Diffusion Models for Non-autoregressive Text Generation: A Survey

Yifan Li[1], **Kun Zhou**[2,3], **Wayne Xin Zhao**[1,3 *] and **Ji-Rong Wen**[1,2,3]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Information, Renmin University of China
[3]Beijing Key Laboratory of Big Data Management and Analysis Methods
{liyifan0925, batmanfly}@gmail.com, francis_kun_zhou@163.com, jrwen@ruc.edu.cn

Abstract

Non-autoregressive (NAR) text generation has attracted much attention in the field of natural language processing, which greatly reduces the inference latency but has to sacrifice the generation accuracy. Recently, diffusion models, a class of latent variable generative models, have been introduced into NAR text generation, showing an improved text generation quality. In this survey, we review the recent progress in diffusion models for NAR text generation. As the background, we first present the general definition of diffusion models and the text diffusion models, and then discuss their merits for NAR generation. As the core content, we further introduce two mainstream diffusion models in existing work of text diffusion, and review the key designs of the diffusion process. Moreover, we discuss the utilization of pre-trained language models (PLMs) for text diffusion models and introduce optimization techniques for text data. Finally, we discuss several promising directions and conclude this paper. Our survey aims to provide researchers with a systematic reference of related research on text diffusion models for NAR generation. We present our collection of text diffusion models at https://github.com/RUCAIBox/Awesome-Text-Diffusion-Models.

1 Introduction

Text generation [Gatt and Krahmer, 2018] (*a.k.a.,* natural language generation) aims to generate human-like text (*i.e.,* a sequence of word tokens) given the input data (*e.g.,* sentence or keywords), enabling a wide range of real-world applications such as machine translation [Bahdanau *et al.*, 2015] and text summarization [Nallapati *et al.*, 2017]. Due to the excellent sequence modeling capacity, deep learning has become the mainstream approach to developing the backbone for text generation models, exemplified by RNN [Cho *et al.*, 2014] and transformer [Vaswani *et al.*, 2017]. More recently, pre-trained language models (PLMs) [Li *et al.*, 2021; Zhao *et al.*, 2023] further raise the performance bar of text generation. After being pre-trained on the large-scale general corpus, PLMs can be effectively fine-tuned for downstream tasks, leveraging the pre-learned rich knowledge to improve task performance. Generally, existing text generation methods mostly adopt the autoregressive way (AR) that generates the output tokens one by one. Such a way is able to capture the sequential dependency relations among tokens, but would be time-consuming when generating long texts. Thus, non-autoregressive (NAR) generation methods, which generate all tokens in parallel and greatly reduce the inference latency, have been proposed [Gu *et al.*, 2018].

However, NAR models generally underperform AR ones on text generation accuracy, since the token dependency relations cannot be well captured by the parallel generation. To narrow the performance gap, previous works have proposed various improvement techniques for NAR methods, *e.g.,* knowledge distillation [Zhou *et al.*, 2020] and large-scale pre-training [Qi *et al.*, 2021]. More recently, diffusion models [Sohl-Dickstein *et al.*, 2015; Ho *et al.*, 2020], a class of generative models that have shown superiority in image generation, are introduced into NAR text generation. In essence, diffusion models perform a multi-step denoising process to progressively convert a random noise into a data sample. To adapt to NAR text generation tasks, diffusion models iteratively refine the intermediate generated results conditioned on the input data, which are shown to be potentially more capable of handling complex control conditions in producing high-quality text [Li *et al.*, 2022]. Further, by designing proper sampling acceleration methods [Song *et al.*, 2021], diffusion models can well balance the inference latency and generation quality, leading to an improved generation ability.

Existing studies have explored two types of representative diffusion processes from image generation into text generation, *i.e., continuous diffusion* [Li *et al.*, 2022] and *discrete diffusion* [Hoogeboom *et al.*, 2021; Austin *et al.*, 2021] that perform the diffusion process in continuous latent representations and discrete text tokens, respectively. We provide a detailed illustration of these studies and the major features in Table 1. However, due to the discrete essence and complex semantics of texts, it is not easy to effectively adapt the above diffusion models to NAR text generation tasks. Prior studies have introduced or devised specific strategies to improve the original settings of diffusion models for NAR text genera-

*Corresponding Author.

Model	NAR	Diffusion space	Noise schedule	Tasks	x_0-param	PLMs	Clamping
D3PM [2021]	✓	Discrete	Mutual information	UCG	✓	✗	✗
Diffusion-LM [2022]	✓	Continuous	Sqrt	A2T	✓	✗	✓
Diffuseq [2022]	✓	Continuous	Sqrt	T2T	✓	✗	✓
SED [2022]	✓	Continuous	Cosine	UCG, A2T	✓	✗	✗
SSD-LM [2022]	✓	Continuous	Cosine	UCG, A2T	✓	✓	✗
DiffusionBERT [2022]	✓	Discrete	Spindle	UCG	✓	✓	✗
CDCD [2022]	✓	Continuous	-	T2T	✓	✗	✗
Difformer [2022]	✓	Continuous	Linear	T2T	✓	✓	✗
LD4LG [2022]	✗	Continuous	Linear	UCG, A2T	✓	✓	✗
SeqDiffuSeq [2022]	✓	Continuous	Adaptive	T2T	✓	✗	✗
Diff-Glat [2022]	✓	Discrete	-	T2T	✓	✗	✗
GENIE [2022]	✓	Continuous	-	T2T	✗	✗	✓
DINOISER [2023]	✓	Continuous	Linear	T2T	✓	✗	✗
GlyphDiffusion [2023]	✓	Continuous	-	T2T	✗	✗	✗
Diffusion-NAT [2023]	✓	Discrete	Linear	T2T	✓	✓	✗

Table 1: Comparison of existing text diffusion models. x_0-param, PLMs and Clamping denote using the x_0-parameterized loss, PLMs, and the clamping trick, respectively. **UCG**, **A2T**, **T2T** refers to **U**n**C**onditional **G**eneration, **A**ttribute-**T**o-**T**ext and **T**ext-**T**o-**T**ext, respectively.

tion, including revising the training objective, adopting noise schedules tailored for text, and integrating PLMs. Despite this progress, the field of diffusion models for text generation is still nascent, requiring a deep investigation in this line. For this purpose, a comprehensive survey that summarizes the recent advances is highly needed. To the best of our knowledge, this survey is the first literature review that concentrates on the research of diffusion models for NAR text generation.

