PCVAE: Generating Prior Context for Dialogue Response Generation

Zefeng Cai*, Zerui Cai*
East China Normal University
oklen@foxmail.com, zrcai_flow@126.com

Abstract
Conditional Variational AutoEncoder (CVAE) is promising for modeling one-to-many relationships in dialogue generation, as it can naturally generate many responses from a given context. However, the conventional used continual latent variables in CVAE are more likely to generate generic rather than distinct and specific responses. To resolve this problem, we introduce a novel discrete variable called prior context which enables the generation of favorable responses. Specifically, we present Prior Context VAE (PCVAE), a hierarchical VAE that learns prior context from data automatically for dialogue generation. Meanwhile, we design Active Codeword Transport (ACT) to help the model actively discover potential prior context. Moreover, we propose Autoregressive Compatible Arrangement (ACA) that enables modeling prior context in autoregressive style, which is crucial for selecting appropriate prior context according to a given context. Extensive experiments demonstrate that PCVAE can generate distinct responses and significantly outperforms strong baselines.

1 Introduction
Researchers from both academic and industrial communities have paid increasing attention to open domain dialogue responses generation since it is promising in real-world applications. In this task, for a given context, usually, there are more than one valid responses, which is the so-called one-to-many problem [Csaky et al., 2019]. For CVAEs [Sohn et al., 2015], they have been widely used in dialogue responses generation. The conventionally used continual latent variables in CVAE make it convenient to sample in the latent space for generating different responses, however, they are not suitable for generating distinct and specific responses but generic results. In CVAE, a context is mapped to distribution in latent space, and we sample a latent variable in latent space according to the distribution. This scheme tends to capture latent variables around the center of the distribution. However, the distinct responses are far away from each other [Sun et al., 2021], which causes them hardly be sampled. To deal with this problem, we propose to introduce a novel discrete variable called prior context as shown in Figure 1. Specifically, we employ vector quantization to quantize the encoded responses into discrete latent variables (codewords) as prior context. Thus, during the test time, we can sample from possible codewords to aid the model to generate distinct and specific responses. Although vector quantization for discrete latent variables has been studied and applied in various areas, two critical issues that remain to be addressed to achieve our goals: i) It is necessary to ensure distinctness and diversity of prior context for generating more specific responses. However, the training of vector quantization is unstable, and the frequently happened codebook collapse problem causes only a small portion of discrete latent variables to be used, which inevitably leads to sub-optimal performance. ii) selecting an appropriate prior context according to the input context is extremely important for superior performance, however, so far it is a rarely explored problem. To resolve it, a straightforward way is to employ an autoregressive model to predict them. However, in our early experiment, simply training a model such as a GRU [Cho et al., 2014] is unable to reach convergence, which causes poor results in testing. To address the above two challenges, we present a novel CVAE model, namely Prior Context Variational AutoEncoder (PCVAE), to

Figure 1: Illustration of prior context in the latent space of CVAE. The input context is encoded into a distribution of latent variables and the decoder can decode a sampled latent variable into a response. When sampling in latent space, it is more likely to obtain latent variables around the center of the distribution (deeper colored areas), however, latent variables corresponding to distinct and specific responses are usually far away from each other. As a result, most of them hardly be sampled since they can not get together around the center. Intuitively, employing prior context in the latent space enables our model to generate responses that are initially hard to be sampled, and we find these responses are often specific and distinct in our early experiments.
model prior context with two key components:

(1) We propose active codeword transport (ACT) to actively pull the input embedding towards unused codeword, which not only resolves the codebook collapse problem but also improve the distinctness and diversity of prior context.

(2) For the non-convergence problem, we conjecture the dependency between codewords may be initially unordered, which is different from a natural language where the next words are subject to the previous words, causing difficulty for a autoregressive style model to learn them. To deal with it, we design an autoregressive codeword arrangement (ACA) that regularizes the conditional probability distribution between codewords to fit the autoregressive patterns predicted by a GRU, which is crucial for successful training.

In our experiments, we compare our model with various dialogue CVAE baselines on two authoritative dialogue datasets and conduct further experiments to verify the effectiveness of our proposed model.

To conclude, our contributions can be summarized as follow:

• We introduce the prior context for generating distinct and specific responses in dialogue generation and propose PCVAE that models prior context with discrete latent variables. To the best of our knowledge, PCVAE is the first model to learn and select prior context automatically without manual intervention.

