Diffusion-NAT: Self-Prompting Discrete Diffusion for Non-Autoregressive Text Generation

Anonymous ACL submission

Abstract

Recently, continuous diffusion models (CDM) have been introduced into non-autoregressive (NAR) text-to-text generation. However, the discrete nature of text increases the difficulty of CDM to generate coherent and fluent texts, and also causes the incompatibility problem between CDM and advanced NLP techniques, especially the popular pre-trained language models (PLMs). To solve it, we propose Diffusion-NAT, which introduces discrete diffusion models (DDM) into NAR text-to-text generation and integrates BART to improve the performance. By revising the decoding process of BART and the typical settings of DDM, we unify the inference process of BART and the denoising process of DDM into the same NAR masked tokens recovering task. In this way, DDM can rely on BART to perform denoising, which can benefit from both the rich pre-learned knowledge of BART and the iterative refining paradigm of DDM. Besides, we also propose the iterative self-prompting strategy to further improve the generation quality. Experimental results on 7 datasets show that our approach can outperform competitive NAR methods, and even surpass autoregressive methods. Our code and data will be publicly released.

Model	Type	PLMs	Cost	NAR	T2T
D3PM	Dis.	×	Low	✓	×
Diffusion-LM	Con.	×	Low	✓	×
SED	Con.	×	Low	✓	×
SSD-LM	Con.	✓	High	✓	×
DiffusionBERT	Dis.	✓	High	✓	×
LD4LG	Con.	✓	Low	×	×
DiffuSeq	Con.	×	Low	✓	✓
SeqDiffuSeq	Con.	×	Low	✓	✓
GENIE	Con.	×	High	✓	✓
Difformer	Con.	×	Low	✓	✓
Ours	Dis.	✓	Low	✓	✓

Table 1: A comparison of existing diffusion methods for text generation. **Dis.** and **Con.** refer to discrete and continuous diffusion. **PLMs**, **Cost**, **NAR** and **T2T** denote using PLMs, Training Cost, Non-AutoRegressive model and Text-to-Text generation, respectively.

1 Introduction

Text-to-text generation (Sutskever et al., 2014; Vaswani et al., 2017) is an essential task in natural language processing, which aims to generate human-like texts satisfying the task demand. To efficiently generate high-quality texts, non-autoregressive (NAR) models (Gu et al., 2018; Lee et al., 2018) are widely explored for text-to-text generation by predicting all tokens in the target text simultaneously, having a lower inference latency.

Despite the efficiency, the generation accuracy of NAR models generally underperform autoregressive (AR) models with the token-by-token generation, since parallel token prediction cannot effectively capture the dependency among the tokens. To enhance the generation quality, a variety of techniques have been proposed for NAR models, with either improved architectures (Qian et al., 2021) or training methods (Qi et al., 2021). More recently, inspired by the success of diffusion models in computer vision (Ho et al., 2020; Dhariwal and Nichol, 2021), they have been introduced to improve NAR models for text-to-text generation (Chen et al., 2023; Floto et al., 2023; Lyu et al., 2023; Mahabadi et al., 2023). As shown in Table 1, these studies typically adopt the continuous diffusion method on the latent space of token embeddings in the NAR manner, and iteratively refine all the target token embeddings via a parameterized denoising process.

However, these attempts are highly limited by the discrete nature of text, and thus it is necessary to incorporate special strategies to adapt continuous diffusion models for text generation. Typically, they rely on an additional rounding step (Li et al., 2022b) to map the generated embeddings into tokens, and add corresponding loss during training. However, the added step and training loss would burden the diffusion models, causing them hungry for more training steps and data to capture the mapping relation between input and output. Although large-scale pre-trained language models (PLMs) (Devlin et al., 2019; Lewis et al., 2020) seem to be a promising solution to alleviate this hunger problem, due to the large model discrepancy, it is difficult to use existing PLMs for improving the text generation models when integrating with continuous diffusion models, even leading to performance degradation (Li et al., 2022b).

To address these issues, we aim to develop a more effective approach to integrating diffusion models and PLMs for NAR text-to-text generation. Instead of using continuous diffusion, we utilize discrete diffusion (Austin et al., 2021; Gu et al., 2022) for text generation, which performs denoising on discrete states (*e.g.,* vocabulary) to recover the original tokens. It is more suitable for modeling discrete text data, making it feasible to develop more *unified and compatible* solutions to integrate diffusion models and well-trained PLMs for improving NAR text generation. However, both discrete diffusion models and PLMs neither naturally fit with each other nor the NAR text-to-text generation manner, making it hard to directly combine them for improving the NAR generation quality.

In this paper, we propose **Diffusion-NAT**, a self-prompting discrete diffusion model using PLMs for NAR text-to-text generation. The core contribution lies in that we unify the *inference process* of PLMs and *denoising process* of discrete diffusion models into the same ***masked token recovering task*** in the NAR manner. In this way, PLMs can play the role of the parameterized denoiser in discrete diffusion models, hence we can combine the merits of both diffusion models (*using iterative refining generation*) and PLMs (*with rich semantic knowledge*) for improving NAR text generation. Specifically, we select the Seq2Seq PLM, BART (Lewis et al., 2020) as our backbone by revising its decoding process into the NAR masked tokens recovering task. Then, we adjust the typical discrete diffusion method to better fit the PLM by adding mask tokens as noise, revising the learning objective and removing the time step embeddings. Further, as our approach performs the denoising process fully based on the PLM, we devise an iterative self-prompting strategy to guide the PLM performing multi-turn deliberation and refinement on the intermediate generated results, to enhance the quality of the final output.

To demonstrate the effectiveness of our approach, we conduct extensive experiments on seven text-to-text generation datasets. Experimental results show that our approach can outperform competitive NAR text generation methods, *e.g.,* improving the best NAR models by +2.48 BLEU-2 on PersonaChat, +4.33 Distinct-2 on DailyDialog.Our approach even surpasses state-of-the-art autoregressive PLMs, *e.g.,* Ours (62.68) *v.s.* BART (49.59) on BLEU-2 in DailyDialog, and Our (44.2) *v.s.* BART (38.3) on ROUGE-L in MSNews.Besides, our approach also supports DDIM (Song et al., 2021a) for fast inference, which also provides a way to trade off the time cost and the generation quality during inference. By setting proper diffusion steps (*e.g.,* 100 and 2), our approach can outperform competitive AR and NAR models with similar inference latency, respectively.

