Text Diffusion with Reinforced Conditioning

Yuxuan Liu[1], Tianchi Yang[2], Shaohan Huang[2], Zihan Zhang[2], Haizhen Huang[2]
Furu Wei[2], Weiwei Deng[2], Feng Sun[2], Qi Zhang[2]

[1]Peking University
[2]Microsoft Corporation
yx.liu@stu.pku.edu.cn

Abstract

Diffusion models have demonstrated exceptional capability in generating high-quality images, videos, and audio. Due to their adaptiveness in iterative refinement, they provide a strong potential for achieving better non-autoregressive sequence generation. However, existing text diffusion models still fall short in their performance due to a challenge in handling the discreteness of language. This paper thoroughly analyzes text diffusion models and uncovers two significant limitations: degradation of self-conditioning during training and misalignment between training and sampling. Motivated by our findings, we propose a novel **T**ext Diffusion model called TREC, which mitigates the degradation with **Re**inforced **C**onditioning and the misalignment by Time-Aware Variance Scaling. Our extensive experiments demonstrate the competitiveness of TREC against autoregressive, non-autoregressive, and diffusion baselines. Moreover, qualitative analysis shows its advanced ability to fully utilize the diffusion process in refining samples.

Introduction

Diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2021) are the de facto state-of-the-art generative models in the field of vision (Rombach et al. 2022; Ho et al. 2022) and audio (Kong et al. 2021; Liu et al. 2022b) given their promising capability in generating high-quality samples. However, due to the discrete nature of language modality, it is non-trivial to extend diffusion to the field of natural language generation (NLG), and how to empower NLG with diffusion models is becoming a rapidly emerging research area.

On this front, Austin et al. (2021a) and Hoogeboom et al. (2021) design a discrete diffusion process based on categorical distributions, while He et al. (2022) explored diffusion with state absorption (i.e., mask tokens as noise injection). Li et al. (2022) first proposed to directly remedy the discrete nature by mapping words onto a continuous embedding space. However, the above studies only achieved unconditional or coarse-grained control of sequence generation, whose empirical applications are limited.

Consequently, subsequent works mainly focus on conditional generation, which is a more universally applicable scenario in NLG. Later improvements in the conditioning strategies are mainly categorized three-fold. The first line includes conditioning on controlling attributes, like topics or sentiments (Lovelace et al. 2022; Liu et al. 2022a; Li et al. 2023). The second line applies diffusion models to text-to-text generation, i.e., conditioning on input sequences (Gong et al. 2023; Yuan et al. 2022; Gao et al. 2022; Ye et al. 2023). This yields more applicable tasks like machine translation or paraphrasing, which are considered more challenging (Li et al. 2023). The third line study conditioning on predictions of previous steps, namely self-conditioning (Chen, Zhang, and Hinton 2023) to boost model performance.

In this paper, we start by taking a thorough analysis of the vanilla self-conditioning approach and observe it suffers from degradation - marginalizing the diffusion latent. Hampered by such degradation, sampling with self-conditioning heavily depends on the quality of the first step (from pure Gaussian) and fails to fully utilize the diffusion process. Besides, by analyzing current sampling methods in text diffusion models, we discover and study the misalignment issue, bringing out insights in designing a better variance schedule.

Motivated by our findings, we propose TREC, a novel approach that empower Text Diffusion models with Reinforced Conditioning. Specifically, we develop a novel reinforced self-conditioning that mitigates the degradation by directly motivating quality improvements from self-conditions with reward signals. Furthermore, we propose time-aware variance scaling that facilitates training of diffusion. We conduct a series of experiments on various tasks of NLG, including machine translation, paraphrasing, and question generation. Results show that composing operators within our method manages to generate high-quality sequences, outperforming a series of autoregressive, non-autoregressive, and diffusion baselines. Detailed analysis demonstrates the effectiveness of TREC in mitigating degradation of self-conditioning with reward signals, as well as leveraging the diffusion process to iteratively refine its output.

Preliminaries

Denoising Diffusion Probablistic Models

Denoising diffusion probabilistic models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) learn a series of state transitions from prior data distribution $z_0 \sim q(x)$ to pure Gaussian $z_T \sim \mathcal{N}(0, \mathbf{I})$ through forward and reverse diffusion process. Each forward diffusion step $t \in [1, 2, ..., T]$ is a Markov process: $q(z_t|z_{t-1}) = \mathcal{N}(z_t; \sqrt{1-\beta_t} z_{t-1}, \beta_t \mathbf{I})$, where β_t is a schedule for variance scale added at each forward step. Using the superposition property of the Gaussian distribution, we obtain the following closed form for sampling z_t from z_0:

$$q(z_t|z_0) = \mathcal{N}(z_t; \sqrt{1-\bar{\beta}_t} z_{t-1}, \bar{\beta}_t \mathbf{I}), \tag{1}$$

where $\bar{\beta}_t := 1 - \prod_{i=0}^{t}(1-\beta_i)$. In the reverse diffusion process, we learn a denoising function: $p_\theta(z_{t-1}|z_t) = \mathcal{N}(z_{t-1}; \mu_\theta(z_t, t), \Sigma_\theta(z_t, t))$, where μ_θ and Σ_θ denote model's prediction on mean and variance for z_{t-1}, respectively. With the reverse process, we could reconstruct z_0 by gradually denoising z_T following the trajectory $z_T \rightarrow z_{T-1} \rightarrow ... \rightarrow z_0$. To parameterize the model, we define $\mu_\theta(z_t, t)$ as $\mu(z_{t-1}|z_t, \hat{z_0})$ and predict $\hat{z_0}$ via a neural network: $\hat{z_0} = f_\theta(z_t, t)$. Then we could train the diffusion model through minimizing the prediction error (Ho, Jain, and Abbeel 2020):

$$\mathcal{L}_{\text{Diffusion}}(\widehat{z_0}) = \mathbb{E}_{z_0,t}\left[\|\hat{z_0} - z_0\|^2\right]. \tag{2}$$