To start with, we first briefly review diffusion models, formulate text diffusion models and describe the connections between text diffusion models and NAR models in Section 2. Then, we introduce two mainstream diffusion models used for NAR text generation in Section 3, and review four key designs of the diffusion process in Section 4, *i.e.,* denoising network, noise schedule, objective function and conditioning strategy. Next, we discuss the utilization of pre-trained language models (PLMs) for text diffusion in Section 5 and present other optimizations of diffusion models for text data in Section 6. Finally, we prospect the future directions and conclude this survey in Section 7.

2 Overview of Text Diffusion Models

Diffusion models have made remarkable progress in generating continuous data, *e.g.,* image [Dhariwal and Nichol, 2021] and audio [Kong *et al.*, 2021]. Recently, their applications in discrete text data, referred to as *text diffusion models*, are gaining growing attention. In this section, we first present a brief overview of the typical diffusion model [Ho *et al.*, 2020], then give a formulated definition of text diffusion models, and finally compare them with traditional NAR models for text generation.

2.1 Diffusion Models

Diffusion models are a class of latent variable models characterized by a forward and a reverse Markov process. The forward process $q(x_t|x_{t-1})$ gradually corrupts the data sample x_0 using random noise. The reverse process $p_\theta(x_{t-1}|x_t)$ relies on a denoising network f_θ to progressively recover a random noise into the desired data sample.

To be more specific, given a data sample $x_0 \sim q(x)$, the forward process generates a sequence of latent variables $x_1, ..., x_T$ by sampling from

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where $\beta_t \in (0,1)$ is the noise scale. Following a pre-defined noise schedule, β_t increases as the timestep grows and eventually corrupts x_0 into a random noise. Then, based on the reparameterization trick, arbitrary intermediate latent variable x_t can be sampled from x_0 in a closed form:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, \sqrt{1-\bar{\alpha}_t}\mathbf{I}), \tag{2}$$

where $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$. The reverse process is the approximation of the posterior $q(x_{t-1}|x_t)$, which can be seen as a Gaussian when β_t is small enough. In this way, the reverse process is also formulated as a Gaussian distribution:

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

where $\mu_\theta(x_t,t)$ and $\Sigma_\theta(x_t,t)$ are parameterized by a denoising networks f_θ like U-Net [Ronneberger *et al.*, 2015] or transformer [Vaswani *et al.*, 2017]. During inference, the reverse process begins with sampling noise from a Gaussian distribution $p(x_T) = \mathcal{N}(x_T; 0, \mathbf{I})$ and iteratively denoise it by $p_\theta(x_{t-1}|x_t)$ until obtaining x_0. The learning objective of diffusion models is derived from the variational lower bound of the negative log-likelihood of input x_0, denoted as:

$$\begin{aligned}\mathcal{L}_{\text{vlb}} =& \mathbb{E}_q[\underbrace{D_{\text{KL}}(q(x_t|x_0)||p_\theta(x_T))}_{\mathcal{L}_T}] - \underbrace{\log p_\theta(x_0|x_1)}_{\mathcal{L}_0} \\ &+ \mathbb{E}_q[\sum_{t=2}^{T}\underbrace{D_{\text{KL}}(q(x_{t-1}|x_t,x_0)||p_\theta(x_{t-1}|x_t))}_{\mathcal{L}_{t-1}}].\end{aligned} \tag{4}$$

The final training objective is derived from $\mathcal{L}_{t-1}$. With additional condition on x_0, the posterior of the forward process $q(x_{t-1}|x_t, x_0)$ can be calculated using Bayes theorem, then the simplified objective $\mathcal{L}_{\text{simple}}$ can be expressed as:

$$\mathcal{L}_{\text{simple}} = \sum_{t=1}^{T} \mathbb{E}_q \big[||\mu_t(x_t, x_0) - \mu_\theta(x_t, t)||^2 \big], \quad (5)$$

where μ_t is the mean of posterior $q(x_{t-1}|x_t, x_0)$. Therefore, the denoising network f_θ is trained to predict μ_t given x_t and t. Through different parameterization strategies, the prediction objective can also be the noise ϵ_t [Ho *et al.*, 2020] or original data x_0 [Li *et al.*, 2022].

2.2 Text Diffusion Models

Text diffusion models aim to gradually recover a random noise to a desired text based on the given input data. The starting noise can be discrete (*e.g.,* [MASK] tokens) or continuous (*e.g.,* random Gaussian noise), corresponding to the discrete or continuous diffusion model (Section 3). The denoising process relies on a parameterized denoising network, which is generally implemented by the transformer architecture [Vaswani *et al.*, 2017]. During training, the denoising network learns to recover the intermediate noised results based on the settings of noise schedule, objective function and conditioning strategy (Section 4). During inference, starting from a random noise $\mathcal{Y}_T$, the denoising network progressively denoises it at each step, until producing the target text. Note that at each step, following the NAR generation manner, text diffusion models predict all the latent variables in parallel. The above process can be formulated as:

$$p(\mathcal{Y}|c) = \prod_{t=0}^{T-1} \prod_{i=1}^{n} p(y_i|\hat{\mathcal{Y}}_{t+1}, c, t), \quad (6)$$

where $\mathcal{Y}$ is the target text consisting of a sequence of tokens y_i, $\hat{\mathcal{Y}}_{t+1}$ denotes the latent variables predicted at the $t+1$ timestep, c is the input condition and t denotes the timestep.

To improve the performance, it is significant to incorporate advanced NLP techniques with text diffusion models. As an important progress in the NLP area, pre-trained language models (PLMs) have been explored for integration with text diffusion models (Section 5). Moreover, a variety of optimization strategies have been proposed in existing text diffusion models to better capture the unique characteristics of text data (Section 6).

