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 **V**ariational **A**uto**E**ncoder (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 **P**rior **C**ontext VAE (PCVAE), a hierarchical VAE that learns prior context from data automatically for dialogue generation. Meanwhile, we design **A**ctive **C**odeword **T**ransport (ACT) to help the model actively discover potential prior context. Moreover, we propose **A**utoregressive **C**ompatible **A**rrangement (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

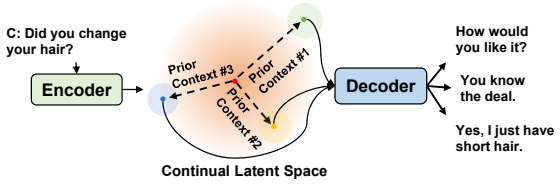

Figure 1: Illustration of prior context in the latent space of CVAE. The input context is encoded into a distribution of latent variables and the decoder can decode a sampled latent variable into a response. When sampling in latent space, it is more likely to obtain latent variables around the center 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.

*Equal Contribution.

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 hyperparameters 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 external 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 $x \in \mathcal{X}$ as a response utterance, $c \in \mathcal{C}$ 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 $y \in \mathcal{Y}$ and $z \in \mathcal{Z}$ 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, ϕ and θ 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*

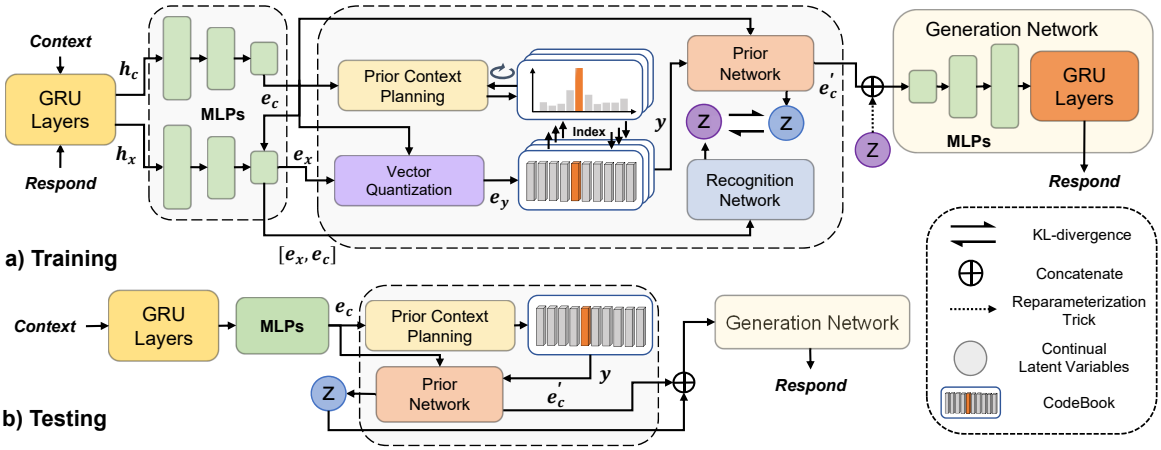

Figure 2: Overview of PCVAE.

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_D layers of **MLP**s, 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:

$$\begin{bmatrix} \mu_c \\ \log(\sigma_c^2) \\ e'_c \end{bmatrix} = \mathrm{MLP}_c([e_c, y]), \begin{bmatrix} \mu_x \\ \log(\sigma_x^2) \end{bmatrix} = \mathrm{MLP}_x([e_x, e_c])$$

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 $p_\phi(z|y,c) \sim \mathcal{N}(\mu_c, \sigma_c^2\mathbf{I})$ and $p_\theta(z|x,c) \sim \mathcal{N}(\mu_x, \sigma_x^2\mathbf{I})$. The reparameterization trick [Kingma and Welling, 2014] is used to sample a latent variable z from the $p_\theta(z|x,c)$ in training phrase and $p_\phi(z|y,c)$ in testing phrase, respectively. We employ KL divergence $\mathcal{L}_K = D_{KL}(p_\theta(z|x,c)\|p_\phi(z|y,c))$ to make $p_\phi(z|y,c)$ 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_d through N_D 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 $\mathcal{L}_G$ as the objective function of generation.

