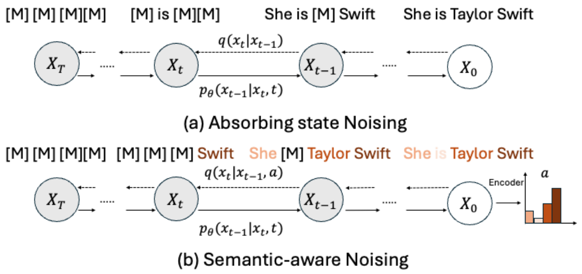
Discrete Diffusion Language Model for Long Text Summarization

Anonymous ACL submission

Abstract

While diffusion models excel at conditional generating high-quality images, prior works in discrete diffusion models were not evaluated on conditional long-text generation. In this work, we address the limitations of prior discrete diffusion models for conditional long-text generation, particularly in long sequence-to-sequence tasks such as abstractive summarization. Despite fast decoding speeds compared to autoregressive methods, previous diffusion models failed on the abstractive summarization task due to the incompatibility between the backbone architectures and the random noising process. To overcome these challenges, we introduce a novel semantic-aware noising process that enables Transformer backbones to handle long sequences effectively. Additionally, we propose CrossMamba, an adaptation of the Mamba model to the encoder-decoder paradigm, which integrates seamlessly with the random absorbing noising process. Our approaches achieve state-of-the-art performance on three benchmark summarization datasets: Gigaword, CNN/DailyMail, and Arxiv, outperforming existing discrete diffusion models on ROUGE metrics as well as possessing much faster speed in inference compared to autoregressive models.



(a) Absorbing state Noising

(b) Semantic-aware Noising

Figure 1: In contrast to conventional discrete diffusion models, we feed the full target sequence through the encoder to obtain attention scores, reflecting the relative importance of each token to the target sentence's overall semantic meaning, and use those scores to alter the absorbing probability. The higher the attention scores, the lower the probability it is absorbed to [MASK] token, as we denote as [M].

1 Introduction

Diffusion models are highly effective at generating realistic, high-quality images and have garnered considerable attention for their potential in producing discrete data types like text (Austin et al., 2021; Li et al., 2021; Lou et al., 2024), biological sequences (Avdeyev et al., 2023), and graphs (Sun and Yang, 2023; Vignac et al., 2022). Unlike autoregressive (AR) methods, diffusion-based models are not limited to sequential data generation, which could enhance long-term planning, controllable generation, and sampling speed. However, discrete diffusion methods currently underperform compared to AR models (Austin et al., 2021; Gulrajani and Hashimoto, 2024; He et al., 2023; Lou et al., 2024), particularly in the domain of language modeling. Recent methods aim to improve the framework by applying continuous diffusion to token embeddings (Gong et al., 2022; Li et al., 2022; Strudel et al., 2022; Dieleman et al., 2022) or logits (Han et al., 2022; Mahabadi et al., 2023), necessitating complex rounding schemes to convert continuous vectors into discrete tokens. These approaches also require numerous sampling iterations, resulting in slower performance compared to autoregressive models. For example, the DiffuSeq model (Gong et al., 2022) is significantly slower than a similarly scaled autoregressive baseline. Another research direction focuses on diffusion processes directly in discrete state spaces (Hoogeboom et al., 2022; Austin et al., 2021; He et al., 2023; Zheng et al., 2023), but this area is less explored and often produces inferior results in text generation. Consequently, despite their potential advantages in planning and controllable generation, diffusion models still face challenges in matching the efficiency and performance of autoregressive models in text generation tasks.

Furthermore, while discrete diffusion methods theoretically could enhance the efficiency in long-sequence processing, the capability of discrete diffusion models for conditional long-text generation tasks such as abstractive summarization remains underexplored. The task of summarizing long documents presents unique complexities compared to shorter texts. Long documents often encompass multiple ideas, subtopics, and supporting details, making it challenging to identify and distill the most salient information into a coherent summary. In this work, we find out that prior works in discrete diffusion models completely fail on abstractive text summarization, as shown later in the section. 4, due to the random absorbing noising process from D3PM (Austin et al., 2021) because the task requires a structured manner in language modeling.

Additionally, to tackle that problem, we propose a novel forward process - A semantic-aware noising process, that utilizes the Transformer encoder-decoder architecture to force the model to generate important words first, shifting the language modeling paradigm from random to important-information-first modeling. We also introduce CrossMamba to leverage Mamba (Gu and Dao, 2023) for encoder-decoder architecture, which is well-suited for the random noising process and takes advantage of Mamba's inherent efficiency for scaling to long sequences. By introducing the new decoding algorithm and the noising scheduler, our new framework can effectively model arbitrarily long textual sequences with linear processing time.