• We propose ACT which resolves the codebook collapse problem and prompts the model to automatically discover potential prior context. Meanwhile, ACA is designed to deal with the non-convergence problem in the training of the autoregressive model, which is crucial for selecting appropriate codeword combinations of prior context for a given context.

• Empirical experiments demonstrate the effectiveness of our model. Further analyses reveal the unique advantages of our methods.

2 Related Work

Dialogue Generation. The dialogue generation in the open domain is a challenging task. Early works [Graham, 2015; Sordoni et al., 2015] suffered from the generic response problem. To tackle this problem, there are two major approaches including improving the architecture of the neural dialog model and introducing external knowledge. In this paper, we focus on the former one which includes enhancing the model with attention mechanism [Bahdanau et al., 2015; Luong et al., 2015], applying Reinforce Learning [Liu et al., 2020; Zhang et al., 2018a], GAN [Feng et al., 2020; Zhang et al., 2018b], and variational reasoning [Zhao et al., 2017; Gao et al., 2019; Sun et al., 2021; Zhao et al., 2018].

Vector Quantization. Vector quantization in VAE is first proposed by [van den Oord et al., 2017] for image generation. The process of quantization is selecting a vector (codeword) that is closest to the input vector from a random-initialized vector table (codebook). We refer to the selected vector as the quantized vector of the input.

However, vector quantization suffers from the notorious codebook collapse problem. The codebook collapse is that the input vectors are only mapped to a very small portion of the codewords, which results in the inferior representation of the discrete latent variables. The existing methods that deal with codebook collapse include random restart [Dhariwal et al., 2020; Lancucki et al., 2020] that reinitializes the codeword in the codebook to improve the usage and Population-Based Training (PBT) [Jaderberg et al., 2017; Dieleman et al., 2018] that dynamically adjusts the hyper-parameters in the objective function of vector quantization. However, the random restart inevitably changes the indexes of codewords which disturbs the process of selecting prior context and the PBT method can only prevent the decrease of codeword usage but we expect more unused codewords could be utilized. In a word, no existing method is well fitted for our needs. Thus, we propose an ACT to satisfy our demand.

Conditional Variational Autoencoder. The CVAE [Sohn et al., 2015; Yan et al., 2016] is a variational reasoning model that uses a conditional signal (context) to generate more specific data (responses). CVAEs have been widely applied in dialogue response generation. Previous work that introduces manually defined information including dialog act [Zhao et al., 2017], word-level representation [Gao et al., 2019] can be viewed as utilizing a discrete latent variable with explicit semantic meaning. Except for methods using manually predefined information, there are unsupervised methods including DI-VAE and DI-VST [Zhao et al., 2018] focus on improving the interpretability of learned discrete latent variables, and SpeaCVAE [Sun et al., 2021] uses the clustering method to find group information to resolve the one-to-many and many-to-one problem.

Our work differs from these as follows: (1) We focus on automatically discovering potential prior context, while previous work uses a fixed number of discrete latent variables which is rather limited in the real world. (2) We improve the quality of the codebook in vector quantization and overcome the well-known codebook collapse problem. (3) a comprehensive solution is proposed to properly select prior context.

3 Proposed Methods

We define \( x \in X \) as a response utterance, \( c \in C \) as a given context. In a dialogue CVAE, the goal is to model \( p(x) = \int p(x|c)p(c)dc \). In our model, we further introduce the \( y \in Y \) and \( z \in Z \) that are latent variables of prior context and response, respectively. Thus, this goal can be rewritten as modeling \( p(x) = \int p(x|z,c)p(z,c)dzdc \) and \( p(z,c) = \int p(z|y,c)p(y|c)p(c)dy \). We employ neural networks to model those distributions. We refer to the continual latent variables \( p_\phi(z|y,c) \) as a prior network and we introduce a recognition network \( p_\theta(z|x,c) \) to approximate the true posterior distribution \( q(z|y,c) \). Here, \( \phi \) and \( \theta \) represent the parameters of the prior network and recognition network, respectively. Both \( p_\phi(z|y,c) \) and \( p_\theta(z|x,c) \) are assumed to follow isotropic Gaussian distribution. The \( p(x|z,c) \) generation network follows a Dirac distribution and the \( p(y|c) \) is a discrete latent variable that follows a sequential conditional probabilistic distribution modeled by a prior context planning.
network. The overview of our model is shown in Figure 2.

In the following, we will first introduce the encoding phrase including prior network, recognition network. Then we introduce decoding phrase including generation network. Then we illustrate how our model learns prior context automatically with vector quantization to generate discrete latent variables $y$ and prior context planning network to learn $p(y|c)$. Finally, we illustrate active codeword transport and autoregressive codeword arrangement in detail.