2 Related Work

Non-Autoregressive Text Generation. Compared with autoregressive (AR) methods (Lewis et al., 2020) that need to predict the target text in a token-by-token manner, Non-autoregressive (NAR) methods can generate all tokens in parallel, which can greatly reduce the inference latency (Gu et al., 2018; Ghazvininejad et al., 2019). However, in this way, NAT methods can not fully capture the dependency relations among tokens during decoding, leading to the sacrifice of accuracy. To address it, existing works adopt several training and inference strategies to improve the performance of NAR methods, *e.g.,* knowledge distillation (Zhou et al., 2020), glancing sampling (Qian et al., 2021), iterative decoding (Geng et al., 2021) and large-scale pre-training (Qi et al., 2021; Li et al., 2022a). In this work, we introduce the discrete diffusion model into NAR text generation, narrowing the performance gap with AR methods.

PLMs for Text Generation. Pre-trained language models (PLMs) have shown remarkable performance in generating human-like texts (Li et al., 2021). After pre-training, most existing PLMs (Raffel et al., 2020) are fine-tuned following the AR paradigm for text generation. In this way, they either reformulate generation tasks into the language model format (*e.g.,* GPT (Radford et al., 2019)), or leverage the sequence-to-sequence manner to generate the text using an autoregressive decoder (*e.g.,* BART (Lewis et al., 2020)). However, as these PLMs only focus on fine-tuning under the AR paradigm, they can not be directly used for NAR text generation. Recently, BANG (Qi et al., 2021) and ELMER (Li et al., 2022a) rely on large-scale pre-training for improving the NAR text generation. Considering the pre-training cost, we aim to efficiently adapt BART into an effective NAR model with diffusion models.

Diffusion Models for Text Generation. Diffusion models (DM) (Ho et al., 2020; Song et al., 2021b) are a class of latent variable models that can progressively denoise a random Gaussian noise into a data example. Existing DMs can be roughly categorized into continuous diffusion models (Ho et al., 2020; Tang et al., 2023a; Nikolaidou et al., 2023) and discrete diffusion models (Austin et al., 2021; Zheng et al., 2023; Qian et al., 2022), which perform diffusion on continuous signals and discrete states, respectively. Recently, DMs have been utilized for text generation and have demonstrated superiority in controllable text generation tasks (Tang et al., 2023b; Li et al., 2022b). For text-to-text generation tasks, existing works generally follow the continuous diffusion paradigm, and improve the performance by refining the model architecture (Yuan et al., 2022), adding regularization (Gao et al., 2022) and large-scale pre-training (Lin et al., 2022). In this work, we introduce discrete diffusion models into text-to-text generation tasks, and utilize a PLM to improve it.

3 Preliminary

Problem Statement. This work focuses on text-to-text generation tasks using non-autoregressive (NAR) models. Generally, text-to-text generation tasks (Sutskever et al., 2014; Vaswani et al., 2017) (*e.g.,* dialog and summarization) can be formulated as modeling the conditional probability $P(Y|C)$, where $C = \{c_1, c_2, \cdots, c_m\}$ and $Y = \{y_1, y_2, \cdots, y_n\}$ denote the input text and output text respectively, both consisting of a sequence of tokens from a vocabulary $\mathcal{V}$.

Different from AR models with the left-to-right token-by-token generation manner, NAR models (Gu et al., 2018; Lee et al., 2018) predict all tokens of the output text Y simultaneously, where each token y_i is predicted only based on the input text C. Thus, the conditional probability can be factorized as

$$P(Y|C) = \prod_{i=1}^{n} P(y_i|C), \quad (1)$$

Diffusion Models. Diffusion models (DM) (Ho et al., 2020; Song et al., 2021b) sample an example from a data distribution $p(x)$ by gradually denoising a random noise. Typically, starting from a noise x_T, the denoising process (also so-called reverse process) can be regarded as a Markov process, where the noises at $T-1, T-2, \cdots, 0$ steps are progressively predicted and removed to obtain the latent variables $x_{T-1}, x_{T-2}, \cdots$, until reaching the final sample x_0. Conversely, given the sample x_0, we can generate $x_1, x_2, \cdots, x_T$ as a Markov chain, denoted as the *forward process*:

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t \boldsymbol{I}), \quad (2)$$

where $\beta_t \in (0,1)$ is the pre-defined scaling of noise variance at the t-th step. Given the above forward process as prior, DMs are trained to reverse it following the denoising process for recovering x_0, where each step is parameterized as:

$$p(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \quad (3)$$

where $\mu_\theta(\cdot)$ and $\Sigma_\theta(\cdot)$ can be implemented by a U-Net (Ronneberger et al., 2015) or Transformer (Vaswani et al., 2017), and time step embeddings are adopted to represent t.

Discrete Diffusion Models. Discrete diffusion models (Austin et al., 2021; Gu et al., 2022) perform the forward and denoising processes in discrete random variables with K categories, where $K = |\mathcal{V}|$ for text data. For a sentence, x_0 is the vector consisting of the indexes of its contained tokens, and the forward process of adding noise is

$$q(x_t|x_{t-1}) = v^\top(x_t)\mathbf{Q}_t v(x_{t-1}), \quad (4)$$

where $v(x_t)$ maps each token index from x_t into K-dimension one-hot vector, $\mathbf{Q}_t$ is the probability transition matrix and $[\mathbf{Q}_t]_{i,j}$ denotes the probability of the token i to be replaced by the token j. In this way, according to Bayes' theorem, the denoising process $q(x_{t-1}|x_t, x_0)$ can be deduced as:

$$q(x_{t-1}|x_t, x_0) = \frac{(v^\top(x_t)\mathbf{Q}_t v(x_{t-1}))(v^\top(x_{t-1})\bar{\mathbf{Q}}_{t-1} v(x_0))}{v^\top(x_t)\bar{\mathbf{Q}}_t v(x_0)} \quad (5)$$

where $\bar{\mathbf{Q}}_t = \mathbf{Q}_1\mathbf{Q}_2\cdots\mathbf{Q}_t$. Based on the above prior, we can use a parameterized model $p_\theta(x_{t-1}|x_t, t)$ to learn the denoising process.

4 Approach

In this section, we introduce Diffusion-NAT, an effective approach to integrating the discrete diffusion model and the Seq2Seq PLM BART, for improving NAR text-to-text generation. The overview of our approach is shown in Figure 1.