In summary, our contributions are:

- We introduce the problem of prior discrete diffusion frameworks in the long sequence-to-sequence task.
- We propose Semantic-Aware Noising Process, a novel noise scheduler, that supports the Transformer backbone to conditionally generate long sequences in an organized manner.
- We propose CrossMamba, a conditioning method that leverages Mamba to encoder-decoder architecture with outstanding speed in long contexts.
- We conduct extensive experiments on three common abstractive text summarization benchmarks, i.e. Gigaword, CNN/DailyMail, and Arxiv, and achieve state-of-the-art results compared to other discrete diffusion models. Furthermore, our framework outperforms autoregressive and continuous diffusion models in terms of decoding time.

2 Related Works

2.1 State-Space Models

A state-space model represents a system's dynamics using a set of input, output, and state variables defined through linear differential or difference equations involving system matrices (Brogan, 1974; Gu et al., 2022; Fu et al., 2023). The model computes the output by applying the state and input variables to the output equation involving the system matrices. Mamba (Gu and Dao, 2023), which belongs to the family of state-space models, has demonstrated significant capability in handling long sequences across a wide range of application domains. For instance, VisionMamba (Zhu et al., 2024) effectively leverages the Mamba kernel to encode images, achieving robust performance in image classification tasks. In the video domain, recent works (Chen et al., 2024; Liu et al., 2024) demonstrate Mamba's proficiency in managing image classification and complex spatiotemporal dynamics, offering both superior performance and favorable efficiency-performance trade-offs. In summarization task, we make the first attempt to integrate Mamba model to solve this complex language understanding task, competing with Transformer-based models.

2.2 Diffusion Models

Diffusion models are trained to progressively reverse a forward corruption process q that adds noise to clean data $\mathbf{x}$ drawn from the distribution $q(\mathbf{x})$, generating latent variables $\mathbf{z}_t$ for $t \in [0, 1]$ that represent increasingly noisy versions of $\mathbf{x}$ (Ho et al., 2020; Sahoo et al., 2023; Sohl-Dickstein et al., 2015; Song et al., 2020). The standard forward process for continuous $\mathbf{x}$ is defined as:

$$\mathbf{z}_t = \sqrt{\alpha_t}\mathbf{x} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and α_t is a noise schedule that decreases monotonically with t. The reverse diffusion model p_θ, parameterized over $\mathbf{x}$ and $\mathbf{z}_t$, is trained to maximize a variational lower bound on the log-likelihood (ELBO). With T discretization steps, defining $s(i) = \frac{(i-1)}{T}$ and $t(i) = \frac{i}{T}$, and using $D_{KL}[\cdot]$ to represent the Kullback-Leibler divergence, the Negative ELBO (NELBO) is given by (Sohl-Dickstein et al., 2015):

$$
\begin{aligned}
L_{vb} = {} & \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}|\mathbf{z}_{t(0)}) \right] \\
& + \sum_{i=1}^{T} D_{KL} \left[q(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}, \mathbf{x}) \parallel p_\theta(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}) \right] \\
& + D_{KL} \left[q(\mathbf{z}_{t(T)}) \parallel p_\theta(\mathbf{z}_{t(T)}) \right]
\end{aligned}
$$

For simplicity, we omit i from $t(i)$ and $s(i)$ in the following discussions; generally, s will denote the time step prior to t.

2.3 Discrete Diffusion Models

The application of diffusion modeling to discrete data can be categorized into two main groups. The first group consists of methods that embed discrete structures into a continuous space and then apply Gaussian diffusion (Chen et al., 2022; Dieleman et al., 2022; Gulrajani and Hashimoto, 2024; Han et al., 2022; Li et al., 2022; Strudel et al., 2022; Lovelace et al., 2024).

Methods that define a diffusion process directly on discrete structures have greater potential for substantial improvements in speed. The D3PM framework (Austin et al., 2021) introduces a Markov forward process by the multiplication of transition matrices over discrete time steps. Extending this framework to continuous time, as done in Eq. 1, utilizes continuous time Markov chain (CTMC) theory (Campbell et al., 2022). The CTMC framework further generalizes the score-matching perspective on diffusion modeling (Song and Ermon, 2019) to discrete data (Lou et al., 2024; Sun et al., 2022). Notably, SEDD (Lou et al., 2024) integrates score-based approaches with ELBO maximization, allowing for effective likelihood-based training of score-based models.

2.4 Abstractive Text Summarization

Abstractive summarization involves compressing a longer input text into a shorter output summary that retains the essential information and main ideas using new phrases and sentences rather than simply extracting key phrases or sentences from the original content. Transformer-based models have dominated this field due to the ability to capture long-range dependencies and contextual relationships within the text, thanks to self-attention mechanism (Liu and Lapata, 2019; Lewis et al., 2019; Zhang et al., 2020). However, these models fail on long abstractive summarization benchmarks due to quadratic complexity of self-attention block, which limits the number of tokens these models can handle (Keles et al., 2022). Consequently, recent works have attempted to address this issue by incorporating new attention mechanisms (Guo et al., 2022; Zaheer et al., 2021). Our work tackles this problem by leveraging the linear time complexity of the Mamba model while also maintaining comparable performance with Transformer-based models on summarization benchmarks.

