Unifying Continuous and Discrete Text Diffusion with Non-simultaneous Diffusion Processes

Bocheng Li[*1,2], **Zhujin Gao**[*1,2], **Linli Xu**[†1,2]
[1]School of Computer Science and Technology, University of Science and Technology of China
[2]State Key Laboratory of Cognitive Intelligence
{bcli,gaozhujin}@mail.ustc.edu.cn, linlixu@ustc.edu.cn

Abstract

Diffusion models have emerged as a promising approach for text generation, with recent works falling into two main categories: discrete and continuous diffusion models. Discrete diffusion models apply token corruption independently using categorical distributions, allowing for different diffusion progress across tokens but lacking fine-grained control. Continuous diffusion models map tokens to continuous spaces and apply fine-grained noise, but the diffusion progress is uniform across tokens, limiting their ability to capture semantic nuances. To address these limitations, we propose **N**on-simultan**e**ous C**o**ntinuous **Diff**usion Models (NeoDiff), a novel diffusion model that integrates the strengths of both discrete and continuous approaches. NeoDiff introduces a Poisson diffusion process for the forward process, enabling a flexible and fine-grained noising paradigm, and employs a time predictor for the reverse process to adaptively modulate the denoising progress based on token semantics. Furthermore, NeoDiff utilizes an optimized schedule for inference to ensure more precise noise control and improved performance. Our approach unifies the theories of discrete and continuous diffusion models, offering a more principled and effective framework for text generation. Experimental results on several text generation tasks demonstrate NeoDiff's superior performance compared to baselines of non-autoregressive continuous and discrete diffusion models, iterative-based methods and autoregressive diffusion-based methods. These results highlight NeoDiff's potential as a powerful tool for generating high-quality text and advancing the field of diffusion-based text generation.

1 Introduction

Diffusion models have demonstrated remarkable success in generating high-quality samples in various domains, including vision (Dhariwal and Nichol, 2021; Nichol and Dhariwal, 2021; Ho and Salimans, 2021; Rombach et al., 2022) and audio (Chen et al., 2020; Kong et al., 2020). Inspired by their achievements, there has been a growing interest in applying diffusion models to text generation tasks (Li et al., 2022; Gong et al., 2022; Gao et al., 2024; Zheng et al., 2023).

The core idea behind diffusion models is to corrupt the data through a forward process and then learn to reverse this process to generate new samples. In text generation, existing diffusion models can be broadly categorized into two classes: discrete and continuous diffusion models. Discrete diffusion models treat tokens as discrete random variables and perform state transitions independently for each token using a categorical distribution. While straightforward, this approach fails to capture the continuous and fine-grained nature of language, limiting the potential benefits of multi-step generation. Continuous diffusion models, on the other hand, operate in a continuous space by mapping tokens to continuous representations, enabling more fine-grained perturbations. However, these models typically apply diffusion at the sentence level, resulting in uniform noise levels across all tokens within a sentence, restricting the model's ability to leverage contextual information and recover tokens with varying noise levels based on the surrounding context (Chen et al., 2023; Wu et al., 2024).

To address these limitations, we propose integrating the complementary strengths of discrete and continuous diffusion approaches, enabling fine-grained noise control at the token level. This unified approach aims to provide precise token-level control while maintaining continuous-valued noise distributions, which is absent in existing frameworks. While recent text diffusion models (Han et al., 2023; Gong et al., 2023; Wu et al., 2024) have made advances, they do not fully address

[*]Equal contribution.
[†]Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11530–11551
July 27 - August 1, 2025

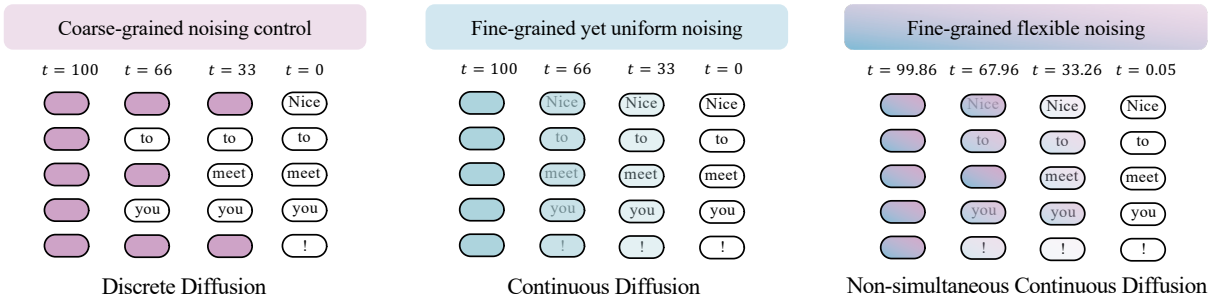

Figure 1: Comparison of the noising paradigms employed by Non-simultaneous Continuous Diffusion and two other diffusion models. The color intensity on the text tokens represents the token-level noising progress (intrinsic time τ). Discrete diffusion applies an independent but coarse-grained noising paradigm to each token within a sentence. In contrast, continuous diffusion utilizes a fine-grained noising schedule but applies it uniformly across all tokens. NeoDiff distinguishes itself by assigning an independent, fine-grained intrinsic time τ to each token, with finer noising schedule in extrinsic time t.

this requirement, necessitating a unified theoretical framework that bridges discrete and continuous diffusion paradigms through a carefully designed forward process.