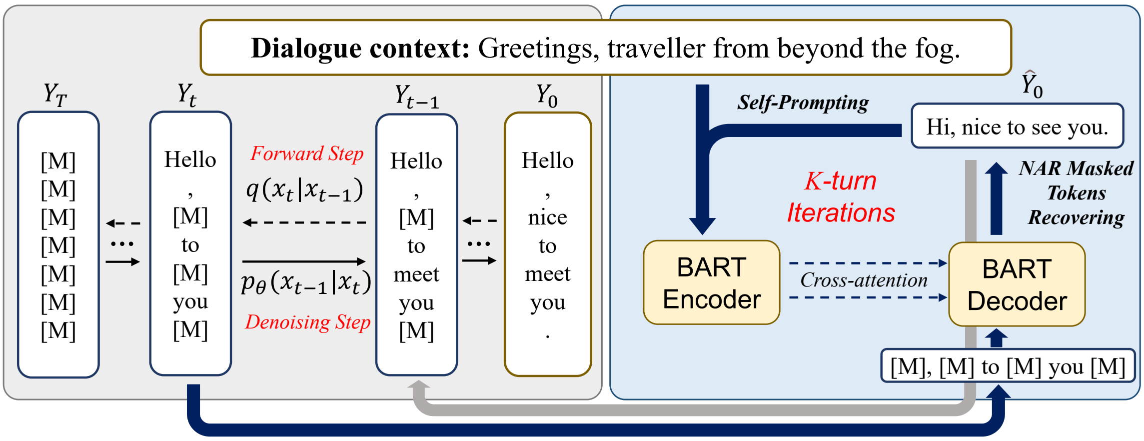


Figure 1: The overview of our Diffusion-NAT. We show an example that generates a response in the t-th step using K-turn self-prompting. The given dialog context and the K-turn prompt (*i.e.*, estimated $\hat{Y}_0$) are fed into BART encoder, and the response in the t-th Y_t is fed into BART decoder for estimating the original tokens.

4.1 Overview

Since discrete diffusion models (DDM) and BART adopt different ways for training (*i.e.,* noise prediction and masked text infilling respectively), it is hard to directly integrate both for NAR text-to-text generation. Our solution is to regard the mask token `[MASK]` of BART as the *noise* in DDM, and incorporate an absorbing state `[MASK]` into the Markov transition matrices. In this way, the forward process of DDM gradually replaces all the tokens by `[MASK]`, and the denoising process can be reformulated as a *NAR Masked Tokens Recovering (NMTR)* task:

$$f_{\text{NMTR}}([\mathtt{M}], \cdots, [\mathtt{M}]) = \{y_1, \cdots, y_n\}, \quad (6)$$

where [M] denotes the `[MASK]` token of BART. To apply this framework for NAR text generation, we further make adaptations for BART and DDM. For BART, its pre-training task of masked text infilling is similar to the above objective except that it is in a NAR manner, and thus we revise the decoding process of BART to support the NAR inference in Section 4.2. For DDM, we learn to predict the original tokens instead of noise and remove the time step embeddings in Section 4.3, for better adaptation to BART. In this way, we can unify the inference process of BART and the denoising process of discrete diffusion models with the same formulation of *NAR masked tokens recovering*.

With this unified formulation, DDM can fully rely on BART to conduct the denoising process, with no need for additional parameters or specific training. In this way, the generated results based on BART can be iteratively refined via the denoising process, leading to improved generation text. Since BART is employed as the backbone of our approach, we can naturally leverage advanced techniques of PLMs to improve the diffusion process, *e.g.,* prompt learning (Liu et al., 2021b). Thus, we propose the iterative self-prompting strategy to perform multi-turn deliberation and refinement on the intermediate generated results in Section 4.4, further enhancing the quality of the output.

4.2 Adapting BART for NAR Generation

Since BART utilizes a token-by-token autoregressive mechanism for decoding, this part discusses how to revise its decoding process to fit the NAR generation framework.

BART. BART (Lewis et al., 2020) is a Seq2Seq PLM that has been widely used on various text-to-text generation tasks. It adopts the encoder-decoder Transformer architecture. Given the input text C, the encoder produces its representation vectors $\mathbf{E}$, and the decoder performs cross-attention with $\mathbf{E}$ to inject the condition from the input text. During pre-training, the masked text infilling task is mainly adopted to learn the model parameters on a large-scale corpus, aiming to recover the masked span from the input text. During inference, using a special start token as the initial input of the decoder, the output text will be generated token by token.

Revised NAR Decoding Process. In the denoising process of our approach, BART is employed to recover the masked tokens from the noised target text at each time step. Thus, we revise the decoding process of BART into the NAR manner that can recover all masked tokens simultaneously. Concretely, at the t-step, given the condition text C and the noised target text Y_t containing [MASK] tokens, we feed them into the encoder and decoder of BART respectively, and simultaneously recover all the [MASK] tokens into the target tokens as:

$$\text{BART}(\{y_1^{(t)} \cdots [\text{M}]\}, C) = \{y_1^{(t-1)} \cdots y_n^{(t-1)}\}, \tag{7}$$

where $y_1^{(t)}$ is the token of the first position at the t-th step. In this way, the decoding process follows the unified formulation in Eq. 6. Thus, we can employ BART in the denoising process by leveraging its pre-learned knowledge and generation capacity.

4.3 Adapting DDM for NAR Generation

In this part, we discuss how to adapt the discrete diffusion model (DDM) to NAR masked tokens recovering for text generation.

Markov Transition Matrices with [MASK]. As introduced in Section 3, discrete diffusion models rely on the probability transition matrix $\mathbf{Q}_t$ to perform the forward and denoising processes over the state space. To align DDM with the NAR decoding process of BART (Section 4.2), we incorporate the [MASK] token as the absorbing state of the Markov transition matrices. Concretely, at the t-th step of the forward process, if token i is not the [MASK] token, it has the probabilities of α_t and γ_t being unchanged and replaced by the [MASK] token respectively, leaving the probability of $\beta_t = 1 - \alpha_t - \gamma_t$ transiting to other tokens in $\mathcal{V}$ as:

$$[\mathbf{Q}_t]_{i,j} = \begin{cases} \alpha_t, & \text{if } j = i, \\ \gamma_t, & \text{if } j = [\text{M}], \\ 1 - \alpha_t - \gamma_t, & \text{otherwise}, \end{cases} \tag{8}$$

where α_t and γ_t are determined by the pre-defined noise schedule, *e.g.,* cosine schedule (Nichol and Dhariwal, 2021). While, if token i is the [MASK] token, it will be unchanged. Based on such a forward process, all tokens in the output text would become [MASK] after a sufficient number of steps, corresponding to the all-[MASK] input in Eq. 6. In the denoising process, we adopt BART to gradually recover the all-[MASK] sequence into output text in the NAR manner, where each denoising step is equivalent to the decoding of BART in Section 4.2.

