DiffusionDialog: A Diffusion Model for Diverse Dialog Generation with Latent Space

Jianxiang Xiang[1], **Zhenhua Liu**[1], **Haodong Liu**[2],
Yin Bai[2], **Jia Cheng**[2], **Wenliang Chen**[1]*
[1]School of Computer Science and Technology, Soochow University, China
[2]Meituan
{jxxiang0720, zhliu0106}@stu.suda.edu.cn, wlchen@suda.edu.cn
{liuhaodong05, baiyin, jia.cheng.sh}@meituan.com

Abstract

In real-life conversations, the content is diverse, and there exists the one-to-many problem that requires diverse generation. Previous studies attempted to introduce discrete or Gaussian-based continuous latent variables to address the one-to-many problem, but the diversity is limited. Recently, diffusion models have made breakthroughs in computer vision, and some attempts have been made in natural language processing. In this paper, we propose DiffusionDialog, a novel approach to enhance the diversity of dialogue generation with the help of diffusion model. In our approach, we introduce continuous latent variables into the diffusion model. The problem of using latent variables in the dialog task is how to build both an effective prior of the latent space and an inferring process to obtain the proper latent given the context. By combining the encoder and latent-based diffusion model, we encode the response's latent representation in a continuous space as the prior, instead of fixed Gaussian distribution or simply discrete ones. We then infer the latent by denoising step by step with the diffusion model. The experimental results show that our model greatly enhances the diversity of dialog responses while maintaining coherence. Furthermore, in further analysis, we find that our diffusion model achieves high inference efficiency, which is the main challenge of applying diffusion models in natural language processing.

Keywords: Dialogue System, Diffusion Model, One-to-many Modeling

1. Introduction

Open-domain dialogue generation is a crucial component in dialogue systems. With the development of pre-trained language models, current models are capable of generating fluent and relevant dialogues(Radford et al., 2019; Raffel et al., 2020). However, there is still a lack of exploration in generating diverse responses, because there may be multiple appropriate responses when presented with a single context, and that's known as the one-to-many mapping problem, shown as figure 1. To model the one-to-many relationship between dialog history and response, Bao et al. (2019) introduce discrete latent variables, but the diversity of response is constrained by the categories of discrete latent variables, making it challenging to achieve fine-grained diversity generation. Sun et al. (2021) and Chen et al. (2022b) introduce continuous latent variable which can relief the problem of the discrete latent variables, but the prior of the model is limited by the inflexible prior distribution, which cannot model the distribution of the response well.

As an alternative solution of one-to-many problem, we propose the integration of a diffusion model (Ho et al., 2020), which have shown its' superiority of generating high-quality and diverse results in the fields of image and audio generation (Dhariwal and Nichol, 2021; Ramesh et al., 2022; Rombach et al., 2022; Kong et al., 2020). As for text-generation, DiffuSeq (Gong et al., 2022) uses the Diffusion-LM (Li et al., 2022) structure for sequence-to-sequence tasks in a non-autoregressive manner, and both models perform diffusion operations in the embedding space. However, there are several important drawbacks. Firstly, the inference speed of the model will be greatly limited by the context length, especially in multi-turn dialogue scenarios where time consumption can be disastrous. Secondly, these models need to be trained from scratch and cannot take advantage of pre-trained language models. Some work has

Figure 1: one to many problem in dialog generation.

*Corresponding author

LREC-COLING 2024, pages 4912–4921

also attempted to combine diffusion models with latent variable. For example, LATENTOPS (Liu et al., 2022) applies diffusion models in latent space for controllable text generation tasks, this approach involves training multiple classifiers for different control requirements, and using the corresponding classifier to guide the inference of diffusion model in order to achieve controlled generation of text. However, as a complex conditional generation task, it is difficult to train classifiers to guide the latent inference process for dialogue generation.

In this work, we propose a structure that combines a latent-based diffusion model with a pre-trained language model to address the one-to-many modeling problem in multi-turn dialogues, called **DiffusionDialog**. DiffusionDialog integrates a encoder-decoder structured pre-trained language model Bart (Lewis et al., 2019) and a latent-based (Vaswani et al., 2017) diffusion model with transformer decoder structure. It performs inference of the diffusion model in the fixed-dimensional latent space, and combines the diffusion model with the language model for specific response generation. Instead of learning to approximate the fixed prior (e.g. Gaussian distribution) of the latent variable, our diffusion model learns a more flexible prior distribution from the encoder, enabling the generation of responses with finer-grained diversity. And due to the low-dimensional nature of the latent space, our diffusion model overcomes the slow inference speed issue which is the major problem of diffusion models.