2.3 Merits of Text Diffusion Models for NAR Generation

As mentioned before, the parallel generation manner of NAR methods would greatly reduce the inference latency, but is incapable of learning the dependency relations among tokens, leading to a decreased accuracy. While, text diffusion models have several merits that can help improve the NAR generation accuracy. In this part, we show three major merits of text diffusion models, *i.e.,* constrained iterative refinement, introducing intermediate control and trading off time-cost and quality.

Constrained Iterative Refinement. Typical NAR models generate all the target tokens in parallel, hence the inference latency would be rather smaller than in AR methods. Therefore, existing works [Ghazvininejad *et al.*, 2019; Gu *et al.*, 2019] also incorporate the iterative refinement strategy to enhance the quality of the generated results. However, with the increase of the iteration steps, it also raises the problem of how to effectively control or supervise the intermediate refinement process on the discrete target tokens, restraining the improvement of the generation performance. As a promising solution, text diffusion models provide a constrained iterative refinement process for gradually enhancing the generation quality, where each step is constrained to denoise a random noise with a pre-defined variation.

Imposing Intermediate Control. In the iteration process, it is also hard to directly control the intermediate results for existing NAR methods, especially for injecting complex controlled conditions (*e.g.,* following a syntax parse tree). For text diffusion models, existing works have extensively studied injecting the control conditions in the intermediate results, by adding extra classifiers [Li *et al.*, 2022] or using classifier-free controls [Strudel *et al.*, 2022]. As theoretically and empirically proved, these approaches can better steer the intermediate prediction steps toward the generation of the target text that satisfies the control requirements.

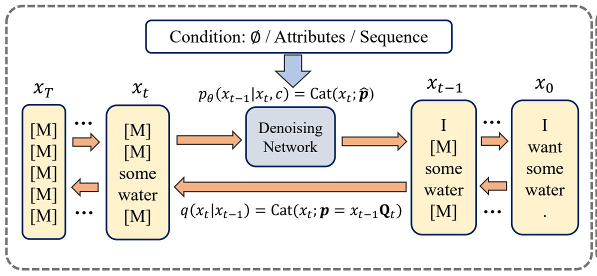
Trading off between Time Cost and Quality. During inference, existing NAR methods seek to strike a balance between time cost and quality. They mainly rely on tuning the iterative turns to achieve the goal, where decreasing the number of iterations would increase the inference speed but potentially sacrifice the generation quality. To provide a more flexible trade-off between quality and inference time, text diffusion models can adopt inference acceleration techniques, *e.g.,* DDIM [Song *et al.*, 2021]. Empirical results have shown that these methods can flexibly adjust the iteration steps with a slight decrease in the generation quality, thereby achieving a better trade-off between cost and quality.

3 Customized Diffusion Models for Text

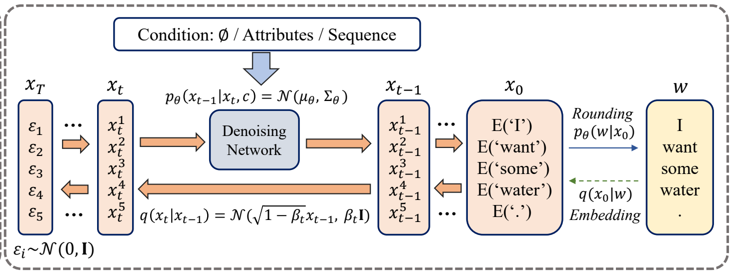
The adaptation of diffusion models to NAR text generation is challenging due to the discrete nature of text. Specially, discrete tokens cannot be directly corrupted by continuous noise, so we need to design specific adaptions of typical diffusion models for text data. In this section, we review the recent progress of diffusion models tailored for text data, which perform either the diffusion process on discrete tokens or the continuous diffusion on latent representations of tokens (*e.g.,* word embeddings).

3.1 Discrete Text Diffusion Model

The overview of the discrete text diffusion model is shown in Figure 1(a). The diffusion process in the discrete domain is first introduced by Sohl-Dickstein *et al.* [2015], which proposes a binomial diffusion process to predict the binary representations of the continuous data. Hoogeboom *et al.* [2021] further explore the diffusion process for discrete states with


Condition: ∅ / Attributes / Sequence
x_T
x_t
$p_\theta(x_{t-1}|x_t,c) = \mathrm{Cat}(x_t;\hat{\boldsymbol{p}})$
x_{t-1}
x_0
[M] [M] [M] [M] [M]
[M] [M] some water [M]
Denoising Network
I [M] some water [M]
I want some water .
$q(x_t|x_{t-1}) = \mathrm{Cat}(x_t;\boldsymbol{p}=x_{t-1}\mathbf{Q}_t)$

(a) Discrete text diffusion model.


Condition: ∅ / Attributes / Sequence
x_T
x_t
$p_\theta(x_{t-1}|x_t,c) = \mathcal{N}(\mu_\theta, \Sigma_\theta)$
x_{t-1}
x_0
w
ε_1 ε_2 ε_3 ε_4 ε_5
x_t^1 x_t^2 x_t^3 x_t^4 x_t^5
Denoising Network
x_{t-1}^1 x_{t-1}^2 x_{t-1}^3 x_{t-1}^4 x_{t-1}^5
E('I') E('want') E('some') E('water') E('.')
Rounding
$p_\theta(w|x_0)$
$q(x_0|w)$
Embedding
I want some water .
$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$
$\varepsilon_i \sim \mathcal{N}(0,\mathbf{I})$

(b) Continuous text diffusion model.