Training with NAR Masked Tokens Recovering. During training, existing diffusion models mostly learn to predict the noise in the current time step. However, such training objective is not consistent with PLMs. Inspired by existing works (Li et al., 2022b; Gong et al., 2022), we predict all the original tokens $Y_0 = \{y_1^{(0)}, \cdots, y_n^{(0)}\}$ using BART in the NAR manner at each time step as:

$$\text{BART}(\{y_1^{(t)} \cdots [\text{M}]\}, C) = \{y_1^{(0)} \cdots y_n^{(0)}\}. \tag{9}$$

As Y_t usually contains several [MASK] tokens, the above process can be regarded as recovering all the masked tokens into the original ones, which is actually similar to the pre-training objective of BART. In this way, the training objective is formulated as:

$$\mathcal{L}_Y = -\sum_{i=1}^{n} \log p_\theta(y_i^{(0)} | Y_t, C) \tag{10}$$

where Y_t denotes the intermediate recovered text in the t-th step. During inference, given Y_t, our model first estimates $\hat{Y}_0$, and then adds the $(t-1)$-step noise into it for producing Y_{t-1}. The above process will iterate for multiple steps, until the final results of Y_0 are obtained.

Removing Time Step Embeddings. As another difference in architecture, diffusion models typically incorporate time step embeddings to represent the time information (Ho et al., 2020; Song et al., 2021a), while BART has never set up corresponding time step embeddings. To reduce such discrepancy, we directly remove the time step embeddings from our diffusion process, so as to adapt DDM to reusing the whole architecture and all pretrained parameters of BART. Actually, as the discrete diffusion process is to progressively recover the all-[MASK] sequence, the PLM can directly acquire the time information by counting the number of [MASK] tokens. Further, by removing the time step embeddings, our diffusion approach can better integrate with other improvement techniques, *e.g.,* DDIM method (Song et al., 2021a) with the non-Markov process for fast inference.

4.4 Iterative Self-Prompting

In a typical denoising process, the denoising network relies on the condition C and Y_t to estimate $\hat{Y}_0$. However, at early steps, [MASK] tokens generally occupy the majority of Y_t, causing the estimation to be more difficult. To reduce the inference difficulty at an early stage, we propose the iterative self-prompting strategy that endows our model with deliberation capacity via prefixed prompts.

Training with Self-Prompting. Inspired by the self-conditioning strategy (Chen et al., 2022), our self-prompting strategy focuses on improving the quality of $\hat{Y}_0$ through multi-round checking and revision. Concretely, given Y_t and C, we first utilize the PLM to produce the estimated $\hat{Y}_0$. Then, as $\hat{Y}_0$ and C are two sequences of tokens, we regard $\hat{Y}_0$ as the prompt of the PLM and prefix it with C to compose the new input condition $C' = [\hat{Y}_0; C]$. Next, the new condition C' and Y_t are further fed into the encoder and decoder of the PLM respectively, where cross-attention in the decoder is employed to generate $\hat{Y}_0$ by considering the previous estimation. During training, with a certain probability (*e.g.,* 50%), we do not use the self-prompting strategy and only optimize the model parameter using Eq. 10. When integrated with this strategy, we first produce $\hat{Y}_0$ and then construct C' for self-prompting, where the training objective becomes:

$$\mathcal{L}_Y = -\sum_{i=1}^{n} \log p_\theta(y_i^{(0)} | Y_t, \hat{Y}_0, C). \tag{11}$$

Inference with Iterative Self-Prompting. To obtain a well-estimated $\hat{Y}_0$, we repeat the following self-prompting process for K times: we first estimate the original tokens $\hat{Y}_0 = \{\hat{y}_1^{(0)}, \cdots, \hat{y}_n^{(0)}\}$ based on the constructed new condition C' and then utilize it to replace the original prompt within C'. Each iterative process can be denoted as:

$$\text{BART}\big(\{y_1^{(t)} \cdots y_n^{(t)}\}, \{\hat{y}_1^{(0)} \cdots \hat{y}_n^{(0)}\}, C\big) = \{y_1^{(0)} \cdots y_n^{(0)}\}. \tag{12}$$

In this way, by setting proper hyper-parameter K, we can balance the accuracy of the estimated $\hat{Y}_0$ and the time cost during inference. Note that such a manner also supports the explicit control in the intermediate prompts for guiding the generation, *e.g.,* correcting grammar errors in $\hat{Y}_0$. We leave it as our future work.

5 Experiments

5.1 Experimental Settings

More details about the datasets, evaluation metrics, baselines, and implementations are shown in Appendix A, B, C and D, respectively.

5.2 Experimental Results

Dialog Generation. As shown in Table 2, for the coherence metrics (*i.e.,* BLEU-1/2), the performance order of aforementioned baselines in the two dialog generation datasets is mostly consistently as: *AR models > Semi-NAR models > NAR models*. It indicates that AR models are more capable of generating coherent and fluent responses than NAR ones. A major reason is that AR models can better capture the dependency of tokens. Whereas, for the diversity metrics, AR models mostly underperform NAR models. The reason may be that AR models are easy to overfit into the frequently co-occurring tokens (*e.g.,* I am OK.) in the training data, causing the "safe response" problem. Besides, the NAR methods using pre-training techniques (*i.e.,* BANG and ELMER) can better balance the coherence and diversity metrics, and greatly outperform other NAR models. It demonstrates the effectiveness of large-scale pre-training in improving the NAR generation performance.

Finally, Diffusion-NAT mostly outperforms Semi-NAR and NAR models on all metrics. Different from these baselines, our approach is based on the discrete diffusion model that can iteratively refine the generated results using a PLM BART. As we have adapted them to better fit with each other by a set of revisions, we can combine the merits of the rich knowledge from BART and the iterative refining mechanism of the diffusion model. In this way, we can improve both the coherence and diversity of the generated responses. Furthermore, our approach outperforms AR models in the average value of all metrics, *e.g.,* Ours (27.90) VS. BART (23.54) in PersonaChat. The reason is that our approach achieves much higher values in the Distinct-1,2 metrics. It shows the effectiveness of our approach for generating diverse responses.