The contributions of this paper can be summarized as follows:

1. We propose a novel approach to address the one-to-many problem in dialogue using a combination of a latent-based diffusion model and a pre-trained language model.
2. To the best of our knowledge, our work is the first to apply a latent diffusion model to dialog generation. By reasoning in the latent space, the inference efficiency of our diffusion model is significantly improved.
3. Through comparative experiments, we demonstrate the effectiveness of our model, which can generate responses that are rich in diversity while ensuring fluency and coherence.

2. Background

2.1. Dialog Generation with Latent Variable

The objective of dialog system is to estimate the conditional distribution $p(x|c)$. Let $d = [u_1, ..., u_k]$ denote a dialogue comprising of k utterances. Each utterance is represented by $u_i = [w_1, ..., w_{|ui|}]$, where w_n refers to the n-th word in u_i. Additionally, we define $c = [u_1, ..., u_{k-1}]$ as the dialogue context, which constitutes the $k-1$ historical utterances, and $x = u_k$ as the response, which denotes the next utterance in the dialogue.

Finding a direct connection between the discrete token sequences x and c can be challenging. To address this issue, we propose the use of a continuous latent variable z, which serves as a high-level representation of the response. In this two-step response generation process, we first sample a latent variable z from a distribution $p_\theta(z|c)$ that resides in a latent space $\mathcal{Z}$. Subsequently, we decode the response x from z and c as $p_\theta(x|z,c)$.And this process can be estimated as

$$p_\theta(x|c) = \int_z p_\theta(z|c)p_\theta(x|z,c)d_z. \tag{1}$$

Since the optimal z is intractable, we optimize the posterior distribution of z as $q_\phi(z|x)$ considering the x. And we approximate the posterior with the prior distribution $p_\theta(z|c)$,

$$\begin{aligned} &\log p_\theta(x|c) = \log \textstyle\int_z q_\phi(z|x)p_\theta(x|z,c) \\ &\geq E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z,c)] \\ &-KL(q_\phi(z|x), p_\theta(z|c)). \end{aligned} \tag{2}$$

2.2. Diffusion Model in Latent Space

Diffusion model is designed to operate in fixed and continuous domain, consisting forward and reverse processes. In this work, we perform forward and reverse process in learned latent space representing the high-level semantic of response. Suppose posterior as $z_0 \sim q_\phi(z|x)$, in the forward process, z_0 is corrupted with standard Gaussian noise in large amount of step, forming a Markov chain of $z_0, z_1, ..., z_T$, with $z_T \sim \mathcal{N}(0, I)$:

$$q(z_t|z_{t-1}) = \mathcal{N}(z_t; \sqrt{1-\beta_t}z_{t-1}, \beta_t I), \tag{3}$$

where $\beta_t \in (0,1)$ controls the scale of the noise in a single step.

In the reverse progress, diffusion model learn to reconstruct z_0 from z_T by learning $p_\theta(z_{t-1}|z_t) = \mathcal{N}(z_{t-1}; \mu_\theta(z_t,t), \Sigma_\theta(z_t,t))$, Since the $q(z_{t-1}|z_t, z_0)$ has a closed form,the canonical objective is the variational lower bound of $\log p_\theta(z_0)$,

$$\begin{aligned} &\mathcal{L}_{\text{vlb}} = \mathbb{E}_q \left[D_{\text{KL}} \left(q\left(z_T \mid z_0\right) \| p_\theta\left(z_T\right)\right)\right] \\ &+\mathbb{E}_q \left[\textstyle\sum_{t=2}^{T} D_{\text{KL}} \left(q\left(z_{t-1} \mid z_t, z_0\right) \| p_\theta\left(z_{t-1} \mid z_t, t\right)\right)\right] \\ &- \log p_\theta\left(z_0 \mid z_1\right). \end{aligned} \tag{4}$$

To promote stability in training, we take advantage of the simplified objective proposed by Ho et al. as $\mathcal{L}_{\text{simple}}$,

$$\mathcal{L}_{\text{simple}}(z_0) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{q(z_t|z_0)} \| \mu_\theta(z_t,t) - \hat{\mu}\left(z_t, z_0\right) \|^2, \tag{5}$$

where $\hat{\mu}(z_t, z_0)$ refers to $q(z_{t-1}|z_t, z_0)$, and $\mu_\theta(z_t, z_0)$ is learned by diffusion model.

3. DiffusionDialog

3.1. Model Architecture

Our model introduces a hierarchical generation process with latent variable. Firstly it obtains latent variable reflecting the semantic of response from the context and then generate the response considering the latent variable and the context (Equation 1), thus the response generation involves three key components: the dialogue context c, the response r, and the latent variable z.

