action name and another 5 sampled utterances, which showing the interpretability.

Results are shown in Tab. 3. Both automatic and human evaluations show that DGMVAE obtains better generation quality and interpretability than DI-VAE on SMD. We perform one-tail t-tests on human evaluation scores and find that the superiority of our model is significant in both quality and consistency with p-values no more than 0.05.

We perform case studies to validate the performance of DGMVAE qualitatively. Some dialog actions with their utterances discovered by DGMVAE are shown in Tab. 4. It can be seen that utterances of the same actions could be assigned with the same discrete latent variable \( c \). We also give some dialog cases generated by DGMVAE in Tab. 5 with their contexts. Given the same context, responses with

---

8We use GloVe (Pennington et al., 2014) word embeddings of 300 dimension trained on 840B tokens from https://nlp.stanford.edu/projects/glove.
different actions are generated by sampling different values of discrete latent variables, which shows that DGMVAE has the ability to generate diverse and interpretable responses. More cases can be found in the supplementary materials.

5 Conclusion

The mode-collapse problem always occurs in GMVAE practically. In this paper, we give a theoretical analysis of this problem. Given the theoretical insights, we propose the DGMVAE, which can effectively alleviate the mode collapse problem. Additionally, experimental results show that DGMVAE outperforms a line of related works, obtaining higher language generation performance and better interpretable results.

References

Christopher M Bishop. 2006. Pattern Recognition and Machine Learning.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 10–21.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: interpretable representation learning by information maximizing generative adversarial nets. Neural Information Processing Systems, pages 2180–2188.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In NIPS 2014 Workshop on Deep Learning, December 2014.

Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. 2016. Deep unsupervised clustering with gaussian mixture variational autoencoders. arXiv preprint arXiv:1611.02648.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 37–49.

Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. 2014. Bootstrapping dialog systems with word embeddings. In NIPS, Modern Machine Learning and Natural Language Processing Workshop, volume 2.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. National Conference on Artificial Intelligence.

Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. International Conference on Learning Representations.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation, 9(8):1735–1780.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In International Conference on Machine Learning, pages 1587–1596.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. 2017. Variational deep embedding: An unsupervised and generative approach to clustering. International Joint Conference on Artificial Intelligence, pages 1965–1972.
Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In Advances in Neural Information Processing Systems, pages 3581–3589.

Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017a. Deep recurrent generative decoder for abstractive text summarization. Empirical Methods in Natural Language Processing, pages 2091–2100.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017b. Dailydialog: A manually labelled multi-turn dialogue dataset. International Joint Conference on Natural Language Processing, 1:986–995.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn treebank. Computational Linguistics, 19(2):313–330.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing lstm language models. arXiv preprint arXiv:1708.02182.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. International Conference on Machine Learning, pages 1727–1736.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. Interspeech, pages 1045–1048.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. Proceedings of ACL-08: HLT, pages 236–244.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082.

Vasile Rus and Mihai Lintean. 2012. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, pages 157–162. Association for Computational Linguistics.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 627–637.

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In Thirtieth AAAI Conference on Artificial Intelligence.

Iulian V. Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In Thirty-First AAAI Conference on Artificial Intelligence.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pages 553–562. ACM.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11(12):3371–3408.
Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational autoencoders for text generation. arXiv preprint arXiv:1903.07137.

Tsunghsien Wen, Yishu Miao, Phil Blunsom, and Steve J Young. 2017. Latent intention dialogue models. International Conference on Machine Learning, pages 3732–3741.

Eric P Xing, Michael I Jordan, and Stuart Russell. 2003. A generalized mean field algorithm for variational inference in exponential families. In Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence, pages 583–591. Morgan Kaufmann Publishers Inc.

Jiacheng Xu and Greg Durrett. 2018. Spherical latent spaces for stable variational autoencoders. Empirical Methods in Natural Language Processing.

Biao Zhang, Deyi Xiong, Hong Duan, Min Zhang, et al. 2016. Variational neural machine translation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 521–530.

Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M Rush, and Yann Lecun. 2018a. Adversarially regularized autoencoders. International Conference on Machine Learning, pages 5897–5906.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2017a. Infovae: Information maximizing variational autoencoders. arXiv: Learning.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. 2018b. 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 Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 654–664.

Chunting Zhou and Graham Neubig. 2017. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. Meeting of the Association for Computational Linguistics, 1:310–320.
A Appendix

A.1 Derivation of Mutual Information Term and Final Objective

In this part, we first split the mutual information term from ELBO of GMVAE. We focus on the $\mathcal{R}_c$ over corpus,

$$\mathbb{E}_{p(x)} \mathbb{E}_{q(z|x)q(c|x)} \log \frac{p(c)}{q(c|x)}.$$  \hfill (5)

Out of the mean-field approximation, expectation on $q(z|x)$ could be removed by integral.

$$\mathbb{E}_{p(x)} \mathbb{E}_{q(c|x)} (\log p(c) - \log q(c|x)) = \mathcal{H}(c|x) + \mathbb{E}_{q(c)} \log p(c)\quad (6)$$

in which, $q(c) = \mathbb{E}_{p(x)} q(c|x)$ and $\mathcal{I}(c, x) = \mathcal{H}(c) - \mathcal{H}(c|x)$ is a mutual information between discrete latent variable $c$ and input data $x$.

Replacing $\mathcal{R}_c$ by Eq. (5) and replacing $\mathcal{R}_z$ by two terms we derived in Theorem 2, the ELBO of dataset is written as

$$\mathbb{E}_x \text{ELBO} = \mathbb{E}_x \mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z) - \text{KL}(q_\phi(c)||p(c)) - \mathcal{I}(c, x) - \mathbb{E}_x [\text{KL}(q_\phi(z|x)||\hat{p}(z|x))] - \frac{1}{2\sigma^2} \mathbb{E}_x \text{Var}_{q_\phi(z|x)} \mu_c.\quad (7)$$

Adding regularization terms $\mathcal{L}_{\text{mi}}$ and $\mathcal{L}_{\text{var}}$ to ELBO, the final objective is

$$\mathcal{L}(\theta) = \mathbb{E}_x \mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z) - \text{KL}(q_\phi(c)||p(c)) - \mathbb{E}_x [\text{KL}(q_\phi(z|x)||\hat{p}(z|x))] - (1 - \beta) \mathbb{E}_x \frac{1}{2\sigma^2} \text{Var}_{q_\phi(c|x)} \mu_c,\quad (8)$$

which consists of a reconstruction term, a KL divergence over $c$, a KL divergence over $z$ and a variance of means term.

A.2 Proof of Theorem 1

We first simplify $\mathcal{R}_c$:

$$\mathcal{R}_c = \mathbb{E}_{p(c_i|\mu_\phi)} \log \frac{p(c_i)}{p(c_i|\mu_\phi)}\quad (9)$$

where $K$ is the number of Gaussian components and

$$g(\mu_\phi) = \sum_i \frac{1}{K} \cdot p(\mu_\phi|c_i).\quad (10)$$

It is because

$$p(c_i) = \frac{1}{K}\quad (11)$$

and

$$p(c_i|\mu_\phi) = \frac{p(\mu_\phi|c_i)}{\sum_i p(\mu_\phi|c_i)}\quad (12)$$

Now we derive the partial derivative of $\mathcal{R}_c$ with respect to $p(\mu_\phi|c_i)$:

$$\frac{\partial \mathcal{R}_c}{\partial p(\mu_\phi|c_i)} = d_1 + d_2 + d_3 + d_4,\quad (13)$$

12
We prove this property in the case of univariate Gaussian distribution priors. It is straightforward to

Adding these components, we get:

\[ \frac{\partial R}{\partial p(\mu_\phi|c_i)} = - \sum_k p(\mu_\phi|c_k) \log \frac{p(\mu_\phi|c_i)}{p(\mu_\phi|c_k)} \cdot p(\mu_\phi)^2. \]  

At the same time,

\[ \frac{\partial p(\mu_\phi|c_i)}{\partial \mu_i} = p(\mu_\phi|c_i)(\mu_\phi - \mu_i). \]