Furthermore, we observe that existing approaches primarily focus on enhancing the forward process, overlooking the inherent varying difficulties in denoising different tokens and the impact of generation context. In analyzing the reverse process, we recognize that tokens with lower noise levels can guide the recovery of more heavily corrupted tokens, thereby enhancing the overall text generation quality.

In response to these challenges, we present **N**on-simultan**e**ous C**o**ntinuous **Diff**usion Models (NeoDiff), which unifies discrete and continuous diffusion models through a bi-temporal framework. The key insight is to generalize the time variable in previous diffusion models into extrinsic time t, representing the diffusion progress of the entire sentence, and intrinsic time τ, tracking the diffusion progress of each individual token. This generalization enables us to introduce a novel Poisson process as the forward process, seamlessly integrating the flexibility of discrete noise with the fine granularity of continuous noise. An overview of this noising paradigm is illustrated in Figure 1.

To optimize the reverse process, we develop a context-aware time predictor that estimates the intrinsic time τ using an adaptive modulation function to guide the denoising process. The extrinsic time schedule is further calibrated through Bayesian optimization, providing precise control over the noise distribution.

NeoDiff achieves a fine-grained, improved diffusion process in both forward and reverse directions, naturally overcoming the constraints of previous discrete and continuous diffusion models, and exhibiting superior generation quality. We evaluate NeoDiff on a diverse set of NLP tasks, including machine translation, paraphrasing, text simplification, and question generation. NeoDiff consistently outperforms previous non-autoregressive diffusion-based and iteration-based methods, as well as autoregressive diffusion baselines. Specifically, our contributions can be summarized as follows:

- We introduce NeoDiff, a unified theoretical framework that combines the advantages of discrete and continuous noise, generalizing and unifying existing text diffusion models.
- We propose the Poisson diffusion process as the forward process, enabling fine-grained corruption of text data, a context-aware time predictor that adaptively modulates the reverse process based on semantic context, and an optimized extrinsic time schedule for precise noising control.
- We conduct extensive experiments to evaluate the effectiveness of NeoDiff and compare it to existing text diffusion models. Our results highlight the advantages of our unified framework and suggest its potential to advance diffusion-based text generation.

2 Background

2.1 Diffusion Models

Diffusion models assume a gradual noise injection process over time for data samples $\boldsymbol{z}_0 \in \mathbb{R}^{N \times d}$. The forward diffusion process forms a series of latent variables $\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_T$ satisfying the Markov

property, and finally become pure Gaussian noise $\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) = \mathcal{N}\left(\boldsymbol{z}_t; \sqrt{\alpha_t}\boldsymbol{z}_{t-1}, \beta_t\mathbf{I}\right), \quad (1)$$

where $\alpha_t + \beta_t = 1$, determining the degree of noising at time t and consituting the noise schedule. The reverse process is parameterized as

$$p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t) = \mathcal{N}\left(\boldsymbol{z}_{t-1}; \mu_\theta\left(\boldsymbol{z}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\boldsymbol{z}_t, t\right)\right), \quad (2)$$

Here, $\mu_\theta(\cdot)$ and $\boldsymbol{\Sigma}_\theta(\cdot)$ are the model's estimates of the distribution mean and covariance matrix, respectively. The training objective is derived from the variational lower bound of the negative log-likelihood loss, and can be then simplified as an MSE loss (Ho et al., 2020; Li et al., 2022; Gao et al., 2024):

$$\mathcal{L}_{\text{VLB}} = \mathbb{E}\left[\|\boldsymbol{z}_\theta\left(\boldsymbol{z}_t, t\right) - \boldsymbol{z}_0\|^2 - \log p(\mathbf{z}_0 \mid \mathbf{z}_1)\right]$$

2.2 Discrete Diffusion Models

Discrete diffusion models directly model the noise on categorical distributions, discarding the assumption that the noise in latent variables follows a normal distribution in continuous space. These models typically represent data as sequences of one-hot vectors and employ a transition matrix to add noise to the data. Among them, Hoogeboom et al. (2021a) proposed a multinomial diffusion model that employs a uniform noising method. Austin et al. (2021) introduced D3PM, which employs a noising method with an absorbing state. Specifically, they added an absorbing state [MASK] to the vocabulary, which can only be entered but not exited. The remaining states, at each diffusion step, either stay in the current state or enter the absorbing state with a certain probability. Recently, Lou et al. (2024) made progress by developing score entropy, which extends score matching to discrete spaces and demonstrates substantial performance improvements. While effectively adapting diffusion models to discrete data, these methods have limitations. The discrete nature of the noise limits its expressiveness, making it difficult to capture the nuances of continuous transitions between states. This restricts the model's ability to represent gradual semantic changes or finely adjust individual token features, potentially limiting the benefits of multi-step generation.