3.1 Encoding

In this section, we show how to use a prior network to encode context and a recognition network to encode responses for obtaining a distribution of their associated latent variables. A two-layer GRU is used to encode input context as $h_c$ and response utterance as $h_x$. Then, to obtain the latent variables that capture deeper semantic meaning, the $h_c$ and $h_x$ are compressed through $N_f$ layers of MLPs, which results in $e_c$ and $e_x$, respectively. After that, a prior network and a recognition network are employed to obtain the parameters of their corresponding continual latent variables distribution, which can be described as follows:

$$
\begin{align*}
\begin{bmatrix}
    \mu_c \\
    \log(\sigma_c^2)
\end{bmatrix} &= \text{MLP}_c([e_c, y]),
\begin{bmatrix}
    \mu_x \\
    \log(\sigma_x^2)
\end{bmatrix} &= \text{MLP}_x([e_x, e_c])
\end{align*}
$$

where $[\cdot, \cdot]$ means concatenation of variables, $e'_c$ is a conditional signal to guide the generation in decoding, $y$ is learned discrete latent variable for prior context, which will be described later. We define $p_\theta(z|y,c) \sim \mathcal{N}(\mu_z, \sigma_z^2\mathbf{1})$ and $p_\theta(z|x,c) \sim \mathcal{N}(\mu_z, \sigma_z^2\mathbf{1})$. The reparameterization trick [Kingma and Welling, 2014] is used to sample a latent variable $z$ from the $p_\theta(z|x,c)$ in training phase and $p_\theta(z|y,c)$ in testing phase, respectively. We employ KL divergence $D_{KL}(\cdot)$ to make $p_\theta(z|y,c)$ approximate to $p_\theta(z|x,c)$.

3.2 Decoding

In this section, we introduce the decoding process of our model that utilizes the output of the prior network and recognition network. In generation network, the conditional signal $e'_c$ and $z$ is concatenated as the input $e_d = [e'_c, z]$. The information of responses are reconstructed from $e_y$ through $N_f$ layers of MLP. The final output is used as the initial states of a two-layer GRU to generate the expected responses. We use negative log-likelihood $L_{CG}$ as the objective function of generation.

3.3 Prior Context Learning

Vector Quantization. We employ a codebook with random initialization codewords $v_i$ and $i \in \{1, \ldots, N_v\}$. The input $e_x$ and $e_c$ are transformed into an input vector for quantization as $e_y = \text{MLP}([e_x, e_c])$. After that, $e_y$ is chunked into $N_K$ parts $e_{y,j}$ with the same size of $v_i$. We achieve quantization by quantizing function as follows:

$$
I(e_{y,j}, v_k) = \begin{cases} 
1 & \text{for } k = \arg\min_{k'} \|e_{y,j} - v_{k'}\|_2 \\
0 & \text{otherwise}
\end{cases}
$$

when $I(e_{y,j}, v_k) = 1$, we map $e_{y,j}$ to $v_k$. Then we concatenate all selected $v_k$ to obtain the discrete latent variable $y$. We apply the straight-through estimator [Bengio et al., 2013] to train the codewords as:

$$
L_{vq} = \sum_{j=1}^{N_K} \sum_{k=1}^{N_v} I(e_{y,j}, v_k)(\|\sgn[v_k] - e_{y,j}\|_2^2 + \beta \|v_k - \sgn(e_{y,j})\|_2^2)
$$

where the stop-gradient operation $\sgn$ is used since the above selecting approach is intractable. $\beta$ is a weight coefficient.

Prior Context Planning. The indexes of selected codewords can be viewed as an ordered index sequence. We employ a GRU $p(y|c)$ to predict $y$, as we do not have the ground-truth response to acquire $e_y$ in the testing phase. To learn the $p(y|c)$, we can obtain it through marginalizing the $x$ in $p(y|c,x)$ in training as follows:

$$
p(y|c) = \sum_x p(y|c,x)p(x)
$$

where the $p(y|c,x)$ is the probability distribution of quantizing function $I$. Then we can train the GRU in an autoregressive...
man \text{e} \text{ generating mean predicting:} 

Figure 3. We dynamically estimate this center using moving are extensive unused codewords around it as shown in the
test if the model has to generate a re-
ness the current unused codewords would be selected and
use as shown in the Figure. 3. We dynamically estimate this center using moving mean predicting:

\[ e_c = e_c \cdot \gamma_m + (1 - \gamma_m) \sum_{i=1}^{N_B} e_y^i \]  