Self-Conditioning

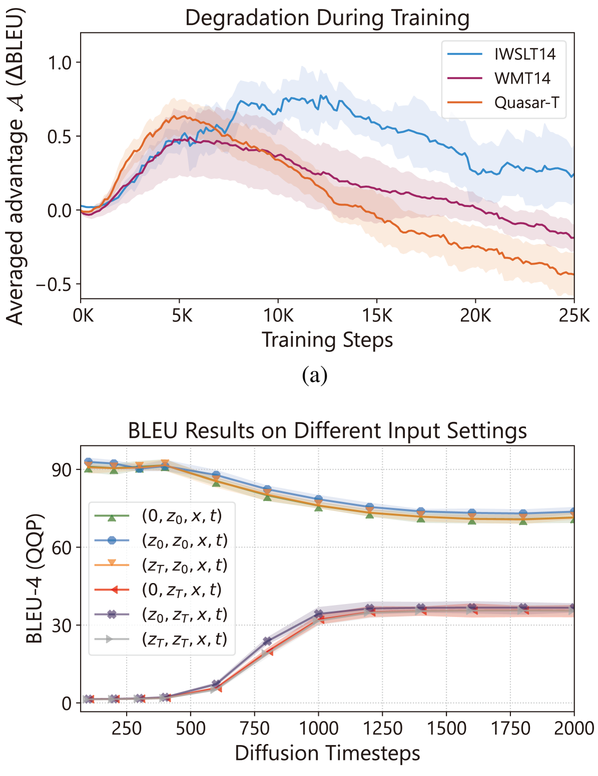
First proposed in Analog Bits (Chen, Zhang, and Hinton 2023), self-conditioning has shown to be an effective method in training denoising diffusion probabilistic models (Gao et al. 2022; Yuan et al. 2022). Self-conditioning slightly alters the denoising function from $f_\theta(z_t, t)$ to $f_\theta(z_t, \widehat{z_0}, t)$, to leverage z_0 prediction from the previous step. During training, self-conditioning is taken at a certain probability (e.g., 50%), otherwise the vanilla denoising function $f_\theta(z_t, 0, t)$ is trained (setting $\widehat{z_0} = \mathbf{0}$). At one training step $t \sim U(0, T)$, we first obtain an initial prediction $\widehat{z_0} = f_\theta(z_t, 0, t)$, then predict z_0 again by feeding the concatenation of z_t and $\widehat{z_0}$ into model (i.e., $z_0^{SC} = f_\theta(z_t, \widehat{z_0}, t)$). Since we only back propagate on z_0^{SC}, such method could be employed with only a small cost increase during training (Chen, Zhang, and Hinton 2023), and that is negligible during sampling.

Continuous Diffusion for Text Generation

Continuous text diffusion models map discrete sequences onto a continuous space (e.g., word vector space) and diffuse over this space (Li et al. 2022; Gong et al. 2023; Yuan et al. 2022; Gao et al. 2022; Dieleman et al. 2022). To bring the optimization objective, we could regard diffusion models as variational auto-encoders, and minimizing the evidence lower bound (ELBO) of $\log p_\theta(y)$ (Vahdat, Kreis, and Kautz 2021; Wehenkel and Louppe 2021) theoretically as:

$$\mathcal{L}(y) = \mathbb{E}_{y,z_0 \sim q(y)}\left[\mathcal{L}_{\text{Diffusion}}(\widehat{z_0}) - \log p_\theta(y|z_0)\right], \tag{3}$$

where y is the target sequence. On estimating $\log p_\theta(y)$, Li et al. (2022) first propose to sample z_0 from noisy word embedding of y: $\mathcal{N}(Emb(y), \beta_0 \mathbf{I})$, and address the reconstruction of y with $\log p_\theta(y|z_0)$. Gao et al. (2022) found this trivial as the gap between noisy start z_0 and $Emb(y)$ is relatively small, and propose to train by directly reconstructing y from model's output, i.e., $\log p_\theta(y|\widehat{z_0})$. In extension to conditioned sequence generation, current approaches alter denoising function $f_\theta(z_t, t)$ by adding source sequence x or controlling attributes a to conditions, i.e., $f_\theta(z_t, x, t)$ and $f_\theta(z_t, a, t)$.



(a)

(b)

Figure 1: Degradation of Self-Conditioning. (a) Quality advantage (ΔBLEU) from self-conditioning on valid set during training, which first increases and then decreases. (b) BLEU scores based on inputs $(\forall, z_0, x, t)$ and $(\forall, z_T, x, t)$ constructed from validation samples. This further validates that the model is extremely sensitive to $\hat{z}_0$ (*the first term, previous prediction*) and insensitive to the z_t (*the second term, noised latent*) to be denoised.

Pitfalls of Status Quo

Degradation of Self-Conditioning During Training

In this section, we recognize and analyze the degradation of self-conditioning during training process of continuous diffusion language models. As elaborated in the previous chapter, self-conditioning is designed to utilize near-accurate prediction $\hat{z}_0$ to provide an additional conditional guidance and act as a hint to the denoising function for better denoising. By adding self-condition, we motivate the model to perform better denoising by providing additional information.

However, such desired improvements on denoising are not aligned with the training process, and thus not ensured during sampling. Start by recalling the training objective $\mathcal{L} = \mathbb{E}_{z,t}(||z_0^{SC} - z_0|| - \log p(y|z_0^{SC}))$ and denoising step $z_0^{SC} = f_\theta(z_t, \widehat{z}_0, x, t)$. When the model is mostly converged, it can provide near accurate $\hat{z}_0$ predictions. Even if the self-condition step fails to further optimize z_0^{SC} over $\hat{z}_0$, the total training objective could still converge due to the improving accurateness of $\hat{z}_0$ predictions. Therefore, the self-condition denoising step $f_\theta(z_t, \hat{z}_0, x, t)$ could easily achieve a low loss by simply copying $\hat{z}_0$ as its output, as reconstruction from $\hat{z}_0$ to z_0^{SC} becomes substantially easier when $\hat{z}_0$'s quality increases progressively. To this end, there would be a great tendency for π_θ^{SC} to *marginalize or even ignore* z_t, which makes self-conditioned training trivial. We define this phenomenon the **degradation of self-conditioning**.

Definition 1 (Degradation of Self-Condition). *Denote* $\hat{z}_0 = f_\theta\left(z_t, 0, x, t\right)$ *the initial prediction of denoising target* z_0 *without self-conditioning; t and x the diffusion step and input condition, respectively. A denoising function* $f_\theta\left(z_t, \widehat{z}_0, x, t\right)$ *is* ***degraded*** *if it marginalizes the noised latent term* z_t.