2.3 Continuous Diffusion Models

Continuous diffusion models map discrete tokens to a continuous vector space using a mapping function, allowing the application of standard continuous diffusion processes. Analog Bits (Chen et al., 2022) uses a binary encoding scheme (int2bit : $\mathbb{Z} \to 0, 1^{\lceil \log_2 V \rceil}$) to represent token indices as binary sequences. After the reverse diffusion process, a quantization operation followed by binary decoding (bit2int : $0, 1^{\lceil \log_2 V \rceil} \to \mathbb{Z}$) recovers the token indices. Han et al. (2023) proposed a mapping function logits-generation : $\mathbb{Z} \to \mathbb{R}^V$, which transforms token indices into a probability simplex. Li et al. (2022) proposed Diffusion-LM, where the token sequence $\mathbf{y}$ is first mapped to a random representation $\mathbf{z}_0$ using a word embedding as the mean. After the reverse diffusion process, the generated vectors are rounded back to discrete tokens. Gong et al. (2022) extended this approach to sequence-to-sequence generation with DiffuSeq, which concatenates the source and target sentences and utilizes an attention mechanism to leverage source information during generation. However, a key limitation of continuous diffusion models is the uniform noise injection applied to all tokens during the forward process. This uniform noise injection hinders the model's ability to effectively leverage contextual information. Ideally, varying noise levels across tokens would allow the model to utilize less noisy tokens as context for restoring more corrupted ones, facilitating better contextual modeling.

2.4 Improvements over Previous Diffusion Models

Recent studies have explored various methods to address the limitations discussed above. Han et al. (2023) introduced a semi-autoregressive generation strategy that generates fixed-length blocks autoregressively while employing non-autoregressive iterative denoising within each block. Wu et al. (2024) proposed a hierarchical noise addition method, where noise levels increase monotonically from left to right within a sentence, enabling autoregressive generation. Gong et al. (2023) presented a hybrid approach that combines standard continuous noise with the probabilistic replacement of tokens with [MASK], integrating discrete and continuous noise. Although these studies have contributed to enhancing the forward diffusion process, their improvements did not fully achieve fine-grained noise at the token level, thus not completely addressing

the limitations of both continuous and discrete diffusion models. Also, these approaches typically employ a fixed reverse process that mirrors the forward diffusion process, without considering the varying difficulties in denoising different tokens and the impact of the actual generation context.

3 Non-simultaeous Continuous Diffusion Models

To address these limitations, we propose a unified diffusion framework called Non-simultaneous Continuous Diffusion Models (NeoDiff). Figure 2 presents an overview of NeoDiff, illustrating its architecture and key components. NeoDiff employs an Encoder-Decoder Transformer architecture(Vaswani et al., 2017), with the decoder serving as the primary component for denoising, and the encoder provides the embedding of the condition sentence $\mathbf{x}$ to a transformer-decoder-based time predictor. In the following sections, we will provide a detailed formulation of NeoDiff and demonstrate how it addresses the limitations of previous approaches.

3.1 Unified Formulation and Training Objective

We present a unified framework for diffusion models by introducing two time dimensions: extrinsic time t and intrinsic time τ. The extrinsic time t represents the global diffusion progress of the entire sentence, while the intrinsic time τ captures the diffusion progress of individual tokens.

This formulation generalizes existing approaches. We can easily derive discrete diffusion models by modeling τ as a monotonically increasing random function of t, with $\tau_t \in \{0, 1\}$, where $\tau_t = 0$ and $\tau_t = 1$ signify original and fully corrupted tokens, respectively. And continuous diffusion can be obtained by setting τ as a deterministic function that typically equals t ($\tau_t = t$). Furthermore, recent hybrid diffusion models, such as DiffuSeq-V2(Gong et al., 2023), can also be formalized under this framework by setting $\tau_t = \max(t + \tau_{\text{mask}}(t), 1)$, where $\tau_{\text{mask}}(t) \sim \text{Bernoulli}(\gamma, \bar{\beta}(t))$ and γ is the ratio of tokens replaced by [MASK] when $t = 1$.

NeoDiff defines $\tau_t \in [0, 1]$ as a continuous random function of extrinsic time $t \in [0, 1]$, enabling fine-grained control over the diffusion process. We impose boundary conditions $\tau_0 = 0$ and $\tau_1 = 1$ to guarantee token preservation at initialization and complete corruption at termination of the diffusion process.

Let $\mathbf{z} \in \mathbb{R}^d$ denote a token embedding and $\mathbf{z}_t$ its latent representation at time t, with initial and final conditions $\mathbf{z}_0 = \mathbf{z}$ and $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The forward process defines the joint distribution as:

$$
\begin{aligned}
q(\boldsymbol{z}_{>0}, \tau_{>0}|\boldsymbol{z}_0) &:= \prod_{t>0} q(\boldsymbol{z}_t, \tau_t|\boldsymbol{z}_0) \\
&= \prod_{t>0} q(\boldsymbol{z}_t|\boldsymbol{z}_0, \tau_t) q(\tau_t),
\end{aligned}
$$

where

$$
q(\boldsymbol{z}_t|\boldsymbol{z}_0, \tau_t) := \mathcal{N}\left(\boldsymbol{z}_t; \sqrt{\bar{\alpha}(\tau_t)}\boldsymbol{z}_0, \bar{\beta}(\tau_t)\mathbf{I}\right),
$$

and $\bar{\alpha}(\cdot)$ and $\bar{\beta}(\cdot)$ denote noise schedules with their domains scaled to $[0, 1]$.