3.3 Prior Context Learning

Vector Quantization. We employ a codebook with random initialization codewords v_i and $i \in \{1,\ cdots, N_E\}$. The input e_x and e_c are transformed into an input vector for quantization as $e_y = \mathbf{MLP}([e_x, e_c])$. 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:

$$I(e_{y,j}, v_k) = \begin{cases} 1 & \text{for } k = \mathrm{argmin}_k \|e_{y,j} - v_k\|_2 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

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:

$$\mathcal{L}_{vq} = \sum_{j=1}^{N_K}\sum_{k=1}^{N_E} I(e_{y,j}, v_k)(\|\mathrm{sg}[v_k] - e_{y,j}\|_2^2 + \beta\|v_k - \mathrm{sg}[e_{y,j}]\|_2^2) \quad (2)$$

Where the stop-gradient operation sg is used since the above selecting approach is intractable. β is a weight coefficient.

Piror 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:

$$p(y|c) = \sum_x p(y|c,x)p(x) \quad (3)$$

where the $p(y|c,x)$ is the probability distribution of quantizing function I. Then we can train the **GRU** in a autoregressive

manner as follow:

$$\mathcal{L}_{pcp} = -\sum_{j}^{N_K} \log p(y_j|y_{j-1}, \cdots, y_1, c) \tag{4}$$

Autoregressive Codeword Arrangement. Intuitively, the ground-truth indexes in training are non-differentiable, which leads to an uncontrolled conditional probability distribution of indexes. It may cause unordered dependency, as we have mentioned. We propose to directly pull the $p(y|c)$ and $p(y|c,x)$ to each other as:

$$D_{KL}(p(y|c)\|p(y|c,x)) = \sum_{j=1}^{N_K} p(y_j|c) \log \frac{p(y_j|c)}{p(y_j|c,x)} \tag{5}$$

However, the gradient of the $p(y|c,x)$ is intractable, which prevents us from training it to obey autoregressive style distribution. To resolve the problem, we introduce a probability predicting network $\Psi(e_y)$ to mimic the $p(y|c,x)$ and we train Ψ by approximating the $p(y|c,x)$ in equation 1. The above steps can be summarized as follows:

$$\begin{aligned} \mathcal{L}_{aca} = &\operatorname*{argmin}_{e_y \sim \mathcal{L}} [D_{KL}(p(y|c)\|\Psi(e_y))] \\ &+ D_{KL}(\Psi(e_y)\|p(y|c,x)) \end{aligned} \tag{6}$$

Notably, introducing $\Psi(e_y)$ makes all parameters involved in generating e_y affected. Since e_y determines $p(y|c,x)$ in the forward pass, $p(y|c,x)$ will also tend to obey autoregressive style distribution, despite it being intractable.

Active Codeword Transport. The general idea of ACT is to move the e_y (target) to the unused codewords (source), and thus the currently unused codewords would be selected and used in the subsequent training. To this end, the first problem is how to select unused codewords. Intuitively, we should select as many unused codewords as possible while minimizing change to e_y. For this purpose, we assume there is a center where the currently used codewords tend to gather, and there are extensive unused codewords around it as shown in the Figure. 3. We dynamically estimate this center using moving mean predicting:

$$e_c = e_c \cdot \gamma_m + (1 - \gamma_m) \sum_{i=1}^{N_B} \frac{e_y^i}{N_B} \tag{7}$$

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 $e_r^i = e_y^i - e_c$ for each e_y. We use those direction vectors to predict the near unused codewords and define

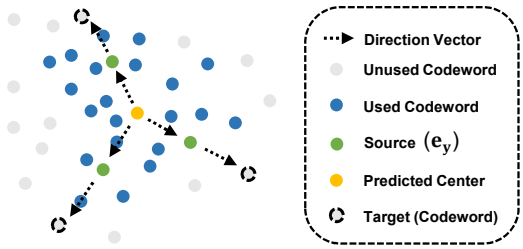

Figure 3: Illustration of active codeword transport.