We combines encoder-decoder structured pre-trained language model Bart with a latent-based diffusion model to handle the two-stage generation, the figure 2 illustrates our model, and we explain our model by illustrating the function of each part of the model.

3.1.1. Bart Encoder

The bart encoder plays a dual role in our model, encoding both the contex and the latent variables.

For context, following the PLATO, in addition to token and position embeddings, it also incorporates turn embeddings to align with the context turn number, and role embeddings to align with the speaker's role. As a result, the final embedding input of the context is the sum of corresponding token, turn, role, and position embeddings.

For latent variables, since the priors are untraceable, bart encoder learns the priors of the latent variable $q_\phi(z|x)$ which represents the high-level semantic information about the response.

To connect the latent space, we concatenate a special token in front of the response to encode the semantic information of the response. We refer to this special token as latent toke. Therefore, the input format for latent variable encoding is $[l, w_1^x, w_2^x ..., w_n^x]$, n refers to the length of response x.

We append a multilayer perceptron to obtain a representation of the posterior distribution $z_0 \sim q_\phi(z|x)$:

$$z_0 = MLP(h_{[L]}), \tag{6}$$

where $h_{[L]} \in \mathbb{R}^d$ refers to the final hidden state of the latent token.

3.1.2. Latent Diffusion Denoiser

After obtaining z_0 from the bart encoder, we sample a time step $t \in [1, T]$ uniformly and add noise to the latent variable according to Equation 3, resulting in a noised latent z_t. The latent diffusion denoiser is trained to denoise the latent. It adopts the structure of a transformer decoder, taking the noised latent variable as inputs and incorporates the context hidden state with cross-attention mechanism, and a timestep embedding is also added before the first Transformer block to inform the model of the current timestep,

$$\tilde{z_0} = Denoiser(z_t, e_t, h_c), \tag{7}$$

where e_t refers to the embedding of the timestep t. Since the context hidden state is fixed during inference, the inference time required for the diffusion model is short.

3.1.3. Bart Decoder

To guide the response generation of the decoder using latent variables, we adopt the memory scheme from **OPTIMUS** (Li et al., 2020). Specifically, we project the latent variable z as a key-value pair and concatenate them to the left of the token hidden state to introduce the latent variable into the decoder.

$$H^{(l+1)} = MultiHead(H^{(l)}, h_{Mem}^{(l)} \oplus H^{(l)}, h_{Mem}^{(l)} \oplus H^{(l)}),$$

where $H^{(l)}$ refers to the token hidden state of the l-th layer, and $h_{Mem}^{(l)}$ is calculated as:

$$h_{Mem}^{(l)} = \begin{bmatrix} z_{key} \\ z_{value} \end{bmatrix} = W_M^l \, z, \tag{8}$$

where $W_M^l \in \mathbb{R}^{d \times 2d}$ is a weight matrix.

3.2. Training

During our training process for dialogue generation, we utilize three different loss functions: negative log-likelihood (NLL) loss, bag-of-words (BOW) loss, and latent denoising (LD) loss. Detailed descriptions will be provided in this section.

3.2.1. Response semantic capture

To enable the latent variable to capture the overall semantic information of the response, we adopt the bag-of-words (BOW)(Zhao et al., 2017) loss, which is used to enable the latent variable to predict the tokens in the response in a non-autoregressive manner.

$$\begin{aligned} \mathcal{L}_{BOW} &= -\mathbb{E}_{z_0 \sim q_\phi(z|r)} \sum_{n=1}^{N} \log p(r_t|z_0) \\ &= -\mathbb{E}_{z_0 \sim q_\phi(z|r)} \sum_{n=1}^{N} \log \frac{e^{f_{r_n}}}{\sum_{v \in V} e^{f_v}} \, . \end{aligned} \tag{9}$$

The symbol V refers to the entire vocabulary. The function f attempts to non-autoregressively predict the words that make up the target response.