Given $t' = t - \Delta t$, the reverse process is defined as

$$
\begin{aligned}
p_\theta(\boldsymbol{z}_{0:1}, \tau_{0:1}) &:= p_\theta(\boldsymbol{z}_1, \tau_1) \prod_{t'<1} p_\theta(\boldsymbol{z}_{t'}, \tau_{t'}|\boldsymbol{z}_t, \tau_t) \\
&= p_\theta(\boldsymbol{z}_1, \tau_1) \prod_{t'<1} p_\theta(\boldsymbol{z}_{t'}|\boldsymbol{z}_t, \tau_t, \tau_{t'}) p_\theta(\tau_{t'}|\boldsymbol{z}_t, \tau_t).
\end{aligned}
$$

We further parameterize the distribution of $\boldsymbol{z}_{t'}$ as

$$
p_\theta(\boldsymbol{z}_{t'}|\boldsymbol{z}_t, \tau_t, \tau_{t'}) = q(\boldsymbol{z}_{t'}|\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t, \tau_t, t), \tau_{t'}),
$$

where $\hat{\boldsymbol{z}}_0$ is the model prediction of $\boldsymbol{z}_0$.

Following Ho et al. (2020) and Li et al. (2022), we derive NeoDiff's training objective from the variational lower-bound $\mathcal{L}_{\text{VLB}}$, and with the simplified $\mathcal{L}_z$ and an anchor loss $\mathcal{L}_{\text{anchor}}$ (Gao et al., 2024) as a regularization term to avoid collapse of the embedding space, the training objective of Neodiff can be written as

$$
\begin{aligned}
\mathcal{L} &= \mathcal{L}_z + \mathcal{L}_\tau + \mathcal{L}_{\text{anchor}} && (3) \\
&= \mathbb{E}_q \Bigg[\underbrace{\|\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t, \tau_t, t) - \boldsymbol{z}_0\|^2}_{\mathcal{L}_z} && (4) \\
&\quad + \sum_{0<t'<1} \underbrace{\mathbb{KL}(q(\tau_{t'}) \| p_\theta(\tau_{t'}|\boldsymbol{z}_t, \tau_t))}_{\mathcal{L}_\tau} && (5) \\
&\quad + \underbrace{-\log p_\theta(y|\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t, \tau_t, t))}_{\mathcal{L}_{\text{anchor}}} \Bigg]. && (6)
\end{aligned}
$$

A detailed derivation can be found in Appendix A.

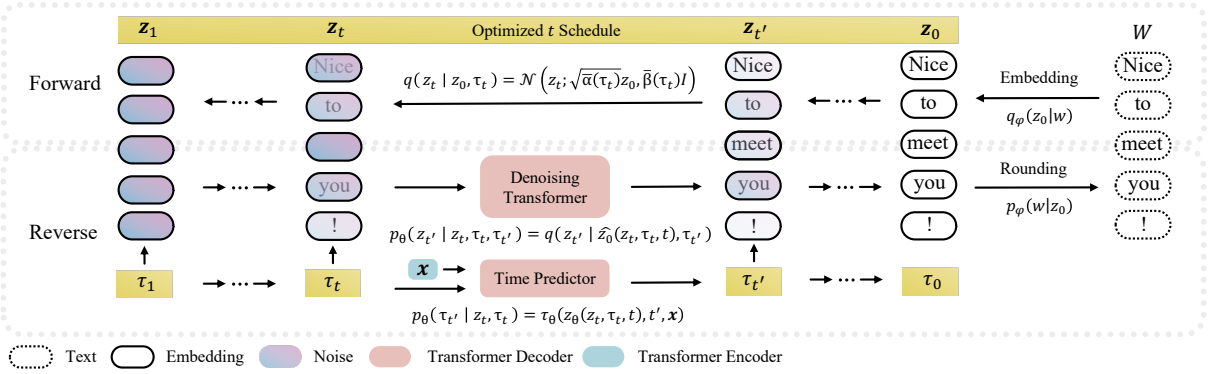

Figure 2: An overview of NeoDiff.

3.2 Fine-Grained Forward Process Using Poisson Diffusion

After establishing the unified formulation, we define a fine-grained forward diffusion process through intrinsic time τ. To quantify the diffusion progression within a single token, we introduce a discrete state function $s_t \in \{0, 1, 2, \cdots, s_{\max}\}$, where uniformly divided states represent distinct levels of the diffusion process from $s_t = 0$ (noiseless) to $s_t = s_{\max}$ (maximum noise). For an infinitesimal time interval Δt, the transition dynamics follow a Poisson process characterized by:

$$\begin{aligned}\mathbb{P}[s_t = s_{t'} + 1] &= \gamma(t)\Delta t + o(\Delta t)\\ \mathbb{P}[s_t = s_{t'}] &= 1 - \gamma(t)\Delta t + o(\Delta t),\end{aligned}$$

where $\gamma(\cdot)$ is a hyperparameter function termed the transition schedule. This formulation yields a tractable distribution for s_t:

$$s_t \sim \text{Poisson}\left(\int_0^t \gamma(\tau)\,\mathrm{d}t\right) = \text{Poisson}\left(\lambda(t)\right).$$