them as a target set T while e_y in the same batch are as a source set S:

$$\mathcal{T} = \{\text{vq}(e_y^i + j * e_r^i)\}, \mathcal{S} = \{e_y^{i(j)}\} \tag{8}$$

where $j \in \{0, 1, \cdots, N_K\}, i \in \{0, 1, \cdots, N_B\}$, vq means the vector quantization operation as we have defined, (j) in $\mathcal{S}$ indicate additional repeated elements to balance the number of elements between $\mathcal{T}$ and $\mathcal{S}$. 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:

$$\mathcal{W}(\varphi, \nu) = \inf_{\pi \in \prod(\varphi, \nu)} \int_{\mathcal{S} \times \mathcal{T}} d(s,t) d\pi(s,t) \tag{9}$$

where the transport plans π that distributes the mass in φ to match that in ν. The ground metric $d(s,t) = \|s - t\|_2^2$ 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 $\mathcal{L}_{act} = \mathcal{W}^*(\varphi, \nu)$ 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 λ_1,λ_2 are weight factors:

$$\mathcal{L}_{total} = \mathcal{L}_G + \lambda_1 \mathcal{L}_K + \mathcal{L}_{vq} + \mathcal{L}_{pcp} + \mathcal{L}_{aca} + \lambda_2 \mathcal{L}_{act}. \tag{10}$$

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

	BLEU-1	BLEU-4	Distinct-1	Distinct-2	METEOR
MultiWoz					
Seq2Seq	0.274	0.130	0.038	0.130	0.166
CVAE	0.402	0.191	0.075	0.506	0.243
kgCVAE	0.449	0.205	0.077	0.513	0.273
SepaCVAE	0.447	0.203	0.078	0.529	0.268
DCVAE	0.451	0.214	0.076	0.511	0.261
PCVAE	**0.505**	**0.241**	**0.086**	**0.557**	**0.301**
Improvement (%)	11.89	12.61	10.83	5.22	10.26
Corncell Movie					
Seq2Seq	0.218	0.094	0.025	0.108	0.111
CVAE	0.248	0.107	0.048	0.313	0.126
SepaCVAE	0.278	0.120	0.050	0.413	0.135
DCVAE	0.265	0.115	0.052	0.464	0.142
PCVAE	**0.411**	**0.203**	**0.071**	**0.661**	**0.207**
Improvement (%)	47.67	68.55	35.87	42.59	46.14

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.

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 **MLP**s 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 γ_m used in moving mean predicting is set to 0.95. The β 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 ratio 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[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 (Corncell 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 corncell 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.

[1]kgCVAE is not tested on Corncell Movie dataset since the dialog cat is unavailable.

Model	BLEU-1	BLEU-4	Distinct-1	Distinct-2
PCVAE-None	0.251	0.107	0.049	0.307
PCVAE-VQ	0.243	0.091	0.046	0.289
PCVAE-ACT	0.267	0.118	0.054	0.322
PCVAE-ACA	0.311	0.146	0.059	0.475
PCVAE	0.411	0.202	0.071	0.661

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

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.

4.3 Qualitative Analysis

The randomly sampled responses generated by PCVAE and baseline models are shown in Table 3. The two samples are from Corncell 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 SepaCVAE 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

Context&Response (1)	C: i will have a lunch in the airport before our trip, will you see me off? R1: sorry, i can't. i have a tennis class later. R2: sure, let's have a lunch before you leave.
kgCVAE	ok, i can buy your a air tickets.
SepaCVAE	wow, sound exciting. how about your trip ?
DCVAE	oh, when will you leave?
PCVAE	sure, i will go with my friends.
Context&Response (2)	C: did you have the receipt sent to me? R1: i will text the receipt to your right now. R2: i will have them send the receipt to your email, talk to you soon.
kgCVAE	it will be ready soon.
SepaCVAE	your order is confirmed.
DCVAE	ok, your order is sent to text message.
PCVAE	a receipt will be texted to your mobile device.

Table 3: Examples of generated responses by previous methods and our model.

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 codeword 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.

N_E	Codeword usage	BLEU-1	BLEU-4	Distinct-1	Distinct-2
8	8	0.418	0.202	0.077	0.514
128	79	0.486	0.231	0.082	0.527
1024	445	0.534	0.252	0.086	0.576
2048	879	0.568	0.261	0.087	0.583
4096	1206	0.577	0.275	0.090	0.601
8192	1471	0.580	0.277	0.091	0.605

Table 4: Effect of prior context on generation performances. The codeword usage refers to the number of actually used codewords.

Metric	Standard	RS	PBT	ACT
Codeword usage	559	845	723	1471
Mean $\mathcal{L}_{vq}$	0.1521	0.1637	0.1591	0.1382

Table 5: Codeword usage and mean vector quantization loss in testing.