3 Methodology

RDMs (Zheng et al., 2023) demonstrate that the multinominal diffusion model (Hoogeboom et al., 2021) does not decode iteratively for further refinement, making it infeasible to generate sequences in a structured strategy. Therefore, in this study, we focus on the absorbing discrete diffusion (Austin et al., 2021). To address the aforementioned issues of diffusion discrete Language Model for long text summarization, we (i) propose a novel forward process, the Semantic-aware Noising Process introduced in the section. 3.1, that helps the Transformer encoder-decoder architecture overcome the limitation of conditionally generating long sequences, and (ii) develop a new backbone architecture based on Mamba, Cross-Mamba introduced in the section. 3.2, which is well-suited for the random noising process and takes advantage of Mamba's inherent efficiency for scaling to long sequences.

Our model is broadly explained in Figure 2. We follow the design from SeqDiffuSeq (Yuan et al., 2022) promoting the encoder-decoder architecture to model the input and output text sequences. In detail, we use the encoder to process the input sequences $source$ and the decoder to model the noisy $target$ sequence. We inject time step information by adding time step embedding t. Using the encoder-decoder architecture offers computational convenience during generation because the input sequences $source$ only require one forward computation through the encoder network during the entire reverse process. Given that the reverse process requires thousands of iterations to produce high-quality output sequences, the computational resource savings can be substantial.

3.1 Semantic Aware Noising Process

The D3PM framework (Austin et al., 2021) introduces a Markov forward process $q(z_t|z_{t-1}) =$

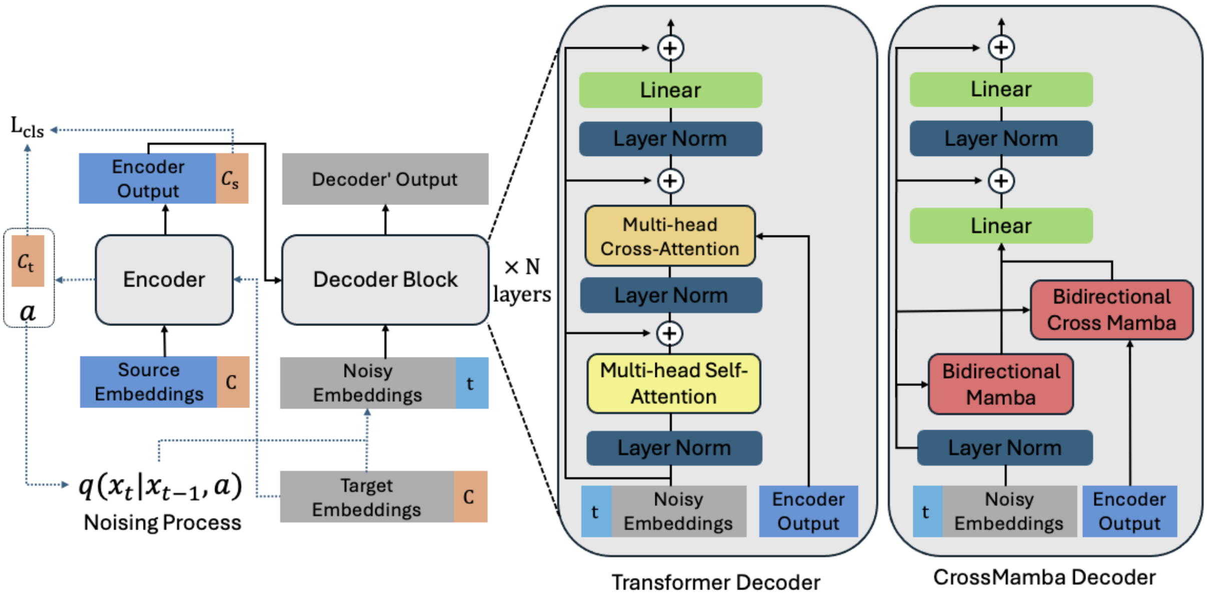


Figure 2: The model consists of an encoder and a decoder. The encoder processes the input sequence ($source$), while the decoder handles the noisy target sequence. Time step information is incorporated by adding time step embeddings t. The semantic-aware pipeline is illustrated by the blue dashes. A [CLS] token C is appended to both the source and target sequences and then passed through the encoder. The similarity loss L_{cls} is computed using the two corresponding [CLS] tokens C_s and C_t (detach). Additionally, the attention scores a from the target sequence are used in the noising process. The decoder can be standard transformer blocks that incorporate conditioning via cross-attention or CrossMamba blocks integrating conditioning with bidirectional CrossMamba.