To consolidate this analysis, we provide two experimental observations, as shown in Figure 1. To track the denoising quality during training phase, we evaluate the quality of $\widehat{z}_0$ and z_0^{SC} with a tractable metric (i.e., BLEU), and calculate the quality improvements $\mathcal{A}$ of self-conditioned denoising over the initial prediction by

$$\mathcal{A} = (BLEU(\hat{y}|z_0^{SC}, y) - BLEU(\hat{y}|\widehat{z}_0, y)) \tag{4}$$

during training. As illustrated in Figure 1(a), the quality advantages first rises then decreases, indicating such degradation do occur during training. In Figure 1(b), we further assure such degradation by feeding six diverse input combinations of $(z_t, \hat{z}_0)$. As illustrated in the Figure 1(b), performance curves with same $\hat{z}_0$: z_0 (ground truth) or z_T (pure Gaussian) *highly overlaps*, showing that given the same $\hat{z}_0$, information provided in z_t provide merely insignificant impact (as there aren't significant difference within group $(0, z_0), (z_0, z_0), (z_T, z_0)$ or $(0, z_T), (z_0, z_T), (z_T, z_T)$). This phenomenon indicates that outputs are heavily conditioned on last-step predictions $\hat{z}_0$, but mostly independent of noised latent z_t, which should have been focused instead. Such degradation trivializes the diffusion process, which obviously contradicts the design goals of self-conditioning and diffusion.

Misalignment With Training During Sampling

Sampling is critical in obtaining high-quality outputs for text diffusion models. From NLP perspective, Li et al. (2022) propose rounding trick to match each predicted embedding to its nearest neighbor during sampling to prevent diffusion on non-vocabulary. However, such KNN is time-heavy, and its loss $\mathcal{L}_{round}$ leads to unstable training. From the diffusion side, latest work include asymmetric time intervals (Chen, Zhang, and Hinton 2023) and noise factor (Gao et al. 2022). Specifically, the former alter the denoising function with small time gap (i.e, from $f_\theta(z_t, t)$ to $f_\theta(z_t, t + \Delta)$), while the latter propose to train with a higher variance prior $\mathcal{N}(0, F^2\mathbf{I}), F \geq 1$, then sample with a smaller one, i.e., $F = 1$.

However, despite their practical gains, they are proposed from a pure empirical perspective without supporting theories, and the in-depth explanations beyond their effects remain under-explored. In this section, we study the misalignment with training during sampling, and derive that existing works (Chen, Zhang, and Hinton 2023; Gao et al. 2022) are complementary in terms of mitigating such misalignment, and in preventing such phenomenon brings us clear insights to designing a better sampling regime.

Definition 2 (Misalignment During Sampling). *Given data sample* (x, y) *(e.g., paired sequences), sampling step t, and diffusion latent* z_{t+1} *from the previous step of reverse diffusion. We define* z_{t+1} *is misaligned with training during sampling, if it becomes a small probability event under the distribution* $z_{t+1} \sim \mathcal{N}(z_0; \sqrt{1-\bar{\beta}_t} z_0, \bar{\beta}_t{}^2\mathbf{I})$.

Study on Misalignment During Sampling Consider a sampling step of diffusion process at given time-step t, in which we sample z_{t-1} based on

$$\widehat{z}_{t-1} \sim q\left(\widehat{z}_{t-1}|\widehat{z}_t, \widehat{z}_0\right) p_\theta\left(\widehat{z}_0|\widehat{z}_t, x, t\right).$$

q denotes the DDIM (Song, Meng, and Ermon 2021) sampler $q\left(\hat{z}_{t-1}|\widehat{z}_t, \widehat{z}_0\right) = \mathcal{N}(\sqrt{1-\bar{\beta}_t^{rev}}\widehat{z}_0,\ \bar{\beta}_t^{rev}\widetilde{\epsilon}_t)$, and p_θ denotes the denoising model. Afterwards, variance ϵ_t added during forward process $z_t \sim q(z_t|z_0, t)$ is estimated according to

$$\widetilde{\epsilon}_t = (z_t - \sqrt{1-\bar{\beta}_t^{train}}\widehat{z}_0) \Big/ \sqrt{\bar{\beta}_t^{train}}. \tag{5}$$

The next latent z_{t-1} is deterministicly sampled following

$$\widehat{z}_{t-1} = \sqrt{1-\bar{\beta}_{t-1}^{rev}}\widehat{z}_0 + \sqrt{\bar{\beta}_{t-1}^{rev}}\widetilde{\epsilon}_t. \tag{6}$$

Now consider the forward process on $(x, y, t-1)$, we have

$$z_{t-1} = \sqrt{1-\bar{\beta}_{t-1}^{train}}z_0 + \sqrt{\bar{\beta}_{t-1}^{train}}\epsilon_{t-1}, \tag{7}$$

where $\epsilon_{t-1} \sim \mathcal{N}(0, \mathbf{I})$. During inference, there exists non-negligible prediction error, given that we couldn't reach exact accuracy during inference. Denote σ_{t-1} the reconstruction error, we could rewrite Eq.(6) into the following form:

$$\widehat{z}_{t-1} = \sqrt{1-\bar{\beta}_{t-1}^{rev}}z_0 + (\sqrt{1-\bar{\beta}_{t-1}^{rev}}\sigma_{t-1} + \sqrt{\bar{\beta}_{t-1}^{rev}}\widetilde{\epsilon}_t). \tag{8}$$

Given that we could not achieve 100% inference accuracy, such predicted error would be addressed as part of added noise in training. We could thus improve sampling by preventing misalignment: the input $\hat{z}_t$, given the non-negligible prediction error σ, should not exceed the trained distributions (i.e., $\mathcal{N}(z_0; \sqrt{1-\beta_t}z_0, \beta_t^2\mathbf{I})$) and its definitive ranges.