Text Summarization and Question Generation. As shown in Table 3 and Table 4, AR models outperform NAR models in a large margin. The reason is that the two types of tasks mainly require the model to accurately generate proper texts, which is more suitable for AR models due to their superiority in capturing the token dependency. Despite this, our approach mostly outperforms all the NAR and Semi-NAR methods, and even surpasses AR models on part of datasets (*e.g.,* MSNews). It is because our approach can effectively combine the merits of the PLM that has pre-learned rich semantic knowledge and the diffusion models that can iteratively refine the produced results, leading to higher-quality generated texts.

Conversational Question Answering. The conversational question answering task is to evaluate

Type	Models	PersonaChat					DailyDialog				
		B-1↑	B-2↑	D-1↑	D-2↑	Overall↑	B-1↑	B-2↑	D-1↑	D-2↑	Overall↑
AR	**Transformer**	41.56	32.95	0.30	0.80	18.90	45.95	40.60	0.91	4.68	23.04
	MASS	41.06	35.75	1.40	6.90	21.28	51.77	45.09	3.99	23.38	31.06
	ProphetNet	46.00	38.40	1.30	7.30	23.25	-	-	-	-	-
	BART	47.60	39.36	1.10	6.10	23.54	56.18	49.59	5.04	27.72	34.63
Semi-NAR	**InsT**	12.63	9.43	0.10	0.30	5.62	-	-	-	-	-
	iNAT	41.17	32.13	0.10	1.10	18.63	-	-	-	-	-
	LevT	24.89	18.94	0.10	0.60	11.13	-	-	-	-	-
	CMLM	<u>44.38</u>	<u>35.18</u>	0.10	0.80	20.12	-	-	-	-	-
	BANG	39.82	30.72	1.90	14.20	<u>21.66</u>	41.47	35.71	1.76	13.98	23.23
NAR	**NAT**	31.53	24.17	0.10	0.80	14.15	-	-	-	-	-
	iNAT	30.56	23.38	0.10	0.70	13.69	-	-	-	-	-
	CMLM	31.44	24.06	0.10	0.60	14.05	-	-	-	-	-
	LevT	26.92	20.47	0.00	0.40	11.95	-	-	-	-	-
	BANG	31.11	23.90	2.50	22.70	20.05	35.50	30.15	1.90	15.13	20.67
	ELMER	31.45	23.99	**3.66**	<u>24.96</u>	21.02	68.32	61.14	5.30	35.64	42.60
Diffusion	**Ours**	**44.55**	**37.66**	<u>3.19</u>	**26.20**	**27.90**	**68.79**	**62.68**	**6.67**	**39.97**	**44.53**

Table 2: The comparison between our approach and baselines on two dialog generation tasks. B-1/2 and D-1/2 denote BLEU-1/2 and Distinct-1/2. **Bold** and <u>underline</u> fonts denote the best and second best methods within NAR and Semi-NAR models, respectively. The baseline results on PersonaChat are collected from (Li et al., 2022a).

both the generative capacity and the world knowledge of the model. As shown in Table 4, our approach also performs well in this task, even slightly outperforming the AR model BART by 0.8 on F1 metric. A possible reason is that our approach can make use of the pre-learned world knowledge from BART. Besides, as our model can also leverage the iterative refining paradigm of the diffusion model, it may also fix the wrong answers in the generated text, leading to more accurate answers.

Human Evaluation. In addition to the automatic metrics, human evaluation is also critical for text generation. Considering the expensive annotation cost, we only focus on the dialog generation task and compare our approach with two best-performing baselines, *i.e.,* BART and ELMER. Following existing works (Li et al., 2022a), we randomly select 500 examples from the test set of the PersonaChat dataset, and invite three annotators to evaluate the quality of the generated responses from the two baselines and ours from the perspectives of Fluency, Informativeness and Relevance. The scoring range is from 1 to 5. As shown in Table 5, we can see that the AR method BART performs better on the Fluency and Relevance metrics while the NAR method ELMER performs well on informativeness. Such results show a similar tendency as the automatic metrics, and indicate the different superiority of AR and NAR models. As a comparison, our approach can well balance the three metrics, with the comparable performance on Fluency as BART and the best performance on Informativeness. It shows the great potentiality of discrete diffusion models with PLMs in NAR text-to-text generation tasks.

Inference Latency. By using DDIM (Song et al., 2021a) or other acceleration strategies, we can also reduce the inference latency of our approach. To verify it, we test the inference latency and performance of our approach using different diffusion steps by using DDIM, and compare them with two best-performing NAR and AR baselines (*i.e.,* ELMER and BART) on PersonaChat dataset. The above experiments are conducted on a NVIDIA 3090-24G GPU with a batch size of 1. As shown in Table 10, we can see that our approach can provide a way to trade off the time cost and the generation quality during inference. By setting proper diffusion steps (100 and 2), our approach can outperform BART and ELMER on average with similar inference latency, respectively.

6 Conclusion

In this paper, we proposed Diffusion-NAT, a self-prompting discrete diffusion model (DDM) using a PLM BART for non-autoregressive (NAR) text generation. In our approach, we unified the inference process of BART and the denoising process of DDM into the same masked tokens recovering task, to combine the merits of both the rich pre-

Type	Models	XSUM			SQuAD v1.1		
		ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	ROUGE-L↑	BLEU-4↑	METEOR↑
AR	**Transformer**	30.66	10.80	24.48	29.43	4.61	9.86
	MASS	39.70	17.24	31.91	49.48	20.16	24.41
	ProphetNet	39.89	17.12	32.07	48.00	19.58	23.94
	BART	38.79	16.16	30.61	42.55	17.08	23.19
Semi-NAR	**InsT**	17.65	5.18	16.05	29.98	2.34	8.15
	iNAT	26.95	6.88	22.43	32.34	3.16	9.18
	LevT	25.33	7.40	21.48	30.81	2.68	9.40
	CMLM	29.12	7.70	23.04	29.60	3.89	9.70
	BANG	34.71	11.71	29.16	**47.39**	**17.62**	21.69
NAR	**NAT**	24.04	3.88	20.32	31.51	2.46	8.86
	iNAT	24.02	3.99	20.36	32.44	2.33	8.84
	CMLM	23.82	3.60	20.15	31.58	2.51	8.85
	LevT	24.75	4.18	20.87	31.38	2.27	9.14
	BANG	32.59	8.98	27.41	44.07	12.75	18.99
	ELMER	38.30	14.17	29.92	40.22	13.49	20.08
Diffusion	**GENIE**	29.3	8.3	21.9	-	-	-
	AR-DIFFUSION	32.2	10.6	25.2	-	-	-
	Ours	**38.84**	**15.30**	**30.88**	46.64	16.19	**21.99**

Table 3: The comparison between different methods on XSUM and SQuAD v1.1 datasets. **Bold** and underline fonts denote the best and second best methods within NAR and Semi-NAR models, respectively. The baseline results are collected from (Qi et al., 2021) and (Li et al., 2022a).