Where the \( N_B \) is minibatch size and \( e_y^i \) represents the \( i \)-th sample in a batch. Then, we calculate the corresponding direction vector \( \hat{e}_c \) which we use to direct vector to predict the near unused codewords and define them as a target set \( T \) while \( e_y \) in the same batch are as a source set \( S \):

\[ T = \{ \mathbf{vq}(e_y^i + j * \mathbf{e}_c^i) \} \]
\[ S = \{ e_y^{(j)} \} \]  

where \( j \in \{0, 1, \cdots, N_K\}, i \in \{0, 1, \cdots, N_B\} \), \( \mathbf{vq} \) means the vector quantization operation as we have defined, \((j)\) in \( S \) indicate additional repeated elements to balance the number of elements between \( T \) and \( S \). After finding targets, the second problem is how to assign the source to the target appropriately. We must avoid different \( e_y \) being transported to the same unused codeword while minimizing the total moving distance. We can formalize this problem as an optimal transport problem and employ Wasserstein distance to resolve it:

\[ W(\varphi, \nu) = \inf_{\pi \in \Pi(\varphi, \nu)} \int_{s \in T} d(s, t) d\pi(s, t) \]  

where the transport plans \( \pi \) that distributes the mass in \( \varphi \) to match that in \( \nu \). The ground metric \( d(s, t) = \|s - t\|_2 \) provides the cost of moving a unit of mass from \( s \sim \varphi \) to \( t \sim \nu \). However, the above equation is intractable, therefore we tend to employ a sinkhorn divergence [Cuturi, 2013] to get an approximate optimal transport solution \( \mathcal{L}_{act} = W^*(\varphi, \nu) \) for training.

3.4 Training Objective

In PCVAE, the training objective includes six parts: (1) response generation loss \( \mathcal{L}_G \), (2) posterior approximating loss \( \mathcal{L}_K \), (3) vector quantization loss \( \mathcal{L}_{vq} \), (4) prior context planning loss \( \mathcal{L}_{pcp} \), (5) autoregressive codeword arrangement loss \( \mathcal{L}_{aca} \), and (6) active codeword transport loss \( \mathcal{L}_{act} \). The total loss is as follows, where \( \lambda_1, \lambda_2 \) are weight factors:

\[ \mathcal{L}_{total} = \mathcal{L}_G + \lambda_1 \mathcal{L}_K + \mathcal{L}_{vq} + \mathcal{L}_{pcp} + \mathcal{L}_{aca} + \lambda_2 \mathcal{L}_{act} \]  

4 Experiment

Datasets. We employ two authoritative datasets for our experiment, including MultiWoz [Zang et al., 2020] for cross-domain task-oriented dialogue and Cornell Movie [Danescu-Niculescu-Mizil and Lee, 2011] for open-domain dialogue. Specifically, we use MultiWoz 2.2, which contains 3,406 single-domain dialogues and 7,032 multi-domain dialogues, and all dialogues are task-oriented. The Cornell Movie consists of 220,579 conversational exchanges between 10,292 pairs of movie characters. We further convert them into two turn dialogue datasets that the model has to generate a response given three context utterances. Although on single turn dialogue the one-to-many situations appear more frequently, it may just contain an uninformative utterance such as “ok” where too many acceptable responses exist.

Baselines. We choose the Seq2Seq model, CVAE, and various dialogue CVAEs as baselines. kgCVAE [Zhao et al., 2017] uses manually predefined dialog acts as additional latent variables. SepaCVAE [Sun et al., 2021] uses an unsupervised clustering method to obtain group information to guide the generation. DCVAE [Gao et al., 2019] replaces conventional continual latent variables with discrete latent variables and adopts the predefined word-level knowledge. Note that we do not compare PLM/RL/GAN-based methods since we
focus on the improvement from introducing prior context, and we can easily replace our backbone with other architectures for better performances.

**Metrics and Evaluation.** We employ several widely used metrics, including BLEU-1, BLEU-4 [Papineni et al., 2002], Distinct-1, Distinct-2 [Li et al., 2016], and METEOR [Banerjee and Lavie, 2005]. All results are the mean values of five runs with different random seeds.