According to Eq.(8), to prevent such misalignment, it is optimal to use a noise schedule that has an explicitly smaller variance during sampling, i.e., $\forall t \in [0, T],\ \ \beta_{rev}(t) < \beta_{train}(t)$, and therefore the vanilla setting ($\beta_{rev} \equiv \beta_{train}$) in DDIM (Song, Meng, and Ermon 2021) is sub-optimal in terms of aligning training and sampling. From this perspective, we reveal that noise factor (Gao et al. 2022) directly benefits from smaller β in sampling, and asymmetric time intervals is equivalent to taking a smaller β: $\beta_{t-\Delta}$ than β_t during sampling, thus we show that they are complementary in terms of misalignment prevention during sampling.

Connection Between the Two Limitations

For a unified comprehension, we make the following concluding remarks on the connections between the two limitations above. Recall the reverse diffusion process when we first call $f_\theta\left(z_t, x, t\right)$ to obtain a initial $\hat{z}_0$ prediction from pure Gaussian, then call $f_\theta\left(z_t, \hat{z}_0, x, t\right)$ along the diffusion trajectory. When f_θ is *degraded* during training, its denoising capability is thereby hindered, resulting in sub-par prediction accuracy of $\hat{z}_0$ and a greater reconstruction error σ. The progressive accumulation of σ along the trajectory results in an *exacerbated misalignment*, hampering the performance of diffusion.

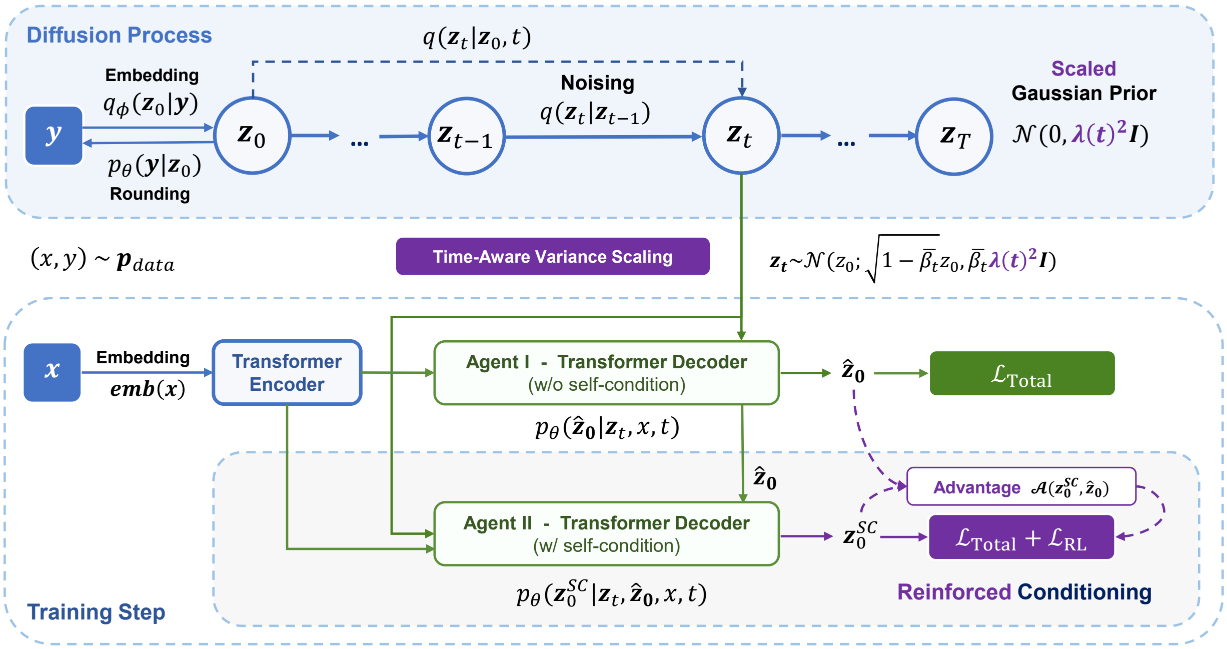

Diffusion Process
$q(z_t|z_0,t)$
Embedding
$q_\phi(z_0|y)$
y
$p_\theta(y|z_0)$
Rounding
z_0
...
z_{t-1}
Noising
$q(z_t|z_{t-1})$
z_t
...
z_T
Scaled Gaussian Prior
$\mathcal{N}(0,\lambda(t)^2 I)$
$(x,y) \sim p_{data}$
Time-Aware Variance Scaling
$z_t \sim \mathcal{N}(z_0; \sqrt{1-\bar{\beta}_t} z_0, \bar{\beta}_t \lambda(t)^2 I)$
x
Embedding
$emb(x)$
Transformer Encoder
Agent I - Transformer Decoder (w/o self-condition)
$p_\theta(\hat{z}_0|z_t,x,t)$
$\hat{z}_0$
$\mathcal{L}_{\text{Total}}$
$\hat{z}_0$
Advantage $\mathcal{A}(z_0^{SC},\hat{z}_0)$
Agent II - Transformer Decoder (w/ self-condition)
z_0^{SC}
$\mathcal{L}_{\text{Total}} + \mathcal{L}_{\text{RL}}$
$p_\theta(z_0^{SC}|z_t,\hat{z}_0,x,t)$
Reinforced Conditioning
Training Step

Figure 2: Illustration of TREC, including Reinforced Conditioning and Time-Aware Variance Scaling.

Methods

Our primary motivation is to empower text diffusion *as a whole* through mitigating degradation and misalignment. For the former, we design **Reinforced Conditioning**, leveraging reinforcement signals to reward quality improvements and enforce the model to better utilize information within noised latent. For the latter, we propose **Time-Aware Variance Scaling** by increasing model's robustness through accommodating potential sampling errors during training.

Combining the above, we propose TREC , namely **T**ext Diffusion model with **Re**inforced **C**onditioning. The design of TREC is illustrated in Figure 2. For a pair (x, y) during training, we encode x with a transformer encoder, and y to z_t with word embedding and forward diffusion process. Afterwards, we calculate the advantage from self-conditioning, then back-propagate through the RL objective. Meanwhile, the variance in the forward diffusion process is scaled with a time-aware factor to ensure $\forall t \in [0, T], \beta_{rev}(t) < \beta_{train}(t)$.

Reinforced Conditioning

In this subsection, we provide a RL-based solution to mitigate the degradation of self-conditioning during training.