Models	MSNews			MSQG			CoQA
	ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	ROUGE-L↑	BLEU-4↑	METEOR↑	F1↑
LSTM	30.0	14.6	27.7	25.3	3.5	14.1	15.1
Transformer	33.0	15.4	30.0	29.3	5.1	16.6	15.7
BART	41.8	23.1	38.3	38.1	10.2	22.1	64.6
BANG	32.7	16.1	30.3	33.1	**11.0**	18.4	31.4
ELMER	35.6	16.1	32.5	26.6	5.00	15.7	63.1
Ours	**46.8**	**31.6**	**44.2**	**33.3**	6.6	**19.3**	**65.4**

Table 4: The comparison between different methods on MSNews, MSQG and CoQA datasets. **Bold** and underline fonts denote the best and second best methods within NAR models, respectively.

Models	PersonaChat		
	Fluency	**Informativeness**	**Relevance**
BART	4.32	4.31	3.47
ELMER	3.88	4.49	2.90
Ours	4.29	4.57	3.19

Table 5: Human evaluation scores of different methods about the generated responses on PersonaChat.

	ELMER	**Diffusion-NAT**			**BART**
Steps	-	2	20	100	-
Latency	13.8ms	19.1ms	76.4ms	267.5ms	253.6ms
BLEU-2	23.99	30.82	36.19	37.66	39.36
Dist-2	24.96	23.68	26.93	26.20	6.10

Table 6: Performance and inference latency changes of two baselines and our approach w.r.t. the diffusion steps using DDIM during inference on PersonaChat dataset.

learned knowledge of BART and the iterative refining paradigm of DDM. Concretely, we revised the decoding process of BART into the NAR manner, and adapted the typical settings of DDM to better fit with BART, including Markov transition matrix, training objective and time step embeddings. Besides, we devised an iterative self-prompting strategy to guide the PLM to deliberate and refine the intermediate generated results, to further improve the quality of final produced texts. Extensive experiments on seven datasets have shown that our approach can outperform competitive NAR and Semi-NAR models, and even surpass AR models.

In future work, we will investigate more effective and efficient way to combine LLMs and DDM for NAR text generation, *e.g.,* prompt learning.

Limitations

This work is to investigate discrete diffusion models with pre-trained language models for non-autoregressive text-to-text generation. An important limitation is the relatively higher inference latency of diffusion models. In this work, we have adopted DDIM to accelerate the inference process by reducing the diffusion steps, and we also conduct experiments to investigate the performance changes w.r.t. different steps in Appendix H. We can see that fewer steps using DDIM would lead to the performance degradation. Fortunately, there are several recent works that have shown effectiveness in solving this problem (Lu et al., 2022). As these methods are general to all diffusion models, they may be able to be utilized in our approach. Besides, as we have adopted a PLM, BART in our approach, it may present biases learned from the pre-training corpus in the generated texts.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17981–17993.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics.

Linyao Chen, Aosong Feng, Boming Yang, and Zihui Li. 2023. Xdlm: Cross-lingual diffusion language model for machine translation. *arXiv preprint arXiv:2307.13560*.

Ting Chen, Ruixiang Zhang, and Geoffrey E. Hinton. 2022. Analog bits: Generating discrete data using diffusion models with self-conditioning. *CoRR*, abs/2208.04202.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Prafulla Dhariwal and Alexander Quinn Nichol. 2021. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 8780–8794.

Griffin Floto, Mohammad Mahdi Abdollah Pour, Parsa Farinneya, Zhenwei Tang, Ali Pesaranghader, Manasa Bharadwaj, and Scott Sanner. 2023. Diffudetox: A mixed diffusion model for text detoxification. *arXiv preprint arXiv:2306.08505*.

Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. 2022. Difformer: Empowering diffusion model on embedding space for text generation. *CoRR*, abs/2212.09412.

Xinwei Geng, Xiaocheng Feng, and Bing Qin. 2021. Learning to rewrite for non-autoregressive neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3297–3308. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6111–6120. Association for Computational Linguistics.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *CoRR*, abs/2210.08933.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11179–11189.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. 2022. Vector quantized diffusion model for text-to-image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10686–10696. IEEE.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1173–1182. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2022a. ELMER: A non-autoregressive pre-trained language model for efficient and effective text generation. *CoRR*, abs/2210.13304.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained language models for text generation: A survey. *CoRR*, abs/2105.10311.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022b. Diffusion-lm improves controllable text generation. *CoRR*, abs/2205.14217.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 986–995. Asian Federation of Natural Language Processing.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Weizhu Chen, and Nan Duan. 2022. GENIE: large scale pre-training for text generation with diffusion model. *CoRR*, abs/2212.11685.

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, Pengcheng Wang, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, Ruofei Zhang, Winnie Wu, Ming Zhou, and Nan Duan. 2021a. GLGE: A new general language generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 408–420. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*.

Yiwei Lyu, Tiange Luo, Jiacheng Shi, Todd C Hollon, and Honglak Lee. 2023. Fine-grained text style transfer with diffusion-based language models. *arXiv preprint arXiv:2305.19512*.

Rabeeh Karimi Mahabadi, Jaesung Tae, Hamish Ivison, James Henderson, Iz Beltagy, Matthew E Peters, and Arman Cohan. 2023. Tess: Text-to-text self-conditioned simplex diffusion. *arXiv preprint arXiv:2305.08379*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR.

Konstantina Nikolaidou, George Retsinas, Vincent Christlein, Mathias Seuret, Giorgos Sfikas, Elisa Barney Smith, Hamam Mokayed, and Marcus Liwicki. 2023. Wordstylist: Styled verbatim handwritten text generation with latent diffusion models.

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, July 6-12, 2002, Philadelphia, PA, USA, pages 311–318. ACL.

Weizhen Qi, Yeyun Gong, Jian Jiao, Yu Yan, Weizhu Chen, Dayiheng Liu, Kewen Tang, Houqiang Li, Jiusheng Chen, Ruofei Zhang, Ming Zhou, and Nan Duan. 2021. BANG: bridging autoregressive and non-autoregressive generation with large scale pretraining. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8630–8639. PMLR.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2401–2410. Association for Computational Linguistics.