Figure 1: Illustrations about the discrete and continuous text diffusion model. In the discrete text diffusion model, "[M]" represents the [MASK] token. In the continuous text diffusion model, E(·) represents the embedding function.

a uniform transition kernel. D3PM [Austin *et al.*, 2021] proposes a general framework for diffusion models for discrete states and first tests discrete diffusion models on large-scale corpora. For discrete data $x \in 1, ..., K$, D3PM designs a transition matrix $[\mathbf{Q}_t]_{ij} = q(x_t = j|x_{t-1} = i)$ to corrupt x. The forward process in Eq. (1) now takes the following form:

$$q(x_t|x_{t-1}) = \mathrm{Cat}(x_t; \boldsymbol{p} = x_{t-1}\mathbf{Q}_t), \tag{7}$$

where x is represented by one-hot vectors and $\mathrm{Cat}(\cdot)$ is a categorical distribution over x. Following a similar derivation in the continuous diffusion process, x_t can be sampled by:

$$q(x_t|x_0) = \mathrm{Cat}(x_t; \boldsymbol{p} = x_0\bar{\mathbf{Q}}_t), \tag{8}$$

where $\bar{\mathbf{Q}}_t = \prod_{i=1}^t \mathbf{Q}_i$. By using Bayes theorem, the posterior $q(x_{t-1}|x_t, x_0)$ can be written as:

$$q(x_{t-1}|x_t, x_0) = \mathrm{Cat}\big(x_{t-1}; \boldsymbol{p} = \frac{x_t\mathbf{Q}_t^\top \odot x_0\bar{\mathbf{Q}}_{t-1}}{x_0\bar{\mathbf{Q}}_t x_t^\top}\big), \tag{9}$$

where "$\odot$" is an element-wise multiplication. Then $\mathcal{L}_{\mathrm{vlb}}$ can be calculated by accumulating the KL-divergence between every component of q and p_θ, following Eq. (4).

By designing different transition matrices $\mathbf{Q}_t$, the above framework can incorporate specific diffusion processes to generate text data. D3PM introduces a transition matrix with an absorbing state, allowing each token to be transformed into a [MASK] token with a probability β_t. During inference, D3PM starts from a sequence of full [MASK] tokens and iteratively substitutes [MASK] with word tokens until the desired text is generated.

3.2 Continuous Text Diffusion Model

The overview of the continuous text diffusion model is illustrated in Figure 1(b), where discrete tokens are first mapped to embeddings before the continuous diffusion process. Diffusion-LM [Li *et al.*, 2022] first applies continuous diffusion models to text generation, and adds an embedding step $q_\phi(x_0|w) = \mathcal{N}(\mathrm{EMB}(w), \sigma_0\mathbf{I})$ to the forward process, where $\mathrm{EMB}(w)$ is a randomly initialized embedding function that projects discrete tokens w to continuous space. For the reverse process, Diffusion-LM incorporates a rounding step $p_\theta(w|x_0) = \prod_{i=1}^n p_\theta(w_i|x_0)$ to map the final generation results to discrete tokens, where $p_\theta(w_i|x_i)$ is a softmax function. The inference process starts from a random noise and follows the typical continuous diffusion process in Section 2.1 to recover the noise to word embeddings, which are finally mapped to word tokens through the rounding step. To jointly learn the denoising network and the mapping relationship between embeddings and word tokens, Diffusion-LM reformulates the training objective in Eq. (4):

$$\mathcal{L}_{\mathrm{vlb}}^{'} = \mathbb{E}_q[\mathcal{L}_{\mathrm{vlb}} + \log q_\phi(x_0|w) - \log p_\theta(w|x_0)], \tag{10}$$

which can be further simplified as:

$$\begin{aligned}\mathcal{L}_{\mathrm{simple}}^{'} =& \mathbb{E}_q[\mathcal{L}_{\mathrm{simple}} + ||\mathrm{EMB}(w) - \mu_\theta(x_1, t_1)||^2 \\ & - \log p_\theta(w|x_0)].\end{aligned} \tag{11}$$

Different from the above works which map tokens to word embeddings, SSD-LM [Han *et al.*, 2022] utilizes a simplex representation over the vocabulary V to represent tokens. Given a token w, its simplex representation $\tilde{\boldsymbol{w}} \in \{-K, +K\}^{|V|}$ can be expressed as:

$$\tilde{w}_{(i)} = \begin{cases} +K & \text{when} \quad w = V_{(i)} \\ -K & \text{when} \quad w \neq V_{(i)} \end{cases}. \tag{12}$$

During inference, SSD-LM starts with a random noise and iteratively denoises it following the standard continuous diffusion, but adopts a logits-projection operation converting the logits to the similar almost-one-hot representations mentioned in Eq. (12) before the next decoding step.

Furthermore, GlyphDiffusion [Li *et al.*, 2023] renders the target text as a glyph graph and casts the text generation task into an image generation task, which can be naturally handled by the continuous diffusion model. Following popular settings in image diffusion models, GlyphDiffusion adopts a U-Net as the denoising network and predicts current noise with classifier-free guidance [Ho and Salimans, 2021]. Its training objective can be written as:

$$\mathcal{L} = \mathbb{E}_{x_0,\epsilon,t}(||\epsilon - \hat{\epsilon}_\theta(x_t, c)||_2^2), \tag{13}$$

where ϵ is the target noise and c is the condition text.

4 Key Designs in Diffusion Process

The denoising network and the related settings (*e.g.,* noise schedule, objective function and conditioning strategy) are crucial parts of text diffusion models and significantly impact the generation quality. In this section, we will introduce these designs and improvements for them in text diffusion models.

4.1 Denoising Network

The denoising network aims to remove noise from intermediate results in the reverse process of diffusion models. Different from vision diffusion models which adopt U-Net [Ronneberger *et al.*, 2015] as denoising networks, text diffusion models typically use transformer [Vaswani *et al.*, 2017] to better capture the dependency between tokens.

Transformer. Transformers have dominated the field of natural language processing in recent years. A transformer is an encoder-decoder neural network composed of multiple transformer layers, with each including several feed-forward networks and multi-head self-attention functions $A(\cdot)$, which can be written as:

$$A(x) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V}, \tag{14}$$

where x is the input sequence and is projected to $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ by different weight matrices and d represents the hidden dimension. Due to their strong performance in text generation, most text diffusion models adopt the encoder or the whole part of transformers as the denoising network. Some studies further utilize PLMs based on transformers (*e.g.,* BERT [Devlin *et al.*, 2019] and RoBERTa [Han *et al.*, 2022]).

4.2 Noise Schedule

The noise schedule β is a function of the noise scale depending on the timestep, which controls the frequency of different input data of the denoising network. During the training stage, the original text is corrupted by a noise β_t where t is a randomly sampled timestep and fed into the denoising network. The noise schedule affects the denoising process for recovering the target text, thereby significantly impacting the generation quality. Existing methods either adopt popular noise schedules in vision tasks or devise new noise schedules tailored to the discrete nature of text data.