Environment and Agents We define the environment as conditioned sequence generation task (i.e., $p(y|x)$), with forward diffusion process. For each training step, we first sample data pair $(x, y) \sim p_{data}$ and diffusion time step $t \sim U(0, T)$, then embed x via transformer encoder and y via forward diffusion (i.e., $z_0 \leftarrow q_\phi(z_0|y)), z_t \leftarrow q(z_t|z_0, t)$), as illustrated in Figure 2. We then employ the decoder with and without self-conditioning as two separate agents, namely SC and non-SC agent respectively. Note that they share a same set of parameters θ (as transformer decoder), but own a different set of policy given their diverse in input conditions. Policies for the non-SC and SC agent could be formalized as:

$$\begin{aligned} \pi_\theta^v &:= \arg\max_{\hat{z}_0} p_\theta(\hat{z}_0|z_t, x, t); \\ \pi_\theta^{SC} &:= \arg\max_{z_0^{SC}} p_\theta(z_0^{SC}|z_t, \widehat{z}_0, x, t), \end{aligned} \tag{9}$$

with which each agent takes actions to predict starting latent z_0 with their input conditions.

Reward and Training Objective In designing the training objective, we start from a clear motivation - to tackle the degeneration of self-conditioning by directly rewarding quality improvements and penalizing degrades. To achieve this, we first evaluate the quality of actions: $\widehat{z}_0$ from the non-SC agent π_θ^v, and z_0^{SC} from the self-conditioning agent π_θ^{SC} with a tractable evaluation metric (i.e., BLEU). Then, we could estimate the advantage of SC agent over non-SC agent as

$$\mathcal{A}(z_0^{SC}, \widehat{z}_0) = \text{clip}(R(z_0^{SC}) - R(\widehat{z}_0), -\epsilon, \epsilon). \tag{10}$$

Inspired by (Schulman et al. 2017), we clip the estimated advantages w.r.t. a clipping threshold ϵ to improve training stability of diffusion. The goal of TREC training is to minimize the negative expected advantage:

$$\begin{aligned} &\mathcal{L}_{RL}(\theta) = \\ &-\mathbb{E}_{(x,y)\sim p_d, t\sim U(0,T), z_0^{SC}\sim\pi_\theta^{SC}, \widehat{z}_0\sim\pi_\theta^v} \left[\mathcal{A}(z_0^{SC}, \widehat{z}_0)\right]. \end{aligned} \tag{11}$$

Leveraging REINFORCE (Williams 1992) algorithm, we could thus compute the gradient estimations of Eq.(11) using a batch of Monte-Carlo samples as follows:

$$\nabla_\theta \mathcal{L}_{RL}(\theta) \approx -\frac{1}{N} \sum_{i=1}^{N} \mathcal{A}_i(z_0^{SC}, \widehat{z}_0) \nabla_\theta \log p_\theta(y|z_0^{SC}). \tag{12}$$

Methods	Machine Translation		Paraphrase	Question Generation
	IWSLT14 De-En	**WMT14** En-De	**QQP**	**Quasar-T**
Transformer (Vaswani et al. (2017)) ($b = 1$)	32.76*	26.37*	30.14‡	16.73‡
Transformer (Vaswani et al. (2017)) ($b = 5$)	33.59*	27.37*	30.86‡	17.45‡
GPVAE-Finetuned T5 (Du et al. (2022))	-	-	24.09†	12.51†
Levenshetein (Gu, Wang, and Zhao (2019)) ($b = 1$)	-	27.27	22.68†	9.30†
CMLM (Ghazvininejad et al. (2019)) ($b = 10$)	33.08	27.03	24.90	7.69
DiffuSeq (Gong et al. (2023)) ($b = 10$)	28.78*	15.37*	24.13	17.31
SeqDiffuSeq (Yuan et al. (2022)) ($b = 10$)	30.03*	17.14*	24.32	17.54
DiNoiSer (Ye et al. (2023)) ($b = 5 \| b = 10$)	32.23	26.08	26.07	-
DiNoiSer (Ye et al. (2023)) ($b = 50 \| b = 20$)	32.25	26.29	25.42	-
Difformer (Gao et al. (2022)) ($b = 5 \| b = 10$) ‡	32.01	26.89	30.58	19.55
Difformer (Gao et al. (2022)) ($b = 20$) ‡	32.80	27.23	30.82	20.11
TREC ($b = 5 \| b = 10$)	32.55	27.05	33.19	21.19
TREC ($b = 20$)	**33.31**	**27.55**	**33.26**	**21.37**

Table 1: BLEU results on sequence generation tasks. 'b' denotes the beam size for AR Transformer, and the total number of samples used in candidate selection (reranking) for NAR and Diffusion models. $(b = u|b = v)$ denotes a beam size of u and v for the first and last two tasks. We highlight BLEU of the best Non-AR methods in bold. * and † indicates baseline scores quoted from Gao et al. (2022) and Gong et al. (2023), respectively. ‡ refers to results from our own implementations and experiments.

During training, following Chen, Zhang, and Hinton (2023), we take a 50% rate for self-conditioned training. For training steps w/o self-conditioning, we take Eq.(3) as training objective, and plug in $\mathcal{L}_{RL}$ (Eq.(12)) when training w/ self-conditioning. By plugging $\mathcal{L}_{RL}$ into our total objective, we directly mitigate the degradation by providing a clear motivation and guidance on quality gains of z_0^{SC} over $\widehat{z}_0$, thus preventing it from regression to simple repetition of $\widehat{z}_0$ caused by the gradual increasing of $\widehat{z}_0$'s quality during training.

Time-Aware Variance Scaling

Time-Aware Variance Scaling To alleviate the misalignment brought by the accumulation of prediction error during sampling, we propose an simple but effective method, namely time-aware variance scaling. Specifically, we scale the variance in the forward diffusion process to ensure $\beta_{rev}(t) < \beta_{train}(t)$, with a **time-aware** factor $\lambda(t) = k_1 + k_2 t$, where k_1, k_2 denotes hyperparameters of scaling factor. Then, each forward diffusion steps could be expressed as:

$$q\left(z_t|z_0, t\right) = \mathcal{N}\left(z_t; \sqrt{1-\bar{\beta}_t} z_0, \bar{\beta}_t \lambda(t)^2 \mathbf{I}\right). \qquad (13)$$

By scaling variance with $\lambda(t)$, we alter the Gaussian prior for different steps to $\mathcal{N}(0, \lambda(t)^2\mathbf{I})$ during training, while we still sample with the original prior $\mathcal{N}(0, \mathbf{I})$ during sampling. By enlarging the variance scale at each diffusion steps during training, we could increase our method's robustness to the scale of prediction error σ_t. Since we adapt an increased variance scale during training, we could thus improve model's generation capability by preventing misalignment during sampling. Time-dependent scaling is designed to address inconsistent difficulty of diffusion time-steps - denoising a lower noise latent is obviously easier, while reconstruction from higher noise scales (i.e., bigger t) is more challenging. With time-dependent scaling, we aim to improve further the robustness of preventing misalignment at higher noise scales. In other words, we prevent model from spending excessive effort on training 'trivial' low-noise steps, thus facilitating the sufficiency of training.