Lihua Qian, Mingxuan Wang, Yang Liu, and Hao Zhou. 2022. Diff-glat: Diffusion glancing transformer for parallel sequence to sequence learning. *arXiv preprint arXiv:2212.10240*.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1993–2003. Association for Computational Linguistics.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021a. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021b. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5976–5985. PMLR.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Zecheng Tang, Pinzheng Wang, Keyan Zhou, Juntao Li, Ziqiang Cao, and Min Zhang. 2023a. Can diffusion model achieve better performance in text generation? bridging the gap between training and inference!

Zecheng Tang, Pinzheng Wang, Keyan Zhou, Juntao Li, Ziqiang Cao, and Min Zhang. 2023b. Can diffusion model achieve better performance in text generation? bridging the gap between training and inference! *arXiv preprint arXiv:2305.04465*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Tong Wu, Zhihao Fan, Xiao Liu, Yeyun Gong, Yelong Shen, Jian Jiao, Hai-Tao Zheng, Juntao Li, Zhongyu Wei, Jian Guo, Nan Duan, and Weizhu Chen. 2023. Ar-diffusion: Auto-regressive diffusion model for text generation.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2022. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *CoRR*, abs/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? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2204–2213. Association for Computational Linguistics.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. 2023. A reparameterized discrete diffusion model for text generation.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Task	Datasets	#Train	#Valid	#Test
Dialog	**DailyDialog**	76,052	7,069	6,740
	PersonaChat	122,499	14,602	14,056
Sum.	**XSUM**	204,045	11,332	11,334
	MSNews	136,082	7,496	7,562
QG	**MSQG**	198,058	11,008	11,022
	SQUAD v1.1	75,722	10,570	11,877
CQA	**CoQA**	108,647	3,935	4,048

Table 7: Statistics of the datasets, where **Dialog**, **Sum.**, **QG** and **CQA** denote Dialog Generation, Text Summrization, Question Generation and Conversational Question Answering, respectively.

A Details of Datasets

We conduct experiments on seven datasets, corresponding to four representative text generation tasks. The statistics of these datasets are shown in table 7.

- **Dialog Generation** aims to predict responses according to the dialog history. We select **DailyDialog** (Li et al., 2017) and **PersonaChat** (Zhang et al., 2018) datasets.

- **Text Summarization** is to summarize the document into a sentence. We choose **XSUM** (Narayan et al., 2018) and **MSNews** (Liu et al., 2021a), two news summarization datasets.

- **Question Generation** aims to generate questions based on given passages and answers. We use **MSQG** (Liu et al., 2021a) and **SQUAD v1.1** (Rajpurkar et al., 2016) datasets.

- **Conversational Question Answering** is to answer the question based on a conversation. We select **CoQA** (Reddy et al., 2019) dataset.

B Details of Evaluation Metrics.

Following existing works (Li et al., 2022a; Qi et al., 2021), we employ corresponding metrics to evaluate model performances on different tasks.

- For dialog generation, we adopt BLEU-1/2 (Papineni et al., 2002) to measure the coherence between the generated and real responses based on the co-occurrence ratio of n-grams, and Distinct-1/2 (Li et al., 2016) to measure the n-gram diversity of the generated texts.

- For text summarization, we utilize ROUGE-1/2/L (Lin, 2004) to compute the overlapping ratio of n-grams between the generated and ground-truth summarizations for estimating the quality.

- For question generation, we use ROUGE-L, BLEU-4 and METEOR (Banerjee and Lavie, 2005) to assess the generation consistency.

- For conversational question answering, we adopt F1-Score (Rajpurkar et al., 2016) to measure the prediction accuracy.

C Details of Baselines

We mainly compare our Diffusion-NAT with a variety of Semi-NAR and NAR models. **NAT** (Gu et al., 2018), **iNAT** (Lee et al., 2018), **InsT** (Stern et al., 2019), **CMLM** (Ghazvininejad et al., 2019) and **LevT** (Gu et al., 2019) are five Transformer-based NAR models with special generation strategies, *i.e.,* iterative refinement, conditional masked language modeling and insertion-deletion operation. **BANG** (Qi et al., 2021) and **ELMER** (Li et al., 2022a) adopt the pre-training technique based on Transformer to further improve the NAR generation performance. Note that InsT, iNAT, LevT, CMLM and BANG also support the semi-NAR manner that can rely on partially generated results for improving the inference. We also compare our approach with two recently proposed diffusion-based methods, *i.e.,* **GENIE** (Lin et al., 2022) and **AR-DIFFUSION** (Wu et al., 2023), which incorporate the pre-training strategy and auto-regressive decoding to improve the generation performance of continuous diffusion models.

We also compare our approach with AR models which have shown better accuracy than NAR ones. **LSTM** (Hochreiter and Schmidhuber, 1997) and **Transformer** (Vaswani et al., 2017) are two classic Seq2Seq models. **MASS** (Song et al., 2019), **BART** (Lewis et al., 2020) and **ProphetNet** (Qi et al., 2020) are PLMs specially for text generation and we use their base version for fair comparison.

D Implementation Details

For all baselines, we use the source code provided by their authors, and all hyper-parameters are set following the suggestions from the original paper. For our Diffusion-NAT, we use the checkpoint of BART-base with 110M parameters for initialization, and do not add any other parameters. We use the linear noise schedule (Ho et al., 2020) for the diffusion process. During training, the diffusion step is set to 1000. During inference, we utilize DDIM (Song et al., 2021a) for fast sampling and reduce the diffusion step into 100. The number of self-prompting turns is set to 2. We use AdamW as the optimizer, and set learning rate to 5e-5. We set the training step for XSUM and SQuAD v1.1 to 120k, and 80k for other datasets. The batch size is set to 512. All experiments are conducted on 8 NVIDIA Tesla V100 GPUs. The training process of each task requires less than 24 hours.