Linear Schedule. DDPM [Ho *et al.*, 2020] proposes the linear schedule where β_t ranges from 10^{-4} to 0.02 linearly. Such a schedule ensures that the noise scale is relatively small at the beginning, making it easier for the denoising network to recover the data, while eventually corrupting the original data to random noise by increasing the noise scale. Difformer [Gao *et al.*, 2022] and LD4LG [Lovelace *et al.*, 2022] follow a similar setting in text diffusion models. Based on the linear schedule, DINOISER [Ye *et al.*, 2023] further introduces noise scale clipping, which only samples noises whose scale is beyond a dynamic bound. Such a way enables the sampled noise to be sufficiently large to corrupt word embeddings and force the text diffusion model to better leverage source conditions.

Cosine Schedule. Nichol and Dhariwal [2021] argue that the noise scale in linear schedule increases more quickly than necessary, making the latent variables in the last quarter of the linear schedule almost purely noise. Thus, they propose the cosine schedule by defining $\bar{\alpha}_t = \frac{f(t)}{f(0)}$, where $f(t) = \cos(\frac{t/T+s}{1+s} \cdot \frac{\pi}{2})^2$. Cosine schedule slows down the growth rate of the noise scale and is adopted by Strudel *et al.* [2022] and Han *et al.* [2022] for NAR text generation.

Mutual Information Schedule. D3PM [Austin *et al.*, 2021] designs the mutual information schedule for the discrete diffusion process by linearly interpolating the mutual information between the latent variables and the original data. Specifically, for discrete diffusion models with absorbing states, the schedule reduces to $\beta_t = (T - t + 1)^{-1}$, which is the same as the schedule in Sohl-Dickstein *et al.* [2015].

Sqrt Schedule. Diffusion-LM [Li *et al.*, 2022] observes that the nearest neighbors of words in the embedding space stay constant after corruption and attributes this phenomenon to the small initial noise scale in traditional schedules. Thus, it introduces the sqrt schedule by defining $\bar{\alpha}_t = 1 - \sqrt{t/T + s}$, which has a higher initial noise scale and increasing rate, while gradually slowing down to avoid producing too many highly corrupted latent variables. Similar methods have also been used in Gong *et al.* [2022].

Spindle Schedule. The easy-first policy [Kasai *et al.*, 2020] for NAR text generation argues that common words should be generated first to serve as the context for the subsequent generation of rare words. Therefore, DiffusionBERT [He *et al.*, 2022] proposes the spindle schedule, which assigns higher corruption probabilities to more informative tokens. As a result, rare tokens will be replaced by [MASK] tokens at the start of the forward process and recovered at the end of the denoising process.

Adaptive Schedule. Yuan *et al.* [2022] propose that the difficulty of predicting x_0 should increase linearly along with the timestep. To this end, they design an adaptive schedule by learning the relationship between the noise scale and the loss from an existing schedule (*e.g.,* sqrt schedule). During training, the noise scale is updated based on the observed loss.

4.3 Objective Function

As another key part, we also need to adapt the objective function of the denoising network in diffusion models to text generation. For example, the original μ_t-parameterized loss may not be the optimal choice for predicting word embeddings, and additional embedding and rounding steps in continuous text diffusion models also need extra loss terms. Further reparameterization can reduce the complexity of the original loss.

x_0-parameterized Loss. As mentioned in Eq. (5), the training objective for typical diffusion models can be simplified as the prediction of μ_t, the mean of the posterior $q(x_{t-1}|x_0, x_t)$. However, Li *et al.* [2022] find that this objective can cause the prediction of x_0 to fail to converge to any word embeddings, because the denoising network lacks sufficient constraints for

x_0 when predicting μ_t. Therefore, they propose to parameterize the training objective by the original text x_0, which can be written as:

$$\mathcal{L}_{\text{simple}} = \sum_{t=1}^{T} \mathbb{E}_q[||f_\theta(x_t, t) - x_0||^2], \tag{15}$$

where f_θ is the denoising network. By doing so, the training objectives of the diffusion model in every timestep can be unified into the same one that predicts x_0. Such a loss is widely adopted in follow-up text diffusion models.

Auxiliary Loss. Since continuous text diffusion models need to ground embeddings to word tokens during inference, Li *et al.* [2022] introduce a new term $\mathcal{L}_{\text{round}} = -\log p_\theta(w|x_0)$ in the objective function to better learn the mapping relationship between w and x_0, where $p_\theta(w|x_0)$ is a softmax distribution over the vocabulary. While this objective jointly trains the embedding and the diffusion process, it tends to learn a shortcut solution where every embedding is close to each other, forming an anisotropic embedding space. In fact, as x_0 is produced by adding a small amount of noise into the token embedding of w, it would be easily predicted and fails to provide enough guidance for the model training. Therefore, Gao *et al.* [2022] propose $\mathcal{L}_{\text{anchor}} = -\log p_\theta(w|\hat{x}_0)$ to substitute $\mathcal{L}_{\text{round}}$, where $\hat{x}_0$ represents the model prediction of x_0. Larger distances between $\hat{x}_0$ and w ensure that the loss can offer sufficient guidance to regularize learned embeddings.

Surrogate Loss. RDM [Zheng *et al.*, 2023] proposes to reparameterize the denoising process of discrete diffusion models by introducing step-wise routing indicators $\boldsymbol{v}_t = [v_t^{(1)}, v_t^{(2)}]$, where $v_t^{(1)} \sim \text{Bernouli}(\lambda_t^{(1)})$ selects noisy tokens for recovering and $v_t^{(2)} \sim \text{Bernouli}(\lambda_t^{(2)})$ chooses denoised tokens for corruption. Under this parameterization method, the training objective at the t-th timestep can be reformulated to:

$$\mathcal{L}_t = \mathbb{E}[-\lambda_{t-1}^{(2)} \sum_{n=1}^{N} (1 - b_{t,n}) x_{0,n}^\top \log f(x_{t,n}; \theta)], \tag{16}$$

where $b_t = 1$ if $x_t = x_0$ and else 0, and $f(x_t, \theta)$ is the predicted x_0 by the denoising network. In this way, the training objective of RDM can be implemented by a multi-class cross-entropy loss.