$\mathrm{Cat}(z_t; Q_t z_{t-1})$ which is defined by the multiplication of matrices Q_t over T discrete time steps. This process results in the following marginal distributions:

$$q(z_t|x) = \mathrm{Cat}(z_t; Q_t Q_{t-1} \cdots Q_1 x)$$

These marginals represent the discrete-state form of equation 1. Specifically, each token in the sequence either remains unchanged or transitions to [MASK] with a certain probability β. The transition matrix at time step t is defined as:

$$[Q_t]_{ij} = \begin{cases} 1 & \text{if } i = j = [M], \\ 1 - \beta_t & \text{if } i = j \neq [M], \\ \beta_t & \text{if } j = [M], i \neq [M] \end{cases} \tag{2}$$

As the target sequence grows longer, the random noising process makes the conditional probability of generating tokens unpredictable. In DiffusionBERT (He et al., 2023), the spindle noise schedule is introduced to estimate the probability that the i-th token remains unchanged at step t. This probability, denoted as $\overline{\alpha}_t^i$, is computed using the following equation $\overline{\alpha}_t^i = 1 - \frac{t}{T} - S(t) \cdot \widetilde{H}(x_o^i)$ where $\widetilde{H}$ represents the entropy, which measures the information content of a random variable, x_i denotes the i-th token in the sequence, and n denotes the length of the sequence. However, this approach requires extracting the frequencies of words in the text corpus and does not have versatility across different tasks.

Built on top of the encoder-decoder, we feed-forward the full target sequence through the encoder yields attention scores, with the $[CLS]$ token's attention scores $[a_1, a_2, \ldots, a_n]$ indicating the relative importance of each input token to the sentence's overall semantic meaning. We reformulate the forward process equation to account for these attention scores:

$$[Q_t]_{ij} = \begin{cases} 1 & \text{if } i = j = [M], \\ 1 - P_t & \text{if } i = j \neq [M], \\ P_t & \text{if } j = [M], i \neq [M] \end{cases} \tag{3}$$
$$\text{with } P_t = \frac{t}{T} - \left(1 - \frac{t}{T}\right) * a_i$$

with β_t defined in Eq.2. This adjustment reflects the varying importance of different tokens at different timesteps.

Moreover, considering the semantic alignment between the input and target sequences, instead of resorting to an external pre-trained model for attention scores, both sequences are passed through the encoder. The model then calculates the cosine similarity loss between the $[CLS]$ tokens from both the source and target as:

$$L_{cls} = 1 - cos(C_s, C_t) \tag{4}$$

fostering end-to-end training, specifically training the encoder. This process enhances the model's semantic coherence between input and generated summaries, assuming that the two should bear a high degree of similarity. Specifically, to avoid trivial sentence embeddings, we detach C_t from optimization. We also add the cross-entropy loss for good predictions of the data x_0 from x_t at each time step. Thus, the total training loss is defined as:

$$L_{vb} + L_{cls} + E_{q(x_0)} E_{q(x_t|x_0)}[-log\, p_\theta(x_0|x_t)] \tag{5}$$

3.2 Cross-Mamba

State Space Models (SSMs) are built on continuous systems that transform a 1D function or sequence, $x(i) \in \mathbb{R}^L$ into $y(i) \in \mathbb{R}^L$ through an internal state $h(i) \in \mathbb{R}^N$. Mathematically, SSMs utilize the following ordinary differential equation (ODE) to represent the input data:

$$\begin{aligned} h'(i) &= Ah(i) + Bx(i) \\ y(i) &= Ch(i) \end{aligned}$$

where $A \in \mathbb{R}^{N\times N}$ is the system's evolution matrix, and $B \in \mathbb{R}^{N\times 1}, C \in \mathbb{R}^{N\times 1}$ are the projection matrices. This continuous ODE is typically discretized in modern SSMs. Mamba (Gu and Dao, 2023) represents a discrete variant of the continuous system, incorporating a timescale parameter Δ to convert the continuous parameters A, B into their discrete forms $\tilde{A}, \tilde{B}$. This conversion is generally done using the zero-order hold (ZOH) method, described by:

$$\begin{aligned} \tilde{A} &= \exp(\Delta A) \\ \tilde{B} &= (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B \\ h_i &= \tilde{A}h_{i-1} + \tilde{B}x_i \\ y_i &= Ch_i \end{aligned}$$

Mamba features a Selective Scan Mechanism (S6) as its primary SSM operator. The parameters $B \in \mathbb{R}^{B\times L\times N}, C \in \mathbb{R}^{B\times L\times N}, \Delta \in \mathbb{R}^{B\times L\times D}$, are directly derived from the input data $x \in \mathbb{R}^{B\times L\times D}$ as:

$$B, C, \Delta = s_B(x), s_C(x), s_\Delta(x)$$

with $s_B(x) = \text{Linear}_N(x)$, $s_C(x) = \text{Linear}_N(x)$, $s_\Delta(x) = \text{Broadcast}_D(\text{Linear}_1(x))$, and $\tau_\Delta =$ softplus, where Linear_d is a parameterized projection to dimension d. The choice of s_Δ and τ_Δ is motivated by their connection to RNN gating mechanisms.