To ensure compatibility with the continuous-time framework of NeoDiff, we normalize the state function to $[0, 1]$ through normalization and clipping:

$$\tau_t = \text{Clip}\left(\frac{s_t}{s_{\max}}, 1\right) = \text{Clip}\left(s'_t, 1\right),$$

where $\text{Clip}(\cdot, \cdot)$ denotes the truncation operation to maintain bounded noise levels. We choose $s_{\max}$ sufficiently large to achieve fine-grained transitions between noise states, and set $\lambda(t) = k s_{\max} t$ to maintain $\mathbb{E}[s_t] = \lambda(t) \propto s_{\max}$. This design ensures that τ_t remains independent of $s_{\max}$ and reduces the process to a homogeneous Poisson process with constant transition schedule $\gamma(t)$.

However, a critical limitation of this basic formulation emerges when examining the coefficient of variation (CV) of the normalized state function $s'(t)$:

$$\text{CV} = \frac{\sqrt{\mathbb{V}[s'_t]}}{\mathbb{E}[s'_t]} = \frac{1}{\sqrt{\lambda(t)}} \propto \frac{1}{\sqrt{s_{\max}}},$$

which indicates that as $s_{\max}$ increases, the relative variation between token states diminishes proportionally to $\frac{1}{\sqrt{s_{\max}}}$. Consequently, when $s_{\max}$ becomes sufficiently large, the discreteness of the process is lost as all tokens effectively share nearly identical τ values, causing NeoDiff to degenerate into a continuous diffusion model.

To address this limitation, we further introduce a variance-controlled rescaling transformation:

$$\tau_t = \frac{\text{Clip}\left(\text{Round}\left(\frac{s_t - \lambda(t)}{\sqrt{\lambda(t)}}\sigma(t) + \lambda(t)\right), s_{\max}\right)}{s_{\max}} \tag{7}$$

Under this transformation, the variables within $\text{Clip}(\cdot, \cdot)$ follow a distribution centered at $\lambda(t)$ with variance $\sigma(t)$. To ensure that the discrete characteristics of our process remain invariant to the choice of $s_{\max}$, we set $\sigma(t) = \lambda(t)$. Since the choice of $\lambda(t)$ and $\sigma(t)$ may result in $\tau_1 \neq 1$, we truncate τ_t to 1 for $t > t_{\max}$, where $t_{\max}$ is a predefined threshold.

3.3 Context-aware Reverse Process with Time Predictor

We propose a context-aware reverse process that explicitly models the conditional distribution $p_\theta(\tau_{t'}|\boldsymbol{z}_t, \tau_t)$, in contrast to previous approaches that simply mirror the forward process by assuming $p_\theta(\tau_{t'}|\boldsymbol{z}_t, \tau_t) = q(\tau_{t'})$. This explicit modeling enables adaptive denoising based on both semantic context and noise states.

Time Predictor Design. When modeling a known distribution, researchers typically employ reparameterization tricks to model its parameters. However, in our case, the Poisson distribution's sole parameter $\lambda(t)$ is a deterministic function of t that measures the overall noise progress of the sample and is equivalent to t. To obtain the noise progression τ for each basic token, we directly treat both $p_\theta(\tau_{t'}|\boldsymbol{z}_t, \tau_t)$ and $q(\tau_{t'})$ as standard discrete distributions and learn them using cross-entropy loss, without using reparameterization tricks.

Model Input Design. While $\boldsymbol{z}_t$ could serve as an input to τ_θ, this choice would enable the model to predict noise levels through direct comparison with all other embedding vectors. Such an approach would result in a reverse process that merely retraces the forward process, providing little value for generation quality control. To address this limitation, we propose using the generated sample $\boldsymbol{z}_\theta$ as input to τ_θ. This design choice increases the modeling complexity of the prediction task while enabling τ_θ to serve dual purposes: noise level prediction and semantic quality assessment of the generated output. To provide temporal context, we incorporate $t' = t - \Delta t$ as an additional input, ensuring the model's awareness of the target time distribution. The complete formulation of $p_\theta(\tau_{t'}|\boldsymbol{z}_t, \tau_t)$ is expressed as $\tau_\theta(\boldsymbol{z}_\theta(\boldsymbol{z}_t, \tau_t, t), t', \boldsymbol{x})$, where $\boldsymbol{x}$ represents the conditioning sentence embedding.