4.4 Conditioning Strategy

By setting different conditions c mentioned in Section 2.2, text generation tasks can be further categorized into unconditional generation, attribute-to-text generation (*e.g.,* topic control) and text-to-text generation (*e.g.,* machine translation). Existing text diffusion models design various conditioning strategies to incorporate different conditions c with denoising networks for these tasks. In this section, we will discuss these conditioning strategies.

Unconditional Generation. When setting c to empty, the task becomes unconditional text generation, where random noises are transformed to text sequences through the reverse process without other constraints. SED [Strudel *et al.*, 2022] and DiffusionBERT [He *et al.*, 2022] follow such a task setting to evaluate the basic text generation ability of text diffusion models.

Attribute-to-text Generation. When setting c to attributes such as topic or sentiment, the task becomes attribute-to-text generation. A classic method for this task is classifier-guidance, which adopts an existing classifier to provide gradient information as guidance during the generation process. Diffusion-LM [Li *et al.*, 2022] focuses on fine-grained control conditions, where c could be semantic contents (*e.g.,* five-star rating) or a syntactic parse tree. Following the plug-and-play method [Dathathri *et al.*, 2020], Diffusion-LM does not incorporate conditions directly into the denoising network. Instead, they introduce an extra attribute classifier to guide the generation results at inference time. Similar to the classifier-guided vision diffusion models [Dhariwal and Nichol, 2021], Diffusion-LM runs gradient updates on the intermediate result during inference. The reverse process in Eq. (3) can be reformulated as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta + s\nabla \log p(c|x_{t-1}), \Sigma_\theta), \tag{17}$$

where $\nabla \log p(c|x_{t-1})$ is the gradient from the classifier and s is the gradient scale. Another alternative approach, classifier-free [Ho and Salimans, 2021], explicitly introduces condition information into the denoising network. LD4LG [Lovelace *et al.*, 2022] uses class embeddings as conditions, which are combined with the latent variables via a cross-attention layer in the denoising network. Similarly, SED [Strudel *et al.*, 2022] uses condition embeddings during training, but combines them with the latent variables through the self-conditioning technique, which will be further discussed in Section 6.

Text-to-text Generation. When setting c to text such as token sequences or passages, the task becomes text-to-text generation, *e.g.,* machine translation and text summarization. These tasks are typically more difficult than attribute-to-text tasks, as they cannot be well controlled by simple attribute classifiers. Therefore, classifier-guidance methods are no longer a suitable choice. Diffuseq [Gong *et al.*, 2022] proposes the partially noising strategy to incorporate the condition text with the continuous diffusion process. Specifically, it concatenates the condition c and the target sequence as the input of the denoising network. During the forward process, the concatenated sequence is partially corrupted by adding noise only to the target sequence while keeping c unchanged. The reverse process begins with the concatenation of c and random noise, with the condition c unchanged during inference. Gao *et al.* [2022] and Yuan *et al.* [2022] propose to use an integral transformer as the denoising network. The encoder is used to generate embedding representations of c, and the decoder combines the corrupted target sequence embedding and the embedding of c via cross-attention to predict the original target sequence.

5 Utilization of Pre-trained Language Models

Since PLMs have achieved remarkable performance on various text generation tasks, we can integrate PLMs into text diffusion models to improve generation performance. In this section, we will introduce existing works that incorporate PLMs with text diffusion models.

5.1 PLMs as Denoising Networks

The denoising network of discrete diffusion models aims to recover the original sequence corrupted by [MASK] tokens, which is similar to the pre-training tasks of existing PLMs, *e.g.,* masked language model [Devlin *et al.*, 2019]. Therefore, it is promising to adopt PLMs as the denoising network in discrete text diffusion models. DiffusionBERT [He *et al.*, 2022] uses a pre-trained BERT as the denoising network. However, the denoising network is typically conditioned on timesteps for prediction, while PLMs do not encounter the timestep information during pre-training. To tackle this issue, DiffusionBERT introduces time-agnostic decoding, which does not explicitly incorporate timestep information in prediction but lets the model infer it based on the number of corrupted tokens. To better adapt text diffusion models to conditional text generation tasks, Diffusion-NAT [Zhou *et al.*, 2023] adopts a pre-trained BART [Lewis *et al.*, 2020] with a revised NAR decoding process as the denoising network. The condition and the corrupted text are fed to the encoder and decoder of BART, respectively. As for continuous text diffusion models, SSD-LM [Han *et al.*, 2022] utilizes a pre-trained RoBERTa [Liu *et al.*, 2019] as the denoising network to accelerate the convergence.

5.2 Diffusion on PLM's Latent Space

Latent diffusion models [Rombach *et al.*, 2022] conduct the diffusion process in the latent space of a pre-trained image autoencoder and achieve impressive results in the task of text-guided image generation. Similar approaches are also applied in text diffusion models. LD4LG [Lovelace *et al.*, 2022] learns a text diffusion model in the latent space of a pre-trained BART [Lewis *et al.*, 2020]. During training, the BART encoder converts the text to embeddings, which are then corrupted and recovered through a continuous diffusion process. During inference, the denoising network recovers embeddings from random noise, which are then decoded into text by the BART decoder. LatentOps [Liu *et al.*, 2022] utilizes a pre-trained GPT-2 [Radford *et al.*, 2019] to map latent vectors obtained by an ODE sampler back to the discrete text space. Difformer [Gao *et al.*, 2022] initializes the embeddings from `bert-base` and `bert-base-multilingual` for text summarization and machine translation respectively.

5.3 Revising Pre-training Tasks of PLMs

Although PLMs can offer an effective model initialization and accelerate convergence for text diffusion models, the latent space of PLMs may be a sub-optimal choice for diffusion models due to the discrepancy in training objectives. Thus, several recent works focus on revising the pre-training tasks of PLMs and pre-train new PLMs specifically for text diffusion models. GENIE [Lin *et al.*, 2022] designs a pre-training task similar to the masked language model task, called continuous paragraph denoising (CPD) and pre-trains a text diffusion model from scratch. Given a document d, CPD replaces a paragraph p in it with [MASK] and passes the corrupted document d' through an encoder to obtain a context representation. Then the paragraph p is corrupted and fed into the denoising network along with the context representation to predict the noise added during the diffusion process.