Initially, we adopted a classic sequence-to-sequence RNN model, as outlined by (Sutskever et al., 2014), to create an encoder-decoder framework using Mamba. However, managing hidden states while maintaining rapid parallel computation proved challenging. To address this, we introduced a $[CLS]$ token at the end of the source sequence and placed the corresponding output token from the encoder at the start of the target sequence during the denoising stage. Quantitative analysis on a simple Quora Question Pairs (QQP) dataset, as shown in Figure 4, highlights the presence of an information bottleneck. Furthermore, we observed that both the self-attention (Vaswani et al., 2017) and Mamba (Gu and Dao, 2023) mechanisms are input-dependent, as they generate $Key, Query, Value$ matrices and B, C matrices through a linear layer, respectively. This insight led us to develop a new method called CrossMamba, which effectively addresses the information bottleneck and tailors the Mamba architecture for use in encoder-decoder models. The equations for the CrossMamba layer are expressed in equation 6.

$$\begin{aligned} & B_c, C_c, \Delta_c = s'_B(e_t), s'_C(e_t), s'_\Delta(e_t) \\ & \tilde{A_c} = \exp(\Delta_c A) \\ & \tilde{B_c} = (\Delta_c A)^{-1}(\exp(\Delta_c A) - I) \cdot \Delta_c B_c \\ & h_i^c = \tilde{A}_c h_{i-1} + \tilde{B}_c x_i \\ & y_i^c = C_c h_i \end{aligned} \tag{6}$$

with e as the encoder's output. Finally, we concatenate $[y_i, y_i^c] \in \mathbb{R}^{2\times L}$ and linear mapping the concatenation back to $\mathbb{R}^L$, similar to conventional bidirectional RNN.

CMLM (Ghazvininejad et al., 2019) deploy a linear layer as a length predictor to predict the length of the target L to avoid generating [PAD] tokens, and we utilize this predictor to adapt the cross-attention mechanism to create cross-Mamba. In detail, we first use Conv1d layers to compress the encoder's output according to the ratio of max source length and max target length. Let N be the length of the encoder's output after compression, if $N < L$, we pad the sequence to the same length L; otherwise, we take the last L tokens from the encoder's output to create the matrices B_c and C_c. The two matrices B_c and C_c are used to compute the target sequence in equation 6.

Models	Gigaword R1↑	Gigaword R2↑	Gigaword R-L↑	CNN/DailyMail R1↑	CNN/DailyMail R2↑	CNN/DailyMail R-L↑	Arxiv R1↑	Arxiv R2↑	Arxiv R-L↑
Discrete Diffusion Models									
D3PM	31.5	11.9	29.7	0.0	0.0	0.0	0.0	0.0	0.0
DiffusionBERT	29.3	9.7	26.1	0.0	0.0	0.0	0.0	0.0	0.0
RDMs	33.6	12.7	30.5	0.0	0.0	0.0	0.0	0.0	0.0
Semantic-aware	**37.2**	**13.2**	**35.4**	**32.8**	**9.5**	**29.6**	0.0	0.0	0.0
Cross-Mamba	35.5	10.6	33.7	23.8	5.3	21.1	**21.4**	**4.3**	**20.4**
Autoregressive Models									
BART	38.6	19.5	35.7	42.9	20.1	40.1	41.70	15.13	37.77
Continuous Diffusion Models									
Tess	-	-	-	41.8	18.3	35.5	-	-	-

Table 1: Comparative analysis of various diffusion text generation models on the abstractive summarization task across Gigaword, CNN/DailyMail, and Arxiv datasets. R1 and R2 are ROUGE-1 and -2 and R-L is ROUGE-L. '-' indicates results are not reported in other works.

4 Experiments

We evaluate our model on various sequence-to-sequence benchmarks and focus on text summarization datasets, including Gigaword (Rush et al., 2015), CNN/DailyMail (CNNDM) (Nallapati et al., 2016), and Arxiv (Cohan et al., 2018). We also compare the decoding speed of our models with autoregressive models. Our implementation is also based on $FairSeq$ toolkit (Ott et al., 2019) like RDMs (Zheng et al., 2023).