**Implementation Details.** We use word embeddings with 200 dimensions and hidden states with 300 dimensions for encoding and decoding GRU. We initialize the word embedding from Glove embedding [Pennington et al., 2014] and use the NLTK tokenizer [Bird et al., 2009]. The number of layers $N_D$ of MLPs for compression and reconstruction is set to 2 with hidden sizes ranging from 200 to 300. We use $N_K = 4$ codebooks and $N_E = 8192$ codewords. The $\gamma_m$ used in moving mean predicting is set to 0.95. The $\beta$ used in vector quantization is set to 0.25. In training, we use batch size $N_B = 192$ and Adam optimizer with an initial learning rate of 1e-3 for both of the datasets. We decrease the learning rate by 0.8 when the worse valid loss is obtained in the validation phase and stop training as the learning rate is down to 1e-5. For other models, we adopt their official code if available. Otherwise, we adapt their key techniques to our model. For a fair comparison, we replace their encoder and decoder with the same as our model.

**4.1 Responses Generation Performance**

The experiment results are shown in Table 1. As we can see, PCVAE outperforms strong baselines significantly on both datasets. The higher BLEU and Distinct implies the effective of specific prior context, which is beneficial for improving the diversity and distinctness of the generated responses. Moreover, PCVAE obtains more performance gain on open-domain dataset (Cornell Movie) than multi-domain task-oriented dataset (MultiWoz), which implies that our model can better handle the one-to-many problem. Thus, we conjecture the performance gains of PCVAE mainly come from automatically discovered potential prior context, while other models can only rely on their limited signals. Further, we find the really used codeword in testing on MultiWoz are about 1400 while on cornell movie them are about 7400, which means our model can utilize more prior context when potential response are more diverse and distinct. It also confirms that our model benefits from the prior context.

Table 1: Responses generation performance. Improvements compute as relative gains compared with the previous state-of-the-art method. The best results are highlighted in boldface.

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU-1</th>
<th>BLEU-4</th>
<th>Distinct-1</th>
<th>Distinct-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiWoz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq2Seq</td>
<td>0.274</td>
<td>0.130</td>
<td>0.038</td>
<td>0.130</td>
</tr>
<tr>
<td>CVAE</td>
<td>0.402</td>
<td>0.191</td>
<td>0.075</td>
<td>0.506</td>
</tr>
<tr>
<td>kgCVAE</td>
<td>0.449</td>
<td>0.205</td>
<td>0.077</td>
<td>0.513</td>
</tr>
<tr>
<td>SepatCVAE</td>
<td>0.447</td>
<td>0.203</td>
<td>0.078</td>
<td>0.529</td>
</tr>
<tr>
<td>DCVAE</td>
<td>0.451</td>
<td>0.214</td>
<td>0.076</td>
<td>0.511</td>
</tr>
<tr>
<td>PCVAE</td>
<td>0.505</td>
<td>0.241</td>
<td>0.086</td>
<td>0.557</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Improvement (%)</th>
<th>11.89</th>
<th>12.61</th>
<th>10.83</th>
<th>5.22</th>
</tr>
</thead>
</table>

| Cornell Movie  |        |        |            |            |
| Seq2Seq        | 0.218  | 0.094  | 0.025      | 0.108      |
| CVAE           | 0.248  | 0.107  | 0.048      | 0.313      |
| SepatCVAE      | 0.278  | 0.120  | 0.050      | 0.413      |
| DCVAE          | 0.265  | 0.115  | 0.052      | 0.464      |
| PCVAE          | 0.411  | 0.203  | 0.071      | 0.661      |

<table>
<thead>
<tr>
<th>Improvement (%)</th>
<th>47.67</th>
<th>68.55</th>
<th>35.87</th>
<th>42.59</th>
</tr>
</thead>
</table>

**4.2 Ablation Study**

In this section, we evaluate the effectiveness of our proposed components. Specifically, we introduce several variants of PCVAE by discarding certain components. PCVAE-VQ removes both ACT and ACA with only vector quantization left, and PCVAE-None further removes the vector quantization to evaluate the backbone performances. PCVAE-ACT and PCVAE-ACA remove ACA and ACT, respectively. The ablated results are shown in Table 2. We can observe that: (1) simply applying vector quantization to the model can not bring any improvement. (2) Performances of all models without ACA are close and disappointing. We believe that this phenomenon is caused by the non-convergence problem of the prior context planning network that prevents a model from utilizing appropriate prior context. (3) Comparing the performance gap between PCVAE-ACT and PCVAE-VQ with that between PCVAE and PCVAE-ACA, although the ACT enables model access to better prior context, we can not obtain a satisfactory result without a proper selection. In turn, once we can appropriately select prior context, our model would fully benefited from prior context discovered by ACT, which significantly improves the performance of our model.