$$f = \text{softmax}\,(W_2 h_z + b_2) \in \mathbb{R}^{|V|}. \tag{10}$$

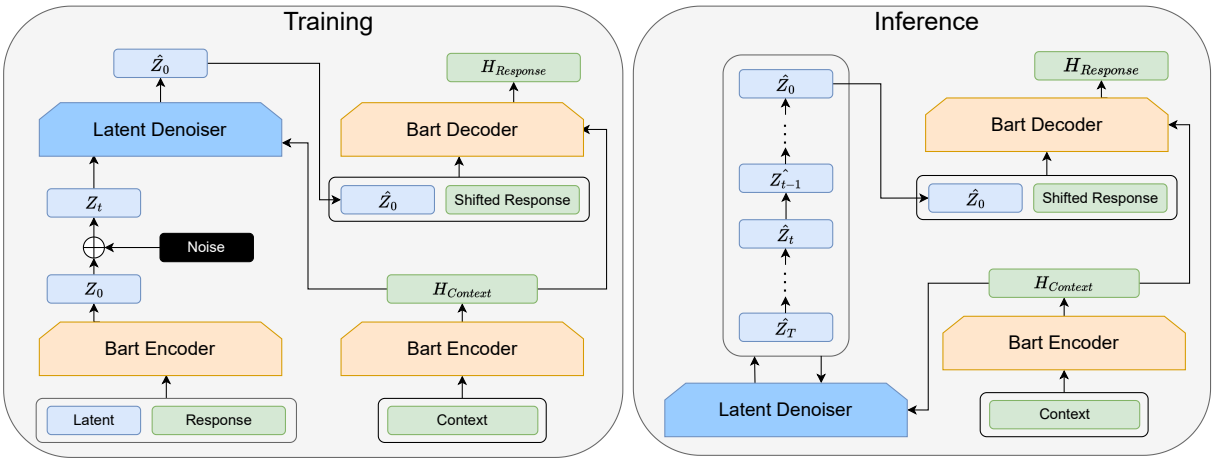

Figure 2: frame work of DiffusionDialog.

In the given equation, h_z represents the hidden state of the latent variable, while $|V|$ denotes the size of the vocabulary. The estimated probability of word r_n is denoted by f_{r_n}. BOW loss disregards the word order and compels the latent variable to capture the overall information of the target response.

3.2.2. Latent Denoising

For each training step, we sample a time step t and obtain z_t referring to Equation 3. To better capture the semantic information of the latent variables, our diffusion model predicts z_0 directly instead of z_{t-1} given z_t, denoted as $\mathcal{L}_{z_0\text{-simple}}$, a variant of $\mathcal{L}_{\text{simple}}$ in Equation 5:

$$\mathcal{L}_{z_0\text{-simple}}\ (z_0) = \sum_{t=1}^{T} \mathbb{E}_{z_t} \left\| p\left(z_t, c, t\right) - z_0 \right\|^2 . \quad (11)$$

where our latent diffusion denoiser $p\left(z_t, h_c, t\right)$ predicts z_0 directly.

Thus at each time step, the loss of latent denoising is:

$$\mathcal{L}_{LD} = \left\| p\left(z_t, t, c\right) - z_0 \right\|^2. \quad (12)$$

3.2.3. Response Generation

In our model, the response is generated by conditioning on both the latent variable and the context. To train the response generation we adopt the commonly used NLL loss,

$$\begin{aligned} \mathcal{L}_{NLL} &= -\mathbb{E}_{\tilde{z_0} \sim p(z|c,z_t,t)} \log p(r \mid c, \tilde{z_0}) \\ &= -\mathbb{E}_{\tilde{z_0} \sim p(z|c,z_t,t)} \sum_{n=1}^{N} \log p\left(r_t \mid c, \tilde{z_0}, r_{<t}\right). \end{aligned} \quad (13)$$

Note that $\tilde{z_0}$ is the posterior distribution predicted by the latent diffusion denoiser, we adopt this approach to reduce the gap between training and inference. In order to optimize the NLL loss, the denoiser's prediction needs to not only be close to the prior distribution z_0 in the spatial domain, but also approximate the response in the semantic domain.

In summary, our model aims to minimize the overall objective function, which is defined as the integrated loss:

$$\mathcal{L} = \mathcal{L}_{NLL} + \mathcal{L}_{BOW} + \mathcal{L}_{LD}. \quad (14)$$

3.3. Inference

The inference in our model consists of two stages. Firstly, starting from a Gaussian noise, the latent diffusion denoiser performs multiple rounds of inference to denoise the latent variable and obtain the final semantic representation z_0, conditioned on the hidden state of the context which is encoded by the encoder. Then the response generator generates the final response in an auto-regressive manner, conditioned on both z_0 and the context hidden state.

For ease of displaying the training and inference process of our model, we outline our approach in Figure 3.

4. Experiments

4.1. Experimental Setup

4.1.1. Datasets and Evaluation

Following PLATO(Bao et al., 2019),we evaluate the performance of our model on two commonly used public dialog datasets.

DailyDialog(Li et al., 2017) is a high-quality conversational dataset that primarily focuses on daily dialogues.