Models	PersonaChat			
	B-1	B-2	D-1	D-2
ELMER	31.11	23.99	3.66	24.96
Ours	44.55	37.66	3.19	26.20
-w/o self-prompting	43.93	37.19	2.62	22.22
-w/o PLM	41.39	35.33	1.74	17.31
+Time step Embedding	40.03	33.80	1.75	16.80
BART=>RoBERTa	38.07	32.17	2.99	18.32

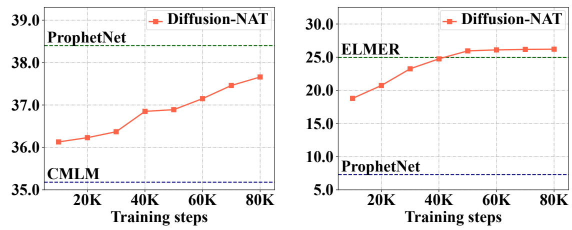
Table 8: Ablation study on PersonaChat dataset.

Models	PersonaChat		XSUM	SQuAD	
	B-1	B-2	R-L	R-L	MT
DiffuSeq	37.79	32.50	20.29	29.29	12.57
Ours	44.55	37.66	30.88	46.64	21.99

Table 9: Performance comparison of continuous diffusion method DiffuSeq (Gong et al., 2022) and our approach on PersonaChat, XSUM and SQuAD datasets.

E Ablation and Variation Study

Our Diffusion-NAT includes several key designs, *i.e.,* the usage of BART, self-prompting strategy, removing time step embeddings. Here, we conduct the ablation and variation study on our approach to verify their effectiveness. Concretely, we propose four variations of our approach as shown in Table 8, where **-w/o self-prompting** and **-w/o PLM** refer to the variations removing the corresponding component, **+Time step Embeddings** and **BART=>RoBERTa** are the variations that add the time step embeddings as continuous diffusion methods (Li et al., 2021) and replaces BART by RoBERTa in our approach, respectively. We can see that all the variations underperform our approach, it demonstrates the effectiveness of the above designs. Among them, we can see that adding time step embeddings cause the performance degrading a lot. The reason is that the additional embeddings may disturb the original semantic representations of BART.



(a) BLEU-2 (b) Distinct-2

Figure 2: Performance changes of our approach w.r.t. the training steps on PersonaChat dataset.

F Discrete Diffusion V.S. Continuous Diffusion

For the NAR text-to-text generation, existing works (Gong et al., 2022) also have incorporated the continuous diffusion method. In this part, we aim to compare our approach with a recently proposed work, DiffuSeq (Gong et al., 2022) that performs continuous diffusion on the latent space of token embeddings and leverages the KNN rounding step to map the embeddings into discrete tokens. We conduct the experiments on PersonaChat, XSUM and SQuAD datasets. As shown in Table 9, we can see that our approach outperforms DiffuSeq in all metrics by a large margin. It shows the effectiveness of our proposed method that utilizes the discrete diffusion method in NAR text-to-text generation tasks. Besides, compared with DiffuSeq, our approach can also benefit from the PLM BART, which also helps generate higher-quality texts.

G Performance w.r.t. Training Steps

As our approach adopts the pre-trained BART for parameters initialization, it is also helpful to faster and better convergence. To verify it, we report the BLEU-2 and Distinct-2 performance changes of our approach w.r.t. the training steps during training. As show in Figure 2, we observe that with the increasing of training steps, the performance of our approach is consistently improving, gradually approaching or surpassing competitive models. It shows the stabilization of our convergence process. Besides, for BLEU-2, with just 10k training steps, our approach can outperform competitive Semi-NAR model CMLM. The reason may be that BART provides a good starting point of the training process, making our approach converge faster.

	PersonaChat					
Diff. Steps	2	10	20	100	200	1000
BLEU-2	30.82	35.88	36.19	37.66	37.63	37.65
Distinct-2	23.68	27.54	26.93	26.20	26.35	26.39

Table 10: Performance changes w.r.t. the diffusion steps (abbreviated as Diff. Steps) on PersonaChat dataset.

	PersonaChat					
SP Turns	0	1	2	3	4	5
BLEU-2	35.00	36.50	37.66	37.69	37.77	37.77
Distinct-2	26.01	26.22	26.20	26.34	26.29	26.30

Table 11: Performance changes w.r.t. the self-prompting turns (abbreviated as SP Turns) on PersonaChat dataset.

H Hyper-parameter Tuning.

Our approach also requires some parameters to tune, *i.e.,* the diffusion steps during decoding and the turns of self-prompting. Generally, more diffusion steps and self-prompting turns would lead to better performance but larger inference latency, hence we can tune their values to balance the inference time cost and quality. In this part, we conduct experiments on the PersonaChat dataset to validate it. As shown in Table 10 and Table 11, we can see that more diffusion steps and more self-prompting turns are able to improve the model performance, while the improvement seems to be saturated after a certain number, *i.e.,* 100 for diffusion steps and 2 for self-prompting turns. Such results can provide a reference for tuning the two hyper-parameters to match the requirement of model performance and inference latency. Besides, we can see that with very few diffusion steps (*e.g.,* 2 steps), our approach can also achieve a decent performance on BLEU-2 and Distinct-2. It demonstrates the potentiality of further reducing the inference latency in our approach.

I Case Study

To provide the qualitative analysis on our approach, we show two generated examples on the PersonaChat dataset in Table 12. We can see that with the help of BART and the diffusion model, our approach can generate relevant and informative responses based on the given dialog context. Besides, the left example shows that our approach is able to generate interesting phrases such as “as healthy as american lifestyle”, which makes the response

Personal Profile	i enjoy cross stitch and quilting. my father served in our military in the war. i am proud to be an american. i am what people call a baby boomer. my parents were proud immigrants to america.	my father was a police officer. my favorite band is nirvana. i like running. i work at pet smart. i am a musician.
Context	hello, how is your day going hi. my day is good. i am hanging out with both of my sons. cool, sounds like fun. just as much fun as cross stitch and quilts i love those crafts! right now i am into my aquarium filled with exotic fish. i am what you would call a baby boomer, explains the quilting passion you might like my collection of decorations. they are 50s vintage! sure, it would bring back old memories. especially when my dad served in the military that is awesome. what do you eat for breakfast? mine is granola every day. oatmeal fan over here. my parents were immigrants, they raised me on oats everyday. that is very healthy. i like it. do you eat fish?	hey. want to chat? i am just listening to my favorite music, nirvana. i was just reading a biography. i love those. i work too much overtime at pet smart to read. what do you do? i just got out of college.
Real Response	yes i do eat fish. i love it	what did you study. i work at pet smart but really like music.
Ours	fish is almost as healthy as american lifestyle, love fish too	i am a musician. and i play music all the time.

Table 12: Examples of generated responses on PersonaChat by our approach.

more humorous and also well reflects the speaker's personal characteristics.