Pseudo Label for Training the Time Predictor. The naive approach of using $\tau_{t'}$ as the direct training label for the time predictor can introduce systematic bias in the learning process. While $\tau_{t'}$ is derived from $\boldsymbol{z}_\theta$, this predicted quality measure may not accurately reflect the actual generation quality after the complete denoising process. For example, tokens initially assigned high noise levels might still produce high-quality outputs after denoising, making their initial $\tau_{t'}$ assignment suboptimal. Instead, we propose a pseudo-labeling strategy for training the time predictor. More specifically, we first compute a confidence score for each generated output using the combined loss $\mathcal{L}_z + \mathcal{L}_{\text{anchor}}$ from the denoised prediction $\boldsymbol{z}_\theta$. To ensure these confidence scores follow a distribution compatible with $\tau_{t'}$, we apply inverse transform sampling. To accomplish this, we compute the normalized rank r for each token's loss within the single sample and map these ranks through the inverse cumulative distribution function (ICDF) of the Poisson distribution: $\tilde{s}(t) = F^{-1}(r; \lambda(t))$, where F denotes the Poisson cumulative distribution function. The resulting $\tilde{s}(t)$ values are then transformed via Eq. (7) to obtain the final pseudo labels.

3.4 Optimized Extrinsic Time Schedule

The choice of time schedule in diffusion models significantly impacts both generation quality and computational efficiency. While previous works such as Dhariwal and Nichol (2021); Chen (2023) focus on optimizing the noise schedule function with fixed extrinsic time steps, we propose to perform direct optimization on the schedule of extrinsic time t. Our method builds upon Li et al. (2024), who introduced post-training Bayesian optimization to select optimal subsets of time steps for inference acceleration. However, where they treat time steps as discrete variables and optimize for subset selection, we formulate the problem as continuous optimization over the complete time schedule $\{t_1, t_2, \ldots, t_K\}$, where K denotes the total number of diffusion steps. This continuous formulation enables more precise calibration through Bayesian optimization, effectively exploring the full space of possible time schedules. We evaluate candidate schedules using a trained model on the validation set via Bayesian optimization, optimizing for the BLEU score as our objective metric. This approach yields task-specific optimal time schedules that further enhances generation quality. The detailed optimization procedure is presented in Appendix B.4.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics We evaluate our approach on several NLP tasks, including machine translation (WMT14 En-De (Bojar et al., 2014), WMT16 En-Ro (Bojar et al., 2016), IWSLT14 De-En (Cettolo et al., 2014)), paraphrasing (QQP), text simplification (Wiki-Auto (Jiang et al., 2020)), and question generation (Quasar-T (Dhingra et al., 2017)). Dataset splits are detailed in Table 16. We use BLEU score (Papineni et al., 2002) as the evaluation metric across all tasks, supplemented with SacreBLEU (Post, 2018) for translation tasks. For comprehensive evaluation, we employ LLM-based evaluation using DeepSeek-V3 685B (DeepSeek-AI, 2024) with specialized prompts, assessing accuracy, fluency, completeness, and task-specific criteria such as creativity for translation and phrasing diversity for paraphrasing. The evaluation pro-

T	Model	b	IWSLT	WMT14	WMT16
D	Absorbing	5	28.32*	21.62*	30.41*
	Multinomial	5	21.28*	6.94*	25.25*
C	AR-Diffusion	1	26.78	-	-
	AR-Diffusion	10	30.64	-	-
	SeqDiffuSeq	1	28.65†	23.63†	23.98†
	SeqDiffuSeq	10	30.03†	24.24†	26.17†
	Differmer	1	30.94	22.32	30.74
	Differmer	10	32.09	23.80	30.93
H	**NeoDiff**	1	32.39$^{\Uparrow}$	24.41$^{\Uparrow}$	30.87$^{\Uparrow}$
	NeoDiff	10	**33.14**$^{\Uparrow}$	**25.28**$^{\Uparrow}$	**32.31**$^{\Uparrow}$

Table 1: Machine translation BLEU scores for NeoDiff and baseline methods. **T**: Model type (AR: Autoregressive, D: Discrete, C: Continuous, H: Hybrid). $\Uparrow$: NeoDiff outperforms baselines with beam size $\leq b$; **bold**: best result. *: Results from Zheng et al. (2023); †: Results from Yuan et al. (2024); remaining data reproduced.

T	Model	b	IWSLT	WMT14	WMT16
D	CMLM	5	29.41*	23.22*	31.26*
	CMLM(MBR)	5	29.32*	23.09*	30.92*
C	DiffusionLM	5	26.61*	15.33*	27.01*
	DiffusionLM	50	29.11*	17.41*	29.39*
	SeqDiffuSeq	1	30.16†	19.16†	-
	SeqDiffuSeq	10	30.45†	19.76†	-
	DiNoiSer	5	31.29*	24.25*	30.93*
	DiNoiSer	50	31.61*	24.26*	31.08*
	Differmer	1	30.06	22.13	30.52
	Differmer	10	31.08	23.26	30.75
H	**NeoDiff**	1	31.50$^{\Uparrow}$	24.09	31.59$^{\Uparrow}$
	NeoDiff	10	**32.20**$^{\Uparrow}$	**24.64**$^{\Uparrow}$	**32.21**$^{\Uparrow}$

Table 2: Comparison on **SacreBLEU** for machine translation tasks. *: Results from Ye et al. (2023); †: Results from Yuan et al. (2024); remaining data are reproduced. $\Uparrow$: NeoDiff outperforms baselines with beam size $\leq b$.