Persona-Chat(Zhang et al., 2018) is sourced from authentic conversations between human annotators who are randomly matched and assigned

Algorithm 1 Training

Input: a dialog corpus $\mathcal{D}$=$\{(c_i, r_i)\}_{i=1}^{|\mathcal{D}|}$
repeat
sample context and response (c, r) from $\mathcal{D}$
$h_c = \text{Encoder}(c)$
$z_0 \sim q_\phi(z|r) = \text{Encoder}([l; x])[0]$
$\mathcal{L}_{BOW} = -\sum_{n=1}^{N} \log p(r_t|z_0)$
$t \sim \text{Uniform}(\{1, \ldots, T\})$
$\epsilon \sim \mathcal{N}(0, I)$
$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$
$\tilde{z_0} = \text{Denoiser}(z_t, h_c, t)$
$\mathcal{L}_{LD} = -\|\tilde{z_0} - z_0\|^2$
$\mathcal{L}_{NLL} = -\sum_{n=1}^{N} \log p\left(r_t \mid c, \tilde{z_0}, r_{<t}\right)$
Take gradient descent step on
$\nabla_\theta[\mathcal{L} = \mathcal{L}_{LD} + \mathcal{L}_{NLL} + \mathcal{L}_{BOW}]$
until converged

Algorithm 2 Inference

$h_c = \text{Encoder}(c)$
$\tilde{z_T} \sim \mathcal{N}(0, I)$
for $t = T, \ldots, 1$ **do**
  $\tilde{z_0} = \text{Denoiser}(\tilde{z_t}, h_c, t)$
  $\epsilon \sim \mathcal{N}(0, I)$
  $\tilde{z_{t-1}} = \sqrt{\bar{\alpha}_{t-1}} \tilde{z_0} + \sqrt{1 - \bar{\alpha}_{t-1}}\epsilon$
end for
response $\tilde{r} = Decoder(\tilde{z_0}, h_c)$

Figure 3: The training and inference algorithm of DiffusionDialog.

	DailyDialog	PersonaChat
train	76052 samples \	122499 samples \
dev	7069 samples 12.1% overlap	14602 samples \
test	6740 samples 13.0% overlap	14056 samples \

Table 1: Summary of datasets used in the experiments, overlap means percentage of data leaks.

a given persona information. Paired annotators engage in natural conversation and attempt to know each other better throughout the dialogue.

Table 1 summarizes the descriptions and statistics of these datasets. In DailyDialog, 12.1% of the development set and 13.0% of the test set appeared in the training set, indicating the presence of data leakage, while no such issue is observed in PersonaChat.

For evaluation, we mainly evaluate our model on fluency and diversity. We adopt the same metrics used in PLATO, which is widely used:

BLEU-1/2(Papineni et al., 2002) which measures the coherence of generated response to the given context by calculating the 1/2-grams overlapping between the generated response and references.

Distinct-1/2(Li et al., 2015) which measures the diversity of generated response by calculating the number of unique 1/2-grams divided by the total number of generated words.

4.1.2. Compared Baselines

In our experiments, the following models were selected as our baselines.

Seq2Seq(Vinyals and Le, 2015) is a sequence-to-sequence model with attention. **IVAE$_{MI}$**(Fang et al., 2019) is also a sequence-to-sequence model with implicit deep latent variable that employs a Variational Autoencoder to improve the quality of latent representations and generate diverse responses. **LIC**(Golovanov et al., 2019) is a transformer-based generative model fine-tuned on GPT, which has demonstrated remarkable performance in the ConvAI2 challenge. **PLATO**(Bao et al., 2019) employs a discrete latent variable to address the one-to-many problem, showing high performance in both response fluency and diversity. **DialogVED**(Chen et al., 2022b) introduces continuous latent variables with VAE model into the enhanced encoder-decoder pre-training framework to increase the relevance and diversity of responses. Both of PLATO and DialogVED address the one-to-many problem in dialogue tasks and are the main objects of comparison in our study.

To accurately evaluate the impact of our latent variable with diffusion model, we compare our model to the version without the latent.

4.1.3. Model Configuration

Our model consists of two parts: one is a encoder-decoder structure Transformer model Bart-base, which is composed of 6 layers of encoder and 6 layers of decoder. The other part of our model is a latent denoiser, which is a structure of 6 layers of transformer decoder with latent token embedding and 128-dimensional time step embedding. Our diffusion steps $T = 2,000$ and noise schedule is square-root schedule. Our maximum context squence length is 256 and our maximum response sequence length 128, The model uses the BPE tokenization(Sennrich et al., 2015) which is commonly used.