Experiments

Experiment Setup

Datasets We validate the performance of TREC on three important tasks of natural language generation (NLG). Specifically, we select tasks mainly following previous works (Gu et al. 2018; Ghazvininejad et al. 2019; Gong et al. 2023), including IWSLT14 De-En (Cettolo et al. 2014) and WMT14 En-De (Bojar et al. 2014) for translation[1], Quasar-T (Dhingra, Mazaitis, and Cohen 2017) for question generation, and Quora (QQP) (Chen et al. 2018) for paraphrase.

Baselines We compare our proposed TREC with a variety of strong autoregressive, non-autoregressive and diffusion baselines. Specifically, we choose Transformer (Vaswani et al. 2017) and GPVAE (Du et al. 2022)-Finetuned T5 (Raffel et al. 2020) for AR models, Levenshtein (Gu, Wang, and Zhao 2019), CMLM (Ghazvininejad et al. 2019) for NAR models, and for diffusion-based models we compare DiffuSeq (Gong et al. 2023), SeqDiffuSeq (Yuan et al. 2022), Difformer (Gao et al. 2022), including a latest work DiNoiSer (Ye et al. 2023).

Implementation We adapt `transformer-base` (Vaswani et al. 2017) architecture of TREC ($n_{\text{layers}} = 12$, $d_{\text{model}} = 512$,

[1] We apply Transformer-Large as teacher for knowledge distillation training in experiments.

Variants	Reinforced Conditioning	Variance Scaling	MBR$_\text{P}$ ($b = 10$)	MBR$_\text{P}$ ($b = 20$)	MBR$_\text{B}$ ($b = 10$)	MBR$_\text{B}$ ($b = 20$)
TREC (1)	✓	Time-Aware	33.19	**33.26**	32.11	32.60
(2)	×	Time-Aware	31.95	32.54	31.86	32.30
(3)	×	Fixed	30.48	30.71	29.90	30.49
(4)	×	×	28.08	28.85	27.31	28.49

Table 2: Ablation of proposed modules; Comparison of MBR re-ranking metric and candidate set sizes b on the paraphrase (QQP) task. MBR$_\text{P}$ and MBR$_\text{B}$ denote re-ranking with perplexity or BLEU, respectively. All results reported are BLEU scores.

$n_{\text{heads}} = 8$, $d_{\text{FFN}} = 2048$), and set embedding dimension $d = 128$. For IWSLT, we reduce n_{heads} and d_{FFN} to 4 and 1024. We take 2000 diffusion steps during training, 20 during sampling, and apply a $sqrt$ schedule (Li et al. 2022). For time-aware variance scaling, we pick $k_1 = 3$ and $k_2 = 7.5e$-4 based on preliminary experiments. We train our model on 4 V100 GPUs. We tokenize MT data using moses (Artetxe et al. 2018), and learn Byte-Pair Encoding (BPE) (Sennrich, Haddow, and Birch 2016). Following recent advances, we adopt length prediction (Lee, Mansimov, and Cho 2020), asymmetric decoding (Chen, Zhang, and Hinton 2023) and MBR decoding (Kumar and Byrne 2004) for candidate selection. We apply a learning rate of $5e$-4 ($2e$-4 for Quasar-T), 10K warmup steps (30K for Quasar-T), and apply the Adam (Kingma and Ba 2015) optimizer.

Overall Results

The experimental results of TREC on natural language generation tasks are shown in Table 1. As demonstrated in the table, TREC surpasses all non-autoregressive and diffusion baselines on a varity of sequence generation tasks (including machine translation, paraphrase and question generation), and also achieves better performance than the autoregressive Transformer on WMT14, QQP and Quasar-T datasets.

Knowledge distillation (KD) is an useful approach in the world of NAR models, and thus we explore TREC on machine translation tasks both w/ and w/o KD. As shown in Table 1, existing continous diffusion language models, including latest works DiNoiSer (Ye et al. 2023), fall behind well-established strong NAR baselines, CMLM (Savinov et al. 2022) for instance. While TREC demonstrates strong competitiveness in translation, surpassing CMLM on both datasets. On WMT, TREC shows good scalability to larger datasets and greater affinity to knowledge distillation, by being competitive against models in the worlds of NAR and diffusion, as well as the AR Transformer.

TREC also show its generic capability on conditioned generation by performing promisingly in question generation (Quasar-T) and paraphrase (QQP). On these tasks, previous Non-AR models fall behind the AR Transformer by a large margin, while TREC outperforms AR model significantly. These results demonstrate TREC's strong capability in generating high-quality responses with regard to input contexts.

Detailed Analysis

In this subsection, we study the effects of the two key parts: Reinforced Conditioning and Time-Aware Variance Scaling.

Mitigation of Degradation To study the effect of $\mathcal{L}_{RL}$ by directly examining the degradation trend of self-conditioning, we use the identical evaluation methods demonstrated in Eq.(4), i.e., evaluating quality advantages of self-conditioning by BLEU metric during training process. As illustrated in Figure 3, when training w/o reinforcements, the advantage $\mathcal{A}$ of SC agent over non-SC agent (Δ BLEU) first rises than drops beyond zero, indicating the degradation of self-conditioning to take place. On the contrary, by adding guidance from $\mathcal{L}_{RL}$ during training, such quality gains from self-conditioning are maintained throughout training, indicating that the trend of degradation is mitigated by reinforcement guidance.