T	Model	b	QQP	QT	WA
AR	Transformer	1	29.65*	16.83*	**41.68***
	Transformer	5	30.83*	16.45*	**43.86***
	GPT2-base FT	-	19.80$^{\diamond}$	7.41$^{\diamond}$	-
	GPT2-large FT	-	20.59$^{\diamond}$	11.10$^{\diamond}$	-
D	CMLM	1	24.02	-	-
	CMLM	10	26.32	-	-
	Absorbing	10	23.82*	17.38*	-
	Multinomial	10	20.70*	16.96*	-
C	SeqDiffuSeq	1	23.28†	17.20†	37.09†
	SeqDiffuSeq	10	24.34†	17.46†	37.12†
	Differmer	1	28.52	16.03	40.37
	Differmer	10	30.43	16.66	40.77
	Meta-DiffuB$^{D_{\theta_1}}$	-	25.52¶	18.20¶	38.77¶
	Meta-DiffuB$^{D_{\theta_2}}$	-	26.32¶	-	39.57¶
	Meta-DiffuB$^{D_{\theta_3}}$	-	22.71¶	-	24.71¶
	TESS	-	30.20‡	19.50‡	-
	TEncDM(BERT)	-	30.20°	-	41.60°
	TEncDM(T5)	-	30.20°	-	41.60°
	TEncDM(RoBERTa)	-	30.00°	-	40.50°
H	DiffuSeq-V2	1	22.10§	-	-
H	**NeoDiff**	1	29.47$^{\Uparrow}$	**20.44**$^{\Uparrow}$	41.57
	NeoDiff	10	**31.32**$^{\Uparrow}$	20.03$^{\Uparrow}$	41.86

Table 3: BLEU scores on QQP, QT, and WA(WikiAuto). *: Results from Zheng et al. (2023); †: Results from Yuan et al. (2024); §: Results from Gong et al. (2023); $\star$: Results from Gao et al. (2024); $\diamond$: Results from Gong et al. (2022); ¶: Results from Chuang et al. (2024). D_{θ_1} = DiffuSeq. D_{θ_2} = SeqDiffuSeq. D_{θ_3} = Dinoiser; ‡: Results from Karimi Mahabadi et al. (2024); $\circ$: Results from Shabalin et al. (2025). Remaining results reproduced. $\Uparrow$: NeoDiff outperforms baselines with beam size $\leq b$.

cess involved providing the LLM with source text, generated text from different models, and specific instructions tailored to each task. Figure 3 shows the prompt templates used. To rigorously assess the diversity of outputs of the model, We also included Inter-Sentence Div-4 as Gong et al. (2022), which measure diversity at the set-of-outputs-per-source level.

Baselines We compared NeoDiff against several strong baselines across multiple diffusion model categories. For discrete diffusion models, we included Absorbing Diffusion (Austin et al., 2021), Multinomial Diffusion (Hoogeboom et al., 2021b), and CMLM (Ghazvininejad et al., 2019). For continuous diffusion models, we benchmarked against DiffusionLM (Li et al., 2022), Differmer (Gao et al., 2024), SeqDiffuSeq (Yuan et al., 2024), AR-Diffusion (Wu et al., 2024), DiNoiSer (Ye et al., 2024), Meta-DiffuB (Chuang et al., 2024), TESS (Karimi Mahabadi et al., 2024) and TEncDM (Shabalin et al., 2025). For hybrid approaches, we compared with DiffuSeq-V2 (Gong et al., 2023). We also included Transformer and fine-tuned GPT2 models as autoregressive baselines.

Implementation Details We set the maximum noise state $s_{\max}$ to 100 for all tasks and datasets, incorporating self-conditioning (Chen et al., 2022) and noise rescaling with $\mathrm{DGS}_{\mathrm{MAX}} = 0.2$ (Gao et al., 2024). We used byte pair encoding (Sennrich et al., 2016) without knowledge distillation (Kim and Rush, 2016) to evaluate under challenging conditions. During decoding, we employed 2D parallel decoding (Gao et al., 2024) and selected the best candidate sentence using the minimum Bayes risk (MBR) method (Kumar and Byrne, 2004) based

Task	Models	b	Semantic Faithfulness	Fluency	Completeness	Phrasing Diversity
QQP	CMLM	10	72.86	81.99	75.60	55.86
	Transformer	1	83.64	92.56	84.96	**57.19**
	Transformer	5	83.70	**94.73**	86.05	54.52
	Transformer	10	83.93	94.64	86.02	54.55
	NeoDiff	10	**87.42**	91.87	**88.79**	45.83
	Models	**b**	**Accuracy**	**Fluency**	**Completeness**	**Creativity**
WMT14	Differmer	10	79.72	80.31	85.24	75.12
	Transformer	5	**85.66**	**86.35**	**90.81**	**80.07**
	NeoDiff	10	80.30	80.81	85.61	76.20

Table 4: LLM evaluation of text generation tasks using DeepSeek-v3 685B. We evaluate Paraphrasing (QQP Dataset) and Machine Translation (WMT14 En-De Dataset). (1) We access QQP on Semantic Faithfulness, Fluency, Completeness, and Phrasing Diversity. (2) We access WMT14 En-De on Accuracy, Fluency, Completeness, and Creativity. Detailed prompts are provided in Fig 3.

on the BLEU score. We also used post-training Bayesian optimization (Li et al., 2024) to calibrate the extrinsic time schedule, limiting the optimization to 100 rounds for all tasks. Details of the experimental settings are provided in Appendix B.