During training, We use Adamw optimizer(Loshchilov and Hutter, 2017) with a learning rate of 1×10^{-4}, the batch size is 128, We also adopt a warmup strategy where we linearly increase the learning rate from initial learning rate 1×10^{-7}, the totall training steps for DailyDialog is 10000, and for PersonaChat is 20000. We select the checkpoint with the lowest validation loss for

Model	DailyDialog			
	BLEU-1	BLEU-2	Distinct-1	Distinct-2
Seq2Seq (Vinyals and Le, 2015)	0.336	0.238	0.030	0.128
iVAE_MI (Fang et al., 2019)	0.309	0.249	0.029	0.250
PLATO w/o latent† (Bao et al., 2019)	0.405	0.322	0.046	0.246
PLATO† (Bao et al., 2019)	0.397	0.311	0.054	0.291
DialogVED‡ (Chen et al., 2022b)	0.481	0.421	0.042	0.232
Our w/o Latent	**0.406**	**0.371**	0.046	0.217
Our Method	0.348	0.318	**0.072**	**0.372**
Our Method Upper Bound	0.471	0.424	0.063	0.348
Model	PersonaChat			
	BLEU-1	BLEU-2	Distinct-1	Distinct-2
Seq2Seq (Vinyals and Le, 2015)	0.448	0.353	0.004	0.016
LIC (Golovanov et al., 2019)	0.405	0.320	0.019	0.113
PLATO w/o latent† (Bao et al., 2019)	**0.458**	0.357	0.012	0.064
PLATO† (Bao et al., 2019)	0.406	0.315	0.021	0.121
DialogVED‡ (Chen et al., 2022b)	0.482	0.399	0.015	0.094
Our w/o Latent	0.410	**0.359**	0.013	0.056
Our Method	0.383	0.329	**0.031**	**0.177**
Our Method Upper Bound	0.472	0.411	0.026	0.166

Table 2: Experimental results on DailyDialog and PersonaChat with automatic evaluations. PLATO and DialogVED is pretrained with large dialog corpus, and DialogVED is based on Bart Large. The best values are underlined, and the best results with base-PLMs is written in bold.

context	[P1]It's a lovely day out today, isn't it? [P2]It's beautiful. Enjoy it while it lasts. It's supposed to get cold tomorrow. [P1]What's the weather forecast for tomorrow?
our w/o Latent	It's supposed to snow
our Method	Cloudy in the morning and overcast in the afternoon
	The weatherman says it's going to snow tomorrow
	It's supposed to snow in the morning and in the afternoon. Is that possible?
	The weatherman says it's supposed to snow all day.
	The weatherman says a storm is coming.
	Yep. It's supposed to snow in the morning.
	Cloudy with a chance of showers.
	Dreadful. It's supposed to snow tomorrow.
context	[P1] Good morning, sir. Is there a bank near here? [P2] There is one. 5 blocks away from here? [P3] Well, that's too far. Can you change some money for me?
our w/o Latent	Yes, Please.
our Method	Yes, Please wait for a moment.
	Yes, madam. I am sure you can. The interest rate is very high.
	What's your account number and your PIN number?
	Yes, Madam. Can I help you?
	How can I help you?
	Yes, certainly.My name is John Sandals.

Table 3: Examples of response generation with our model.

inference. The experiment is carried out on one single 1080Ti GPU.

4.2. Main Results

Table 2 summarizes the experimental results on Persona-Chat and Daily Dialog.

Note that both the PLATO and DialogVED models have been pre-trained on a large corpus of dialogue data. Additionally, the DialogVED model is based on Bart-large(0.47B), which gives it a significant advantage in terms of the number of parameters compared to our model(0.21B).

The PLATO model uses discrete latent variables,

while DialogVED uses a VAE-based continues latent variable, We compared our model with these two models to demonstrate the advantages of handling latent variables using the diffusion model.

To more effectively evaluate the impact of our latent discrete variable, we also conducted a comparison with the version that does not include a latent variable (referred to as 'Our w/o Latent'). It accepts the same context embedding input as our model, and also using Bart-base as it's backbone, sharing the same training settings as our method with latent variables.

The last line represents the upper bound of our model, we generate 10 different latent variable for the same context and use them to generate corresponding responses as candidates. We select the candidate with the highest overlap with the reference, i.e., the highest Bleu-1 score, as our final result.

DiffusionDialog represents the result of our model with one candidate, and all models use Beam Search for decoding, with a beam size of 5. Our diffusion model utilizes the DDIM(Song et al., 2020a) acceleration technique during inference, with a sampling time step of 50 for the purpose of performance and time efficiency, the results under different inference time steps will be discussed in detail in the later section.

As shown in Table 2, our model achieves very high results on the Dist metric. However, compared to the version without latent variable, there is a certain decrease in the Bleu metric, but our model still achieves competitive results in models that have not been pre-trained on dialogue data. The performance of our method without latent variable on the Bleu metric is similar to that of PLATO, which we attribute to the performance of the Bart pre-training model, benefiting from the encoder-decoder architecture and generative pre-training. Compared to DialogVED, which has the same architecture as ours but has more parameters and is pre-trained on dialogue data, our model's Bleu score is much lower.