Finally, we multiply \( \frac{\partial R}{\partial p(\mu_\phi|c_i)} \) and \( \frac{\partial p(\mu_\phi|c_i)}{\partial \mu_i} \) to get \( \frac{\partial R}{\partial \mu_i} \), because \( \frac{\partial p(\mu_\phi|c_i)}{\partial \mu_j} = 0 \) for \( i \neq j \):

\[ \frac{\partial R}{\partial \mu_i} = \frac{\partial R}{\partial p(\mu_\phi|c_i)} \frac{\partial p(\mu_\phi|c_i)}{\partial \mu_i} = - \left( (\mu_\phi - \mu_i)p(\mu_\phi|c_i) \sum_k p(\mu_\phi|c_k) \log \frac{p(\mu_\phi|c_i)}{p(\mu_\phi|c_k)} \right) \cdot p(\mu_\phi)^2. \]

Define \( \frac{\partial R}{\partial \mu_i} = \left[ \frac{\partial R}{\partial p(\mu_\phi|c_i)} \right]_{i=1,2,...,K} \) and \( \frac{\partial S_{\mu_\phi}}{\partial p(\mu_\phi|c_i)} = \left[ \frac{\partial S_{\mu_\phi}}{\partial \mu_i} \right]_{i=1,2,...,K} \),

\[ \left[ \frac{\partial R}{\partial \mu_i} \right] \cdot \frac{\partial S_{\mu_\phi}}{\partial p(\mu_\phi|c_i)} = \sum_i \frac{\partial R}{\partial \mu_i} \frac{\partial S_{\mu_\phi}}{\partial \mu_i} \]

\[ = 2 \sum_i u_i u_i \sum_k u_k(w_i - w_k) \cdot p(\mu_\phi)^2 \]

\[ = 2 \sum_{i<j} u_i u_j (v_i - v_j)(w_i - w_j) \cdot p(\mu_\phi)^2. \]

where

\[ u_k = p(\mu_\phi|c_k), \]

\[ v_k = (\mu_\phi - \mu_k)^2, \]

\[ w_k = \log p(\mu_\phi|c_k), \]

as

\[ \frac{\partial S_{\mu_\phi}}{\partial \mu_i} = 2(\mu_i - \mu_\phi). \]

As \( (v_i - v_j)(w_i - w_j) \leq 0, \left[ \frac{\partial R}{\partial \mu_i} \right] \cdot \frac{\partial S_{\mu_\phi}}{\partial p(\mu_\phi|c_i)} \leq 0 \). So when performing gradient ascent on ELBO, the gradient of \( R_z \) will make \( S_{\mu_\phi} \) smaller. As a result, \( \text{Var}_M (\leq \frac{S_{\mu_\phi}}{R_z}) \) is limited by a decreasing bound.

### A.3 Proof of Theorem 2

We prove this property in the case of univariate Gaussian distribution priors. It is straightforward to
generalize it to diagonal multivariate Gaussian distributions by summing over all dimensions.

The \( R_z \) term can be rewritten as

\[ \mathbb{E}_{q_\phi(z|x)} \sum_c q_\phi(c|x) \log \frac{p(z|c)}{q_\phi(z|x)} = \mathbb{E}_{q_\phi(z|x)} \log \frac{\prod_c p(z|c)q_\phi(c|x)}{q_\phi(z|x)}. \]
We define $p(z|c)$ as a Gaussian distribution with mean $\mu_c$ and standard deviation $\sigma$,

$$p(z|c) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(z - \mu_c)^2}{2\sigma^2}\right). \quad (27)$$

We define $f(z; x) = \prod_c p(z|c)^{q_x(c|x)}$, which can be written as

$$f(z; x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2} \sum_c q_x(c|x)(z - \mu_c)^2\right)$$

$$= \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2} (z - \mathbb{E}_{q_x(c|x)}(\mu_c))^2\right) \cdot \exp\left(-\frac{1}{2\sigma^2} (\mathbb{E}_{q_x(c|x)}(\mu_c)^2 - \mathbb{E}_{q_x(c|x)}[\mu_c^2])\right). \quad (28)$$