4.1 Implementation Details

We set the number diffusion timestep T in training to $T = 50$ and inference for evaluation to $T = 10$. We construct the encoder and decoder with 8 layers for each. Our model with the Transformer backbone has about 90M parameters and the Mamba backbone has roughly 85M parameters. We train the model using the AdamW optimizer (Loshchilov and Hutter, 2017) for 100,000 training steps, with a learning rate of 5×10^{-5}. During the initial 10,000 steps, we employ a linear warmup schedule starting from a learning rate of 5×10^{-8}. All experiments are conducted on 2 NVIDIA RTX 3090 GPUs and we use 1 for sampling.

4.2 Evaluation

Our quantitative results are presented in Table 1, showcasing ROUGE-1 (unigram), ROUGE-2 (bigram), and ROUGE-L (longest common subsequence) scores as done in prior text summarization work (Lewis et al., 2019). Generally, all previous diffusion discrete models have been unable to conditionally generate sequences for the CNN/DailyMail dataset. In contrast, our proposed methods significantly outperform them, achieving improvements of up to 32 and 30 points in ROUGE-1 and ROUGE-L scores, respectively. Although semantic-aware noising continues to struggle with the Arxiv dataset, our Cross-Mamba method maintains consistent performance on this dataset, attaining respectable scores of 21.4 in ROUGE-1 and 20.4 in ROUGE-L.

4.2.1 Decoding Speed

This section presents a performance-runtime comparison of various text generation models. Specifically, the BART decoder is causal, meaning that generation depends on the length of the target sequences rather than a constant number of steps. Continuous diffusion models typically require training with up to $T = 5000$ diffusion steps, resulting in a need for a minimum of $T > 100$ sampling steps to achieve good performance. By incorporating features from other discrete diffusion models and leveraging the efficiency of Mamba, our model achieves exceptional decoding speed

	Step	Speed	
BART	n/a	212	40.1
TESS	100	194	35.6
TESS	1000	23	39.7
Semantic-aware	2	1678	27.5
Semantic-aware	10	446	29.6
CrossMamba	2	**3223**	19.7
CrossMamba	10	869	21.1

Table 2: Decoding speed (tokens/second) of 2 backbone architectures with different numbers of diffusion timesteps, reports on the CNN/DailyMail dataset

t = 2	Stuart [M] [M] [M] [M] [M] [M] for the [M] [M] [M] [M] [M] Freedman [M] [M] his [M] [M] [M] [M] [M] . [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] [M] the [M] [M] [M]
t = 5	Stuart Freedman [M] not been [M] for the club [M] [M] . [M] Freedman [M] [M] his contract as a hero [M] [M] . Freedman has made a [M] [M] to [M] [M] the Nottingham city [M]
t = 10	Stuart Freedman has been a new deal with forest. Freedman has been on the club's new ground in the city. But Freedman has been replaced by the Nottingham City for two weeks.
t = 2	[M] [M] May [M] [M] [M] [M] [M] [M] [M] night. [M] Pacquiao will [M] [M] [M] [M] [M] [M] [M] [M] [M]. [M] [M] [M] [M] [M] [M] [M] fight on [M] [M] [M] [M] [M] [M]
t = 5	Floyd Mayweather will [M] at the [M] in [M]. He is a [M] [M] [M] [M]. the [M] [M] [M] [M] [M] [M] [M] Pacquiao [M] [M] May [M] [M] [M]. M] [M] here for the [M] [M] the news [M] [M]
t = 10	Floyd Mayweather will start at the gym in May. He is a four-time trainer. the Filipino is currently for the night. Manny Pacquiao on May 11. Click here for the latest of the news.

Table 3: Generation of the Transformer encoder-decoder model trained with the Semantic-aware Noising over time. The two different inputs are from the CNN/DailyMail dataset, with [M] representing the [MASK] token. In both examples, the model first generates important words, such as named entities (Stuart Freedman, Floyd Mayweather, Manny Pacquiao).

on the CNN/DailyMail dataset, significantly outperforming autoregressive models. As shown in Table 2, with just 10 inference steps, our model with CrossMamba runs up to 4 times faster than both BART and TESS, while the Semantic-aware method is 2 times faster. Despite having 50 diffusion timesteps for training, both CrossMamba and Semantic-aware can still deliver impressive results with only 2 inference steps, achieving speeds up to 15 times and 8 times faster than BART, respectively. In contrast, TESS experiences a marginal performance decline as the number of steps decreases from 100 to 10, and Genie's R-L performance drastically drops when the inference steps are reduced from 1000 to 100.

4.3 Analysis

In this section, we study how the semantic-aware noising process influences both the decoding stage and the training stage.