We notice that the drop in Bleu score due to the introduction of latent variables is smaller on PersonaChat than on DailyDialog. Combining with the statistics in Table 1, we can infer that the data leakage in the test set and development set of DailyDialog penalizes the diversity of generated results.

The improvement in our model's Dist value compared to PLATO and DialogVED indicates that introducing latent variables based on the diffusion model can more effectively improve the diversity of generated responses compared to discrete latent variables and continuous latent variables based on a fixed Gaussian prior. Meanwhile, our experiments on the upper bound of our model's performance also demonstrate the potential of our model.

Model	speed
Our w/o Latent	0.068 s/sample
PLATO	25.813 s/sample
DialogVED	0.076 s/sample
Our Method-10$^\diamond$	0.072 s/sample
Our Method-100$^\diamond$	0.189 s/sample
Our Method-1000$^\diamond$	1.500 s/sample
DiffuSeq-10$^\diamond$	0.384 s/sample
DiffuSeq-100$^\diamond$	3.810 s/sample
DiffuSeq-1000$^\diamond$	38.246 s/sample

Table 4: Comparison of inference speed among models, models with symbol$^\diamond$ utilize diffusion model.

4.3. Discussions

4.3.1. Case Analysis

In order to further demonstrate the generative capabilities of our model, we provide some generated responses in Table 3. The table illustrates five of these responses, which showcase the model's ability to generate diverse, relevant, and fluent response.

4.3.2. Inference Speed

We compare the inference speed of our model with DialogVED, PLATO, DiffuSeq, and the results are shown in Table 4.

Note that the framwork used for inference among the models are different. PLATO was run on paddlepaddle, DialogVED was run on fairseq, DiffuSeq and our model was just run on pytorch. The number following DiffusionDialog and DiffuSeq represents the number of time steps used for inference.

As the table shows, due to the absence of inference on latent variables, the inference time for our method without latent is very short, and DiffusionDialog is comparable When the number of inference time steps is 10, which demonstrates the high efficiency of our model's inference.

DiffuSeq, like DiffusionDialog, utilizes a diffusion model for text generation. We compare DiffusionDialog with the DiffuSeq model to demonstrate the advantage in inference speed of our diffusion model.

To ensure fairness in comparison, we set the maximum input length of these models to 256. At inference time steps of 10, 100, and 1000, our model required less time for inference than the DiffuSeq model. Moreover, as the number of infer-

Steps	DailyDialog				PersonaChat			
	Bleu-1	Bleu-2	Dist-1	Dist-2	Bleu-1	Bleu-2	Dist-1	Dist-2
10	0.350	0.318	0.071	0.369	0.385	**0.331**	0.031	0.172
100	0.348	0.319	0.073	0.372	0.380	0.328	0.031	0.169
1000	**0.352**	**0.327**	**0.074**	**0.373**	**0.389**	**0.331**	**0.032**	**0.181**

Table 5: Impact of number of sampling steps on performance.

ence steps increased, our model's speed advantage grew.

PLATO introduces discrete latent variables, which require generating all candidate responses based on these latent variables, thereby requiring a considerable amount of time. In this comparative experiment, we used 20 discrete latent variables ($K = 20$), the same as the official version provided. For DialogVED, we used their large version with $P = 64$.

4.3.3. Sampling Steps

During inference, the diffusion model requires a large number of sampling steps, which is a significant bottleneck for the inference speed. And prior work, e.g., DiffuSeq(Gong et al., 2022) suffers from a significant drop in generation quality when reducing the sampling steps. In order to investigate the performance of our model on the test dataset under different numbers of sampling steps, we present the results in Table 5.

As shown in the table, our method achieves competitive results with as few as 10 on both dataset. It should be noted that as the number of sampling steps increases, the performance of our model on PersonaChat, as measured by the BLEU metric, first decreases and then improves. At 1000 time steps, all metrics reach their peak, but the difference between 1000 and 10 steps is not significant.