According to Eq. 28, $f(z; x)$ can be split into two multiplied terms. The first term is the probability density function of a Gaussian distribution, denoted as $\hat{p}(z|x)$,

$$\hat{p}(z|x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2} (z - \mathbb{E}_{q_x(c|x)}(\mu_c))^2\right), \quad (29)$$

whose mean is the expectation of $\mu_c$ under posterior distribution $q_x(c|x)$ and standard deviation is $\sigma$. The second term is actually the variance of means of Gaussian components under the distribution of $q_x(c|x)$,

$$\text{Var}_{q_x(c|x)} \mu_c = \mathbb{E}_{q_x(c|x)}[\mu_c^2] - \mathbb{E}_{q_x(c|x)}[\mu_c]. \quad (30)$$

According to Eq. 28 30, Eq. 26 consists of two terms:

$$-\text{KL}(q_x(z|x)||\hat{p}(z|x))) - \frac{1}{2\sigma^2} \text{Var}_{q_x(c|x)} \mu_c. \quad (31)$$

The expectation on $q_x(z|x)$ of the second term is removed by integral since $\text{Var}_{q_x(c|x)} \mu_c$ is independent on $z$. In Eq. 31 there contains a negative variance term of the prior means.

### A.4 Hyper-Parameters

Adam optimizer is adopted with learning rate of 0.001. Batch size is set to 30. The size of word embedding is set as 300. Sentence longer than 40 will be cut off. Vocabulary size is set to 10,000. All results were obtained by repeating the experiment three times and taking an average.

We also illustrate how $\beta$ will affect the model performance in Fig. 4 and we generally find that larger $\beta$ can get better results. As a result, we use $\beta = 0.9$ for experiment in PTB and DD.

### A.5 More Cases

More examples of actions discovered by DGMVAE is shown in Tab. 6. More examples on responses generated by DGMVAE are shown in Tab. 7. In Tab. 7 an example without context is given to show the ability to begin a dialog in different topics (weather, navigation and scheduling). For DGMVAE, we can sample different continuous latent variables from one component. As shown in Tab. 8 diverse responses with the same actions could be generated.
Table 6: Example actions discovered by DGMVAE on SMD. The action name is annotated by experts.

| Action Name | Utterances                                                                 |
|-------------|-----------------------------------------------------------------------------|
| Request-location | Which location do you want the weather for? Which location should I look up information about? Which city are you asking about? |
| Inform-time/appointment | Your next dinner event is with your father on Friday. Your father will be attending your yoga activity on the 2nd with you. Your doctor’s appointment is Monday at 1 pm. |
| General-thanks | Thanks. Thanks a lot. Perfect. Thanks. |

Table 7: Dialog cases on SMD, which are generated by sampling different c from policy network. The label of sampled c are listed in parentheses with the annotated action name. When context is None, it means to predict the beginning a dialog.

| Context | User: What’s the temperature going to be this week? |
|---------|---------------------------------------------------|
| Sys:    | What city are you wanting to know the temperature for? |
| Predict | (1-1-0, inform-address) Cleveland. (4-0-2, request-weather) Will it rain in Redwood City today? |

| Context | None |
|---------|------|
| Predict | (1-4-0, request-route/address) find me a nearby coffee shop (1-0-4, request-weather) what’s the weather going to be like today and tomorrow (1-2-4, command-reminder) remind me about meeting later |

Table 8: Dialog cases on SMD, which are generated by sampling different z from the same actions (i.e., the c).

| Context | User: What is the highest temperature in Brentwood over the next two days? |
|---------|--------------------------------------------------------------------------|
| Action Name | (2-3-0) inform-weather |
| Predict | It is currently foggy in Brentwood on Tuesday. It will be between 70 - 40F and turn - 40F on Saturday. |

| Context | User: I need gas. |
|---------|------------------|
| Action Name | (2-4-2) inform-route/address |
| Predict | There is a Chevron 3 miles from you. There is a Safeway. |

| Context | User: schedule meeting |
|---------|-----------------------|
| Action Name | (3-0-4) request-time |
| Predict | What day and time should I set your meeting for? What time should I set the alarm? |