4.3.1 Effect of Semantic-aware Noising

In summarization task, the target should encapsulate the core meaning according to the source sequence. Therefore, by minimizing the similarity loss between source and target sequence will ensure the consistency between source input and generated sequence of the model. This will signal the model to produce more concise sequences, including accurately identifying and generating correct entities (such as persons, objects, etc.). As demonstrated in Table 3, the model consistently generates important words first, specifically named entities, across five different seeds, thereby highlighting the efficacy of the semantic-aware noising process.

4.4 Convergence speed

Figure 3 demonstrates that with the implementation of semantic-aware noising, the training process converges significantly faster on the QQP dataset compared to D3PM using random absorbing. At 20,000 training steps, the semantic-aware noising scheduler demonstrates performance comparable to that of random noising scheduler trained for 40,000 steps. Furthermore, at 40,000 training steps, it surpasses the random noising scheduler trained on 60,000 steps by a large margin in terms of BLEU score on QQP dataset. This finding suggests that discrete diffusion models can achieve enhanced performance through the development of appropriate generation strategies.

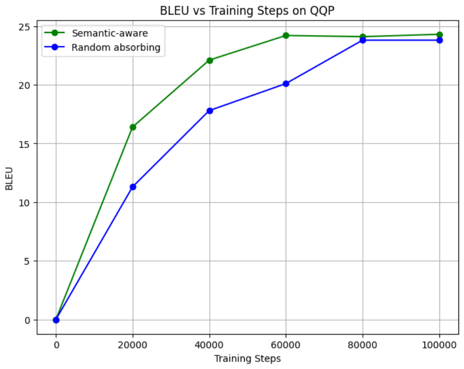


Figure 3: Curves of BLEU score vs training steps on the QQP dataset with absorbing noising and semantic-aware noising.

5 Ablation Studies

In this section, we conduct ablation studies on the effect of the similarity loss, detaching the target's $[CLS]$ token as well as the design choice of Cross-

Mamba.

5.1 Cross-Mamba Layer

To understand more about the design of Cross-Mamba, we compared it with other prominent techniques that utilize RNN-based models, including seq2seq and Diffuseq. We chose the QQP dataset for this experiment because the paraphrasing task it presents is simpler compared to tasks like summarization. Table 4 demonstrates that our method excels at connecting the source and target sequences, and almost matches the attention mechanism whereas seq2seq suffers from an information bottleneck problem, and Diffuseq requires the model to reconstruct the input.

	BLEU	R-L	bertscore
CLS seq2seq	8.3	28	0.62
Diffuseq	16.5	48	0.75
CrossMamba	**21.2**	**56.4**	**0.81**

Table 4: Different approaches adapting Mamba to discrete diffusion models on simple QQP paraphrasing dataset, showing that CrossMamba outperforms other Seq2Seq RNN techniques.

Intuitively, the attention mechanism computes a categorical distribution from K, Q, V across the sequence, whereas Mamba's B and C matrices are derived from the corresponding input tokens and encapsulate the sequence information into hidden states. Therefore, we hypothesize that Mamba's kernels are more independent than the attention kernel, enabling it to perform better during random noise processing. To test this hypothesis, we trained two different combinations of Mamba and attention mechanisms. First, we replaced Cross-Mamba in the Mamba decoder with cross-attention. Second, we tested a Transformer encoder with a CrossMamba decoder. Our results, shown in Table 5, demonstrate that both configurations underperform in handling noise compared to the Mamba encoder - CrossMamba decoder setup. This suggests that the attention mechanism is incompatible with the random noise processing scenario.

	R-1	R-2	R-L
Transformer-CrossMamba	15.8	3.1	14.7
Mamba-CrossAttention	15.1	2.9	14.0
Mamba-CrossMamba	**23.8**	**5.3**	**21.1**

Table 5: Quantitative results on different combinations of Mamba and Transformers on CNN/DailyMail dataset. The left model is the Encoder and the right model is the Decoder.

5.2 Effect of Similarity Loss

Without Similarity Loss: Without the similarity loss, there is no guarantee that the attention scores are consistent with the semantic meaning of the target and the noising process remains random, failing to dismantle the sequence in a structured manner. As shown in 6, removing similarity loss causes R-1 score drops by 6.6 points, R-2 score drops by 3.8 points, and R-L score drops by 5.8 points **Not**

	R-1	R-2	R-L
Removing	26.2	5.7	23.8
Non-detach	26.9	5.5	24.6
Semantic-aware	**32.8**	**9.5**	**29.6**

Table 6: Result of the semantic-aware noising on CNNDM dataset without the similarity loss and non-detach target sequence scenarios

Detach target sequence: Compute the gradient on both the source's $[CLS]$ and the target's $[CLS]$ shift the sequence-to-sequence task to classification, and the model can reach a trivial solution for sentence embedding, and a tremendous decrease in all metrics as illustrated in Table 6. In detail, there are marginal reductions of 5.9, 4.0, 5.0 in R-1, R-2, and R-L, respectively. These empirical evidences highlight substantial performance gains provided by semantic-aware noising.