5. Related Work

5.1. One-to-many Modeling

The existence of multiple suitable responses for a given context is referred to as the one-to-many problem. Some works introduce latent variable to model the relationship, CVAE(Zhao et al., 2017) utilizes Gaussian distribution to capture variations in responses at the discourse level, since a simple distribution over the latent variables has a lack of granularity in modeling the semantic information of the responses, DialogWAE(Gu et al., 2018) develop a Gaussian mixture prior network to enrich the latent space, instead of the single Gaussian prior of VAE. iVAE$_{\text{MI}}$(Fang et al., 2019) address the challenge with implicit learning. DialogVED(Chen et al., 2022b) incorporates continuous latent variables into an enhanced encoder-decoder pre-training framework to increase the relevance and diversity of responses. PLATO(Bao et al., 2019) introduces discrete latent variables to tackle the inherent one-to-many mapping problem in response generation. Both of PLATO and DialogVED are pretrained with large dialog corpus, providing a strong baseline for one-to-many modeling.

5.2. Diffusion Models for Sequence Learning

Since Diffusion model(Dhariwal and Nichol, 2021; Song et al., 2020b) has achieved breakthroughs in the field of image processing. There have been many works attempting to apply diffusion models to the field of natural language processing. Considering the discrete nature of texts, D3PM(Austin et al., 2021) introduce Markov transition matrices to diffuse the source data instead of Gaussian noise, Analog Bits(Chen et al., 2022a) represents discrete data as binary bits, and then training a continuous diffusion model to model these bits as real numbers. Diffusion-LM(Li et al., 2022) develop a non-autoregressive language model based on continuous diffusions with an embedding function and rounding process, iteratively denoises a sequence of Gaussian vectors into words. DiffuSeq(Gong et al., 2022) propose a diffusion model designed for sequence-to-sequence text generation tasks utilizing encoder-only Transformers. And SeqDiffuSeq(Yuan et al., 2022) approach sequence-to-sequence text generation with Encoder-Decoder Transformers. LD4LG(Lovelace et al., 2022) learn the continuous diffusion models in the latent space of a pre-trained encoder-decoder model.

6. Conclusion

This paper presents DiffusionDialog, which combines an encoder-decoder structured pre-trained language model with diffusion model. By utilizing the diffusion model to learn the latent space and infer the latent by denoising step by step, we greatly enhance the diversity of dialog response while keeping the coherence and achieving high inference efficiency. As experimental results shows, our model has achieved a over 50% increase in the dist metric and accelerate inference speed over

50 times compared to the DiffuSeq model. Overall, this work provides a novel idea for applying diffusion model into natural language processing.

7. Limitations

As shown in the experiments, the accuracy of our model is not yet high enough. We identified two main reasons for this: 1) we have not conducted extensive pre-training, and 2) the structure and training methods of the model are not yet optimal. We will attempt to address these issues in future work.

8. Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62261160648 and 62376177). This work is also supported by Collaborative Innovation Center of Novel Software Technology and Industrialization, the Priority Academic Program Development of Jiangsu Higher Education Institutions, and the joint research project of Meituan and Soochow University. We would also like to thank the anonymous reviewers for their insightful and valuable comments.

9. References

Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K Marks, Chiori Hori, Peter Anderson, et al. 2019. Audio visual scene-aware dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7558–7567.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993.

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2019. Plato: Pre-trained dialogue generation model with discrete latent variable. *arXiv preprint arXiv:1910.07931*.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. 2022a. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*.

Wei Chen, Yeyun Gong, Song Wang, Bolun Yao, Weizhen Qi, Zhongyu Wei, Xiaowu Hu, Bartuer Zhou, Yi Mao, Weizhu Chen, et al. 2022b. Dialogved: A pre-trained latent variable encoder-decoder model for dialog response generation. *arXiv preprint arXiv:2204.13031*.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794.

Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. 2019. Implicit deep latent variable models for text generation. *arXiv preprint arXiv:1908.11527*.

Sergey Golovanov, Rauf Kurbanov, Sergey Nikolenko, Kyryl Truskovskyi, Alexander Tselousov, and Thomas Wolf. 2019. Large-scale transfer learning for natural language generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6058.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*.

Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. 2018. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. *arXiv preprint arXiv:1805.12352*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2020. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. *arXiv preprint arXiv:2004.04092*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li, and Zhiting Hu. 2022. Composable text controls in latent space with odes. *arXiv preprint arXiv:2208.00638*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Weinberger. 2022. Latent diffusion for language generation. *arXiv preprint arXiv:2212.09462*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.

Ramon Sanabria, Shruti Palaskar, and Florian Metze. 2019. Cmu sinbad's submission for the dstc7 avsd challenge. In *DSTC7 at AAAI2019 workshop*, volume 6.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Iulian 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020a. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020b. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Bin Sun, Shaoxiong Feng, Yiwei Li, Jiamou Liu, and Kan Li. 2021. Generating relevant and coherent dialogue responses using self-separated conditional variational autoencoders. *arXiv preprint arXiv:2106.03410*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2022. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *arXiv preprint arXiv:2212.10325*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.

Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93.