Training Dynamics Moreover, we study the effect of $\mathcal{L}_{RL}$ from another perspective. In design, we plug $\mathcal{L}_{RL}$ into our total objective to mitigate the degradation by providing a clear motivation and guidance on quality gains of z_0^{SC} over $\widehat{z}_0$. To validate the effectiveness of such design, we start from examining the training dynamics of w/ and w/o $\mathcal{L}_{RL}$. As illustrated in Figure 3, utilizing reinforced conditioning brings lower losses in both diffusion and cross-entropy part of total loss (Eq.(3)) as training progresses, indicating that the $\mathcal{L}_{RL}$ indeed provided helpful guidance for training. Plus, we could also observe that by adding $\mathcal{L}_{RL}$, model enjoys less variance and fluctuations in both part of losses (diffusion and cross-entropy), demonstrating that efforts in preventing degradation do facilitate a stabler training process.

Ablation Study We study the effect of our proposed modules on model performance in Table 2. We first remove the reinforcement learning module (2), and BLEU scores drop consistently across all sampling settings. We further remove the time-aware variance scaling module (4), and the performance decreased significantly. To test the advantage of our time-aware scaling setting, we replace it with a fixed ratio by removing the time-aware part, and its performance (3) is inferior than time-aware scaling (2). Furthermore, we study effect of various sampling hyper-parameters (MBR re-ranking metric and sampling sizes b). As shown in Table 2, Perplexity (via a Transformer with equivalent architecture) outperforms BLEU in re-ranking. Additionally, we observed consistent improvements by increasing candidate sizes b, showing model's flexibility to trade-off between cost and quality.

Case Study We present illustrative examples on diffusion process of TREC. These cases demonstrate that TREC can generate reasonable sequences through diffusion process. The generation process reveals: (1) TREC could quickly generate

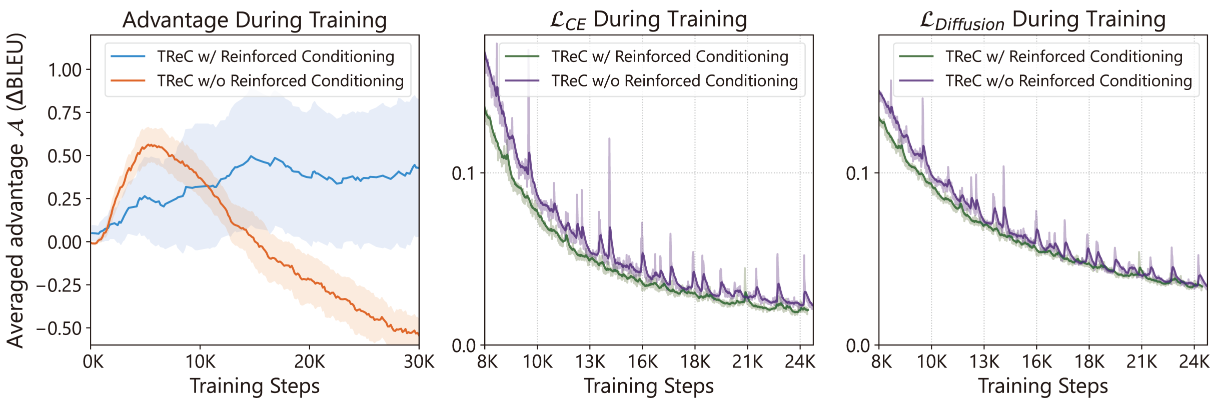


Figure 3: Degradation Tendency and Training Dynamics for TREC w/ and w/o RL on the Quasar task with 3 different seeds.

Steps	Input / Reference / Generated Sequence
Input	does long distance relationship works?
Ref.	how do i survive in a long distance relationship?
1	how do i work a long distance relationship?
2	how do i work with a long distance relationship?
3	how do i cope with a long distance relationship?
Input	if hillary clinton could not continue her presidential campaign, how would the democratic party choose a new candidate?
Ref.	if hillary clinton can no longer serve as the democratic nominee, how would her successor be chosen?
1	if hillary clinton clinton hillary clinton the the the the democratic candidate a presidential candidate?
4	how hillary clinton to the bluepresidential campaign, the democratic party choose a presidential candidate?
8	if hillary clinton fails the election, how would the democratic party choose a new candidate?

Table 3: Cases on QQP. Special tokens are omitted.

a high-quality sentence, and converge with only a few steps of iteration (Case #1). (2) TREC is capable of leveraging diffusion process to iteratively refine erroneous predictions for more challenging samples (Case #2).

Related Work

Initial researches in text diffusion models focus on remedy the discreteness of language and adapt diffusion models herein. On this front, Austin et al. (2021b) and Hoogeboom et al. (2021) design discrete diffusion based on categorical distributions, while He et al. (2022) explores building diffusion upon state absorption (i.e., masking tokens as noise injection). Li et al. (2022) first proposes to directly handle the discreteness by mapping words onto a continuous embedding space. However, the above studies only achieve unconditional or coarse-grained control on generation, whose practical applications are limited.

Consequently, subsequent works mainly focus on conditional generation, which is more practical in NLG. Improvements in the conditioning strategies are mainly categorized three-fold. The first line includes conditioning on controlling attributes, like topics or sentiments (Li et al. 2023), such as Lovelace et al. (2022) apply class embedding as conditions, Liu et al. (2022a) explore classifier guidance on the latent semantic space for styling controls. The second line applies diffusion models to text-to-text generation, i.e., conditioning on input sequences. This yields more applicable tasks like machine translation or paraphrasing, which are more challenging than conditioning on attributes (Li et al. 2023). For instance, Gao et al. (2022) propose partially noising - feeding un-noised conditioning sequences as a reference, while Gong et al. (2023); Gao et al. (2022); Ye et al. (2023) encode text condition with an encoder. The third line study conditioning on predictions of previous steps, namely self-conditioning (Chen, Zhang, and Hinton 2023; Strudel et al. 2022) to improve model performance.

Aside from conditioning strategies, other aspects that facilitates text diffusion have also been explored, including balancing embedding scale (Yuan et al. 2022; Gao et al. 2022), improving sampling methods (Chen, Zhang, and Hinton 2023; Ye et al. 2023) and utilizing pretraining (He et al. 2022; Lin et al. 2022). Unlike the existing works, this paper explores a novel conditioning method - reinforced conditioning, which utilizes reward signal to mitigate degradation effect in training text diffusion models. Plus, we propose time-aware variance scaling to better align training and sampling, through alleviating misalignment issue during sampling.