6 Conclusion

In this paper, we introduce the Semantic-Aware Noising Process, a novel noise scheduler that supports the Transformer backbone, enabling the conditional generation of long sequences in an organized manner. Moreover, we propose CrossMamba, a conditioning method that enhances the encoder-decoder architecture with exceptional speed in handling long contexts. Our approach achieves state-of-the-art results compared to other discrete diffusion models on abstractive text summarization benchmarks, including Gigaword, CNN/DailyMail, and Arxiv datasets. Moreover, our framework surpasses both autoregressive and continuous diffusion models in terms of decoding time. This dual advantage of improved performance and reduced decoding time highlights the effectiveness and potential of our proposed methods in advancing the capabilities of discrete diffusion models for long-context sequence generation tasks.

7 Limitations

We have presented the Semantic-aware noising process and CrossMamba to tackle the main limitation of discrete diffusion models in conditional long-context sequences processing. We achieve strong empirical results relative to previous works on discrete diffusion models but still drop behind Autoregressive Models. One significant limitation is the suboptimal performance of the noising scheduler, which may be attributed to the trainability of the encoder. This issue suggests that more advanced techniques, such as distillation methods, could potentially enhance the encoder's effectiveness and overall model performance. Exploring these methods could be a promising direction for future work. Another challenge we identified is the scalability of the proposed noising scheduler. While it shows promise, it struggles with very long sequences, such as those found in the Arxiv dataset. Future research could focus on developing a more structured noising scheduler that can handle longer sequences more efficiently, such as adapting the attention weights only to the most important tokens.

References

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

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. 2023. Dirichlet diffusion score model for biological sequence generation. In *International Conference on Machine Learning*, pages 1276–1301. PMLR.

William L. Brogan. 1974. *Modern Control Theory*. Publisher Name.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. 2022. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279.

Guo Chen, Yifei Huang, Jilan Xu, Baoqi Pei, Zhe Chen, Zhiqi Li, Jiahao Wang, Kunchang Li, Tong Lu, and Limin Wang. 2024. Video mamba suite: State space model as a versatile alternative for video understanding. *Preprint*, arXiv:2403.09626.

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

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. 2022. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*.

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. 2023. Hungry hungry hippos: Towards language modeling with state space models. *Preprint*, arXiv:2212.14052.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*.

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

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *Preprint*, arXiv:2312.00752.

Albert Gu, Karan Goel, and Christopher Ré. 2022. Efficiently modeling long sequences with structured state spaces. *Preprint*, arXiv:2111.00396.

Ishaan Gulrajani and Tatsunori B Hashimoto. 2024. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. Longt5: Efficient text-to-text transformer for long sequences. *Preprint*, arXiv:2112.07916.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. 2022. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. *arXiv preprint arXiv:2210.17432*.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. 2023. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4521–4534.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Preprint*, arXiv:2006.11239.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. 2021. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. 2022. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pages 8867–8887. PMLR.

Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. 2022. On the computational complexity of self-attention. *Preprint*, arXiv:2209.04881.

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

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

Xuanlin Li, Brandon Trabucco, Dong Huk Park, Michael Luo, Sheng Shen, Trevor Darrell, and Yang Gao. 2021. Discovering non-monotonic autoregressive orderings with variational inference. *arXiv preprint arXiv:2110.15797*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *Preprint*, arXiv:1908.08345.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. 2024. Vmamba: Visual state space model. *Preprint*, arXiv:2401.10166.

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

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. *Preprint*, arXiv:2310.16834.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q Weinberger. 2024. Latent diffusion for language generation. *Advances in Neural Information Processing Systems*, 36.

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

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, and Kyunghyun Cho. 2016. Sequence-to-sequence RNNs for text summarization. *arXiv preprint arXiv:1602.06023*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. *Preprint*, arXiv:1904.01038.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Subham Sekhar Sahoo, Aaron Gokaslan, Christopher De Sa, and Volodymyr Kuleshov. 2023. Diffusion models with learned adaptive noise. *arXiv preprint arXiv:arXiv:2312.13236v2*.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.

Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

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

Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, et al. 2022. Self-conditioned embedding diffusion for text generation. *arXiv preprint arXiv:2211.04236*.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. 2022. Score-based continuous-time discrete diffusion models. *arXiv preprint arXiv:2211.16750*.

Zhiqing Sun and Yiming Yang. 2023. Difusco: Graph-based diffusion solvers for combinatorial optimization. *Advances in Neural Information Processing Systems*, 36:3706–3731.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

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

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. 2022. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*.

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

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences. *Preprint*, arXiv:2007.14062.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *Preprint*, arXiv:1912.08777.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. 2023. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. *Preprint*, arXiv:2401.09417.