6 Other Improvement Strategies for Text Data

In addition to the aforementioned methods, there are many other techniques to further improve the performance of text diffusion models, which are designed for unique characteristics of text data or borrowed from diffusion models in other fields. In this section, we will review these methods.

Clamping Trick. Diffusion-LM [Li *et al.*, 2022] clamps the prediction result of the denoising network $f_\theta(x_t, t)$ to the nearest word embedding during inference before using it for the next prediction. This forces $f_\theta(x_t, t)$ to be concentrated on real words in the vocabulary, resulting in a lower rounding loss. However, the clamping trick needs to calculate the distance between every word embedding and the prediction result, which can lead to high computational costs when applied at every inference step. Considering this issue, GENIE [Lin *et al.*, 2022] only applies the clamping trick at the end of inference.

Self-conditioning. In the reverse process of standard diffusion models, the denoising network makes predictions based on the current latent variables x_t and timestep t. Analog Bits [Chen *et al.*, 2022] further includes the previous prediction of the original data $\tilde{x}_0$ as the input to the denoising network, which is referred to as self-conditioning. In practice, self-conditioning takes the concatenation of x_t and $\tilde{x}_0$ as the input to the denoising network. However, during training, $\tilde{x}_0$ is not available as it is in the inference stage. Therefore, Analog Bits uses $\hat{x}_0 = f_\theta(x_0, \emptyset, t)$ as the approximation to $\tilde{x}_0$, which is applied on half of the samples for self-conditioning, while setting the remaining samples to zero. Self-conditioning has been shown to remarkably improve the generation quality in practice and has been utilized in text diffusion models [Dieleman *et al.*, 2022; Gao *et al.*, 2022; Lovelace *et al.*, 2022]. Following a similar motivation, Diffusion-NAT [Zhou *et al.*, 2023] proposes the self-prompting strategy, which takes the intermediate generated result as the prefix of the original condition text. Such a strategy can be repeated multiple times, reducing the inference difficulty at the early stage.

Semi-NAR Generation. SSD-LM [Han *et al.*, 2022] introduces a semi-NAR decoding strategy. It generates a token block of size B iteratively by concatenating a random noise with previously generated blocks as input. The generated block is then concatenated to the previous context to create a new context. The above process is repeated until the maximum desired length is reached. Such a generation strategy compensates for the lack of dependencies in the NAR generation of text diffusion models.

Additional Normalization. It has been observed that rare tokens tend to have larger norms than common tokens [Gao *et al.*, 2022], while the noise scale added on different tokens is the same in existing diffusion models. Thus, rare tokens require more diffusion steps to be fully corrupted. To address this issue, Difformer introduces a layer normalization module to constrain the scale of word embeddings to the same level.

Timestep Sampling. During training, most existing works randomly sample the timesteps and the corresponding noise to corrupt the original data. However, existing studies [Nichol and Dhariwal, 2021; Gong *et al.*, 2022] find that sampling t uniformly will add noise to the objective function. Hence, they propose the importance timestep sampling:

$$\mathcal{L}_{\text{vlb}} = \mathbb{E}_{t\sim p_t}\left[\frac{\mathcal{L}_t}{p_t}\right], p_t \propto \sqrt{\mathbb{E}[\mathcal{L}_t^2]}, \sum_{t=0}^{T-1} p_t = 1. \tag{18}$$

Since this method assigns higher weights to timesteps that incur larger losses, it has shown to be effective to stabilize the training process.

7 Conclusion and Future Directions

This paper presents an overview of recent progress in text diffusion models. We review both the discrete and continuous text diffusion models, and discuss key designs in the diffusion process. We also summarize the utilization of PLMs in text diffusion models and introduce other optimization techniques for text data. To advance this field, there are several promising directions for improving text diffusion models.

Customized Noise Schedules. Existing noise schedules in text diffusion models are mainly borrowed from image generation tasks, which generally treat all tokens equally in the forward or denoising process. As a result, they tend to neglect the difference between tokens in importance and frequency, causing some tricky problems, *e.g.,* the inaccurate generation of keywords and rare words. For this issue, DiffusionBERT [He *et al.*, 2022] has proposed the spindle schedule, which assigns higher probabilities to corrupt more informative tokens and shows substantial performance improvement. More research can further explore text-tailored and task-specific noise schedules.

Efficient and Effective Utilization of PLMs. Although existing works [He *et al.*, 2022; Lovelace *et al.*, 2022] have successfully utilized PLMs in text diffusion models, they still struggle to surpass the fine-tuned performance of the original PLMs in text generation tasks. The reason is that these PLMs are pre-trained mainly for generating texts in a sequence-to-sequence or autoregressive manner, without the consideration of the diffusion process. It also raises another problem that existing works usually require more training steps to effectively adapt PLMs into the diffusion process. Therefore, it is important to investigate how to efficiently adapt PLMs into text diffusion models and effectively inspire their pre-learned rich knowledge for performing the denoising process.

Unified Multimodal Diffusion Models. Diffusion models have achieved remarkable performance on text-to-image tasks [Ramesh *et al.*, 2022], and some recent studies also pay attention to image-to-text generation such as image captioning [Luo *et al.*, 2022]. Actually, the two lines of works generally adopt similar diffusion mechanisms or settings, but with different data formats (*i.e.,* text and image). Therefore, by unifying the way of modeling text and image, diffusion models are promising to unify and share the latent semantic spaces of both image-to-text and text-to-image generation tasks, achieving the goal of a one-for-all multimodal generation model. There are several studies [Hu *et al.*, 2022; Xu *et al.*, 2022] that have explored this idea, and also discuss the possible issues about the unified diffusion model.

Alignment with Human Values. As diffusion models own a strong capacity to generate diverse results, they may generate improper content that violates human values *e.g.,* race or gender biases [Cho *et al.*, 2022]. This problem would be more serious once PLMs are also utilized in the diffusion model, since existing PLMs are pre-trained on large-scale corpora collected from the Internet, containing sensitive personal information or offensive sentences [Gehman *et al.*, 2020]. Considering the powerful abilities of text diffusion models to impose control on the intermediate results, more efforts should be devoted to developing their strong potential to prevent the above issues or steer the detoxification of the generated content [Liu *et al.*, 2021].

Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant No. 62222215, Beijing Natural Science Foundation under Grant No. 4222027, and Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098. And this work is also partially supported by the Outstanding Innovative Talents Cultivation Funded Programs 2022 of Renmin University of China. Xin Zhao is the corresponding author.

References

[Austin *et al.*, 2021] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*, pages 17981–17993, 2021.

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

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

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734. ACL, 2014.

[Cho *et al.*, 2022] Jaemin Cho, Abhay Zala, and Mohit Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generative transformers. *CoRR*, abs/2202.04053, 2022.

[Dathathri *et al.*, 2020] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *ICLR*. OpenReview.net, 2020.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[Dhariwal and Nichol, 2021] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, pages 8780–8794, 2021.

[Dieleman *et al.*, 2022] Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H. Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, Curtis Hawthorne, Rémi Leblond, Will Grathwohl, and Jonas Adler. Continuous diffusion for categorical data. *CoRR*, abs/2211.15089, 2022.

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

[Gatt and Krahmer, 2018] Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.*, 61:65–170, 2018.

[Gehman *et al.*, 2020] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *EMNLP (Findings)*, volume EMNLP 2020 of *Findings of ACL*, pages 3356–3369. Association for Computational Linguistics, 2020.

[Ghazvininejad *et al.*, 2019] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP/IJCNLP (1)*, pages 6111–6120. Association for Computational Linguistics, 2019.

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

[Gu *et al.*, 2018] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. Non-autoregressive neural machine translation. In *ICLR (Poster)*. OpenReview.net, 2018.

[Gu *et al.*, 2019] Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. In *NeurIPS*, pages 11179–11189, 2019.

[Han *et al.*, 2022] Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. SSD-LM: semi-autoregressive simplex-based diffusion language model for text generation and modular control. *CoRR*, abs/2210.17432, 2022.

[He *et al.*, 2022] Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. *CoRR*, abs/2211.15029, 2022.

[Ho and Salimans, 2021] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[Hoogeboom *et al.*, 2021] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *NeurIPS*, pages 12454–12465, 2021.

[Hu *et al.*, 2022] Minghui Hu, Chuanxia Zheng, Heliang Zheng, Tat-Jen Cham, Chaoyue Wang, Zuopeng Yang, Dacheng Tao, and Ponnuthurai N. Suganthan. Unified discrete diffusion for simultaneous vision-language generation. *CoRR*, abs/2211.14842, 2022.

[Kasai *et al.*, 2020] Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. Non-autoregressive machine translation with disentangled context transformer. In *International conference on machine learning*, pages 5144–5155. PMLR, 2020.

[Kong *et al.*, 2021] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *ICLR*. OpenReview.net, 2021.

[Lewis *et al.*, 2020] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880. Association for Computational Linguistics, 2020.

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

[Li *et al.*, 2022] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, 2022.

[Li *et al.*, 2023] Junyi Li, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Glyphdiffusion: Text generation as image generation. *arXiv preprint arXiv:2304.12519*, 2023.

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

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[Liu *et al.*, 2021] Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *ACL/IJCNLP (1)*, pages 6691–6706. Association for Computational Linguistics, 2021.

[Liu *et al.*, 2022] Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li, and Zhiting Hu. Composable text control operations in latent space with ordinary differential equations. *CoRR*, abs/2208.00638, 2022.

[Lovelace *et al.*, 2022] Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q. Weinberger. Latent diffusion for language generation. *CoRR*, abs/2212.09462, 2022.

[Luo *et al.*, 2022] Jianjie Luo, Yehao Li, Yingwei Pan, Ting Yao, Jianlin Feng, Hongyang Chao, and Tao Mei. Semantic-conditional diffusion networks for image captioning. *CoRR*, abs/2212.03099, 2022.

[Nallapati *et al.*, 2017] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, pages 3075–3081. AAAI Press, 2017.

[Nichol and Dhariwal, 2021] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 2021.

[Qi *et al.*, 2021] 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. BANG: bridging autoregressive and non-autoregressive generation with large scale pretraining. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8630–8639. PMLR, 2021.

[Qian *et al.*, 2022] Lihua Qian, Mingxuan Wang, Yang Liu, and Hao Zhou. Diff-glat: Diffusion glancing transformer for parallel sequence to sequence learning. *CoRR*, abs/2212.10240, 2022.

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

[Ramesh *et al.*, 2022] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022.

[Rombach *et al.*, 2022] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685. IEEE, 2022.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI (3)*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.

[Sohl-Dickstein *et al.*, 2015] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2256–2265. JMLR.org, 2015.

[Song *et al.*, 2021] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*. OpenReview.net, 2021.

[Strudel *et al.*, 2022] Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, and Rémi Leblond. Self-conditioned embedding diffusion for text generation. *CoRR*, abs/2211.04236, 2022.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Xu *et al.*, 2022] Xingqian Xu, Zhangyang Wang, Eric Zhang, Kai Wang, and Humphrey Shi. Versatile diffusion: Text, images and variations all in one diffusion model. *CoRR*, abs/2211.08332, 2022.

[Ye *et al.*, 2023] Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Mingxuan Wang. DINOISER: diffused conditional sequence learning by manipulating noises. *CoRR*, abs/2302.10025, 2023.

[Yuan *et al.*, 2022] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *CoRR*, abs/2212.10325, 2022.

[Zhao *et al.*, 2023] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[Zheng *et al.*, 2023] Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *CoRR*, abs/2302.05737, 2023.

[Zhou *et al.*, 2020] Chunting Zhou, Jiatao Gu, and Graham Neubig. Understanding knowledge distillation in non-autoregressive machine translation. In *ICLR*. OpenReview.net, 2020.

[Zhou *et al.*, 2023] Kun Zhou, Yifan Li, Wayne Xin Zhao, and Ji-Rong Wen. Diffusion-nat: Self-prompting discrete diffusion for non-autoregressive text generation. *arXiv preprint arXiv:2305.04044*, 2023.