Conclusion

In this work, we thoroughly analyze the limitations of text diffusion models: degradation during training and misalignment with training during sampling, and propose TREC to empower text diffusion with reinforced conditioning and time-aware variance scaling. Our comprehensive experiments demonstrate the competitiveness of TREC on multiple language generation tasks, and provide valuable insights into improving training strategies for better diffusion models.

References

Artetxe, M.; Labaka, G.; Agirre, E.; and Cho, K. 2018. Unsupervised Neural Machine Translation. In *International Conference on Learning Representations*.

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

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

Bojar, O.; Buck, C.; Federmann, C.; Haddow, B.; Koehn, P.; Leveling, J.; Monz, C.; Pecina, P.; Post, M.; Saint-Amand, H.; et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, 12–58.

Cettolo, M.; Niehues, J.; Stüker, S.; Bentivogli, L.; and Federico, M. 2014. Report on the 11th IWSLT evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, 2–17.

Chen, T.; Zhang, R.; and Hinton, G. 2023. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*.

Chen, Z.; Zhang, H.; Zhang, X.; and Zhao, L. 2018. Quora question pairs.

Dhingra, B.; Mazaitis, K.; and Cohen, W. W. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*.

Dieleman, S.; Sartran, L.; Roshannai, A.; Savinov, N.; Ganin, Y.; Richemond, P. H.; Doucet, A.; Strudel, R.; Dyer, C.; Durkan, C.; et al. 2022. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*.

Du, W.; Zhao, J.; Wang, L.; and Ji, Y. 2022. Diverse text generation via variational encoder-decoder models with gaussian process priors. *arXiv preprint arXiv:2204.01227*.

Gao, Z.; Guo, J.; Tan, X.; Zhu, Y.; Zhang, F.; Bian, J.; and Xu, L. 2022. Difformer: Empowering Diffusion Model on Embedding Space for Text Generation. *arXiv preprint arXiv:2212.09412*.

Ghazvininejad, M.; Levy, O.; Liu, Y.; and Zettlemoyer, L. 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*, 6112–6121.

Gong, S.; Li, M.; Feng, J.; Wu, Z.; and Kong, L. 2023. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models. In *International Conference on Learning Representations*.

Gu, J.; Bradbury, J.; Xiong, C.; Li, V. O.; and Socher, R. 2018. Non-Autoregressive Neural Machine Translation. In *International Conference on Learning Representations*.

Gu, J.; Wang, C.; and Zhao, J. 2019. Levenshtein transformer. *Advances in Neural Information Processing Systems*, 32.

He, Z.; Sun, T.; Wang, K.; Huang, X.; and Qiu, X. 2022. DiffusionBERT: Improving Generative Masked Language Models with Diffusion Models. *arXiv preprint arXiv:2211.15029*.

Ho, J.; Chan, W.; Saharia, C.; Whang, J.; Gao, R.; Gritsenko, A.; Kingma, D. P.; Poole, B.; Norouzi, M.; Fleet, D. J.; et al. 2022. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Hoogeboom, E.; Nielsen, D.; Jaini, P.; Forré, P.; and Welling, M. 2021. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34: 12454–12465.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*.

Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; and Catanzaro, B. 2021. DiffWave: A Versatile Diffusion Model for Audio Synthesis. In *International Conference on Learning Representations*.

Kumar, S.; and Byrne, W. J. 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 169–176. The Association for Computational Linguistics.

Lee, J.; Mansimov, E.; and Cho, K. 2020. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *2018 Conference on Empirical Methods in Natural Language Processing*, 1173–1182. Association for Computational Linguistics.

Li, X.; Thickstun, J.; Gulrajani, I.; Liang, P. S.; and Hashimoto, T. B. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35: 4328–4343.

Li, Y.; Zhou, K.; Zhao, W. X.; and Wen, J.-R. 2023. Diffusion Models for Non-autoregressive Text Generation: A Survey. *arXiv preprint arXiv:2303.06574*.

Lin, Z.; Gong, Y.; Shen, Y.; Wu, T.; Fan, Z.; Lin, C.; Chen, W.; and Duan, N. 2022. GENIE: Large Scale Pre-training for Text Generation with Diffusion Model. *arXiv preprint arXiv:2212.11685*.

Liu, G.; Feng, Z.; Gao, Y.; Yang, Z.; Liang, X.; Bao, J.; He, X.; Cui, S.; Li, Z.; and Hu, Z. 2022a. Composable Text Controls in Latent Space with ODEs. *arXiv preprint arXiv:2208.00638*.

Liu, J.; Li, C.; Ren, Y.; Chen, F.; and Zhao, Z. 2022b. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 11020–11028.

Lovelace, J.; Kishore, V.; Wan, C.; Shekhtman, E.; and Weinberger, K. 2022. Latent Diffusion for Language Generation. *arXiv preprint arXiv:2212.09462*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Savinov, N.; Chung, J.; Binkowski, M.; Elsen, E.; and van den Oord, A. 2022. Step-unrolled Denoising Autoencoders for Text Generation. In *International Conference on Learning Representations*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1715–1725.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265. PMLR.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *9th International Conference on Learning Representations*.

Strudel, R.; Tallec, C.; Altché, F.; Du, Y.; Ganin, Y.; Mensch, A.; Grathwohl, W.; Savinov, N.; Dieleman, S.; Sifre, L.; et al. 2022. Self-conditioned embedding diffusion for text generation. *arXiv preprint arXiv:2211.04236*.

Vahdat, A.; Kreis, K.; and Kautz, J. 2021. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34: 11287–11302.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wehenkel, A.; and Louppe, G. 2021. Diffusion Priors In Variational Autoencoders. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, 5–32.

Ye, J.; Zheng, Z.; Bao, Y.; Qian, L.; and Wang, M. 2023. DINOISER: Diffused Conditional Sequence Learning by Manipulating Noises. *arXiv preprint arXiv:2302.10025*.

Yuan, H.; Yuan, Z.; Tan, C.; Huang, F.; and Huang, S. 2022. SeqDiffuSeq: Text Diffusion with Encoder-Decoder Transformers. *arXiv preprint arXiv:2212.10325*.