Table 2: The performance of various models for ablation study.

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU-1</th>
<th>BLEU-4</th>
<th>Distinct-1</th>
<th>Distinct-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCVAE-None</td>
<td>0.251</td>
<td>0.107</td>
<td>0.049</td>
<td>0.307</td>
</tr>
<tr>
<td>PCVAE-VQ</td>
<td>0.243</td>
<td>0.091</td>
<td>0.046</td>
<td>0.289</td>
</tr>
<tr>
<td>PCVAE-ACT</td>
<td>0.267</td>
<td>0.118</td>
<td>0.054</td>
<td>0.322</td>
</tr>
<tr>
<td>PCVAE-ACA</td>
<td>0.311</td>
<td>0.146</td>
<td>0.059</td>
<td>0.475</td>
</tr>
<tr>
<td>PCVAE</td>
<td>0.411</td>
<td>0.202</td>
<td>0.071</td>
<td>0.661</td>
</tr>
</tbody>
</table>

**4.3 Qualitative Analysis**

The randomly sampled responses generated by PCVAE and baseline models are shown in Table 3. The two samples are from Cornell Movie and MultiWoz, respectively. In the first sample, all other models response something related to the air tickets or trip, while only PCVAE properly answer the question *whether to see it off*. We conjecture this is mainly because our prior context can provide pertinent detail rather than the related general topic information, and it enables our model to generate more specific response. In the second sample, the context is asking a receipt. We can find that SepatCVAE and DCVAE realize the general meaning of context (purchasing) but fail to figure out the distinctness between ordering and requiring a receipt. For kgCVAE, although it provides a acceptable response for the given context, “it will be ready soon” is rather generic. In contrast, our response is specific and informative. We believe this is because prior

---

1. kgCVAE is not tested on Cornell Movie dataset since the dialog cat is unavailable.
context provided by our model contain more distinct information, which can aids our model generate response with points and avoid generic result.

4.4 Analysis

Effect of Prior Context. We further evaluate the effect of prior context on performance. To this end, we change $N_E$ to vary the representation capacity of a codebook and restrict the influence of prior context to see how it related to the model performance. We conduct the experiment on the MultiMoz dataset and show the results in Table 4. As we can see, increasing $N_E$ always leads to better performances, which verifies our intuition that prior context enables PCVAE to generate more diverse and relevant responses. Additionally, We observe that the performance gains gradually saturated as we keep increasing $N_E$. We believe this is because more available codewords make it harder to select prior context properly. It also means that we can always use a relatively large $N_E$ to obtain competitive performances.

Active Codeword Transport. We evaluate the effectiveness of ACT for overcoming the codebook collapse problem. To this end, we measure the codebook usage and the mean $\mathcal{L}_{vq}$ of vector quantization in testing. The $N_E$ is set to 8192. We also compare four different training setups: (1) Standard: without any heuristics; (2) RS: applying random restarts; (3) PBT: applying population based training. (4) ACT: applying active codeword transport. The experiment result is shown in Table 5. As we can see, the ACT method achieves the highest codebook usage, which demonstrates its superior performance among various previous methods. At the same time, ACT also reduces the mean vector quantization loss, which is beneficial for improving the quality of the codebook.

4.5 Human Evaluation

In this section, we provide a human evaluation of our model. Following [Sun et al., 2021], we randomly sample 200 responses generated by different models on the test set of MultiWOz, respectively. The samples are provided to three annotators with linguistic backgrounds, and we ask them to rank the generated responses considering fluency, diversity, and relevance, respectively. Ties are permitted. Fluency measures the closeness to words from humans, diversity measures the amount of specific information, and relevance measures semantic relevance to the context. The results are shown in Table 6. As we can see, although the fluency score of each model is close, PCVAE outperforms other methods significantly on diversity and relevance. It implies that PCVAE can generate more specific responses about the given context attributed to our superior prior context.

5 Conclusion

This paper proposes a novel hierarchical deep CVAE named PCVAE to automatically learn high-quality prior contexts for generating distinct and specific responses. Specifically, we introduce prior context, a discrete latent variable which aids model resolve the one-to-many problem in dialogue generation effectively. Moreover, we propose active codeword transport and autoregressive codeword arrangement. The former is to discover potential prior context, the later is to effectively train a prior context planning network to select appropriate prior context for a given context. These mechanisms are essential for instantiating our model and achieving superior performance. The experimental results show that PCVAE outperforms strong baselines significantly and further analyses demonstrate the effectiveness of our proposed methods.
References