Metric	Ground	kgCVAE	SepaCVAE	DCVAE	PCVAE
Fluency	1.01	**3.45**	3.95	3.67	3.48
Diversity	2.10	2.85	2.59	3.15	**2.17**
Relevance	1.08	2.62	2.97	3.59	**2.24**

Table 6: Human evaluation scores of each model. Best results are presented in boldface. Note that "Ground" is the ground-truth response from the used datasets.

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

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

[Banerjee and Lavie, 2005] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[Bengio *et al.*, 2013] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *ArXiv*, abs/1308.3432, 2013.

[Bird *et al.*, 2009] Steven Bird, Ewan Klein, and Edward Loper. Natural language processing with python. https://www.nltk.org/, 2009. Accessed: 2022-5-14.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

[Csaky *et al.*, 2019] Richard Csaky, Patrik Purgai, and Gábor Recski. Improving neural conversational models with entropy-based data filtering. In *ACL*, 2019.

[Cuturi, 2013] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

[Danescu-Niculescu-Mizil and Lee, 2011] Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*, 2011.

[Dhariwal *et al.*, 2020] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *ArXiv*, abs/2005.00341, 2020.

[Dieleman *et al.*, 2018] Sander Dieleman, Aäron van den Oord, and Karen Simonyan. The challenge of realistic music generation: modelling raw audio at scale. *arXiv:1806.10474*, 2018.

[Feng *et al.*, 2020] Shaoxiong Feng, Hongshen Chen, Kan Li, and Dawei Yin. Posterior-gan: Towards informative and coherent response generation with posterior generative adversarial network. *ArXiv*, abs/2003.02020, 2020.

[Gao *et al.*, 2019] Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li, Guodong Zhou, and Shuming Shi. A discrete cvae for response generation on short-text conversation. In *EMNLP*, 2019.

[Graham, 2015] Yvette Graham. Re-evaluating automatic summarization with bleu and 192 shades of rouge. In *EMNLP*, 2015.

[Jaderberg *et al.*, 2017] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

[Lancucki *et al.*, 2020] Adrian Lancucki, Jan Chorowski, Guillaume Sanchez, Ricard Marxer, Nanxin Chen, Hans J. G. A. Dolfing, Sameer Khurana, Tanel Alumäe, and Antoine Laurent. Robust training of vector quantized bottleneck models. *IJCNN*, pages 1–7, 2020.

[Li *et al.*, 2016] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B. Dolan. A diversity-promoting objective function for neural conversation models. In *NAACL*, 2016.

[Liu *et al.*, 2020] Qian Liu, Yihong Chen, B. Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. You impress me: Dialogue generation via mutual persona perception. *ArXiv*, abs/2004.05388, 2020.

[Luong *et al.*, 2015] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *ACL*, page 311–318, USA, 2002. Association for Computational Linguistics.

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

[Sohn *et al.*, 2015] Kihyuk Sohn, Xinchen Yan, and Honglak Lee. Learning structured output representation using deep conditional generative models. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 3483–3491, Cambridge, MA, USA, 2015. MIT Press.

[Sordoni *et al.*, 2015] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and William B. Dolan. A neural network approach to context-sensitive generation of conversational responses. In *NAACL*, 2015.

[Sun *et al.*, 2021] Bin Sun, Shaoxiong Feng, Yiwei Li, Jiamou Liu, and Kan Li. Generating relevant and coherent dialogue responses using self-separated conditional variational autoencoders. In *ACL/IJCNLP*, 2021.

[van den Oord *et al.*, 2017] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NIPS*, 2017.

[Yan *et al.*, 2016] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. *ArXiv*, abs/1512.00570, 2016.

[Zang *et al.*, 2020] Xiaoxue Zang, Abhinav Rastogi, Jianguo Zhang, and Jindong Chen. Multiwoz 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. *ArXiv*, abs/2007.12720, 2020.

[Zhang *et al.*, 2018a] Hainan Zhang, Yanyan Lan, J. Guo, Jun Xu, and Xueqi Cheng. Reinforcing coherence for sequence to sequence model in dialogue generation. In *IJCAI*, 2018.

[Zhang *et al.*, 2018b] Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. Generating informative and diverse conversational responses via adversarial information maximization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[Zhao *et al.*, 2017] Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*, 2017.

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