4.2 Results

Our experimental evaluation demonstrates NeoDiff's effectiveness across multiple generation tasks. On machine translation benchmarks (Table 1 and 2), NeoDiff consistently outperforms existing non-autoregressive diffusion-based, iteration-based, and autoregressive diffusion approaches. As shown in Table 3, these improvements extend beyond translation to diverse generation tasks. Unlike baselines such as AR-Diffusion that rely heavily on MBR and show performance drops with single samples ($b = 1$), NeoDiff maintains robust performance even in this constrained setting. NeoDiff also demonstrates strong performance in LLM-based evaluations (Table 4, prompts in Figure 3). Notably, on the QQP task (Table 4), NeoDiff achieves superior scores in semantic faithfulness and completeness. For the WMT14 task, NeoDiff achieves performance comparable to Differmer across multiple aspects. NeoDiff also demonstrates strong inter-sentence diversity (Inter-Sentence Div-4) when generating multiple candidates. Detailed comparisons against AR model on QQP dataset can be found in Appendix C (Table 8). Our results show that NeoDiff can balance the quality-diversity trade-off more effectively than autoregressive models like Transformer as the output space scales (i.e., with increasing b), a characteristic also observed in Gong et al. (2022). The Bayesian Optimization component introduces a manageable overhead (Appendix D.1, approximately 6% of training time on WMT14). Also, NeoDiff's inference speed and memory usage are competitive with similar models (Appendix D.2).

#	Poisson Diffusion Process	Time Predictor	Optimized t Schedule	BLEU
Base				32.09
+P	✓			32.75
+PT	✓	✓		32.97
Full	✓	✓	✓	**33.14**

Table 5: Ablation study on the impact of proposed components on IWSLT14 De-En dataset with a $b = 10$.

4.3 Analysis

Our ablation studies (Table 5) demonstrate clear improvements from each component, with the full model achieving a substantial +1.05 BLEU improvement over the baseline. We further analyze each component's impact on generation quality:

Poisson Process for Multi-token Coherence The Poisson diffusion process enables more fine-grained control over multiple tokens by precise inter-token coordination. This advantage yields a substantial performance gain over standard continuous diffusion ($\tau_t = t$). As evidenced in Table 6A, this improved control manifests itself in better phrase-level coherence.

Time Predictor for Guided Denoising By leveraging information from less-noised tokens to guide the denoising trajectory of noisier ones, the time predictor enhances the model's ability of more contextually informed token generation. Table 6B demonstrates this through more natural word selections and verb choices that better preserve the original meaning.

A	**Src:** das zeigt die enorm große rolle , die ein meeresschutzgebiet spielen kann. **Ref:** and hence , the enormous role that a marine protected area can play. **Base:** and this shows the enormously big role that a **area can play with a sea protected** **+P:** so this shows the enormously big role that a **marine protected area can play**
B	**Src:** er ist ganz glücklich darüber, weil er sie getäuscht hat. **Ref:** he'll be very happy because he's deceived you. **+P:** he's very happy about it because he **decaked** her. **+PT:** he's very happy about it because he **deceived** her.
C	**Src:** die korrelation ist also gering . **Ref:** so the correlation is low . **+PT:** so **it's a small of the correlation**. **Full:** so **the correlation is small**.

Table 6: Example outputs illustrating three key mechanisms of NeoDiff: (A) improved phrase-level coherence with Poisson process, (B) enhanced token-level refinements with time predictor, and (C) better sentence-level organization with optimized schedule.

Optimized Schedule for Global Coherence The optimized extrinsic time schedule enables dynamic adjustments to the diffusion trajectory, facilitating escape from sub-optimal samples where sequence order or overall structure significantly deviates from the target distribution. This global refinement allows for more substantial rewriting when needed, as demonstrated in Table 6C where entire phrases are better reorganized.

Additional examples demonstrating the impact of these components are provided in Table 11, 12, and 13. In Appendix E, we track the step-wise generation processes, demonstrating superior convergence speed and accuracy for NeoDiff compared to continuous diffusion baselines. We also compared NeoDiff against continuous diffusion baselines on token-level controlled generation, demonstrating its unique ability to perform targeted modifications while maintaining semantic consistency across translations (Appendix F).

5 Conclusion

In this work, we introduce Non-simultaeous Continuous Diffusion Models (NeoDiff), a novel diffusion-based text generation framework that unifies discrete and continuous diffusion models. NeoDiff generalizes the time variable, incorporates the Poisson diffusion process, adaptively modulates the reverse process based on semantic context, and uses an optimized extrinsic time schedule for inference. This unified framework enables fine-grained control and achieves superior performance across diverse natural language processing tasks. Our extensive experiments demonstrate the effectiveness of this unified framework, opening up new avenues for advancing diffusion-based text generation.

Limitations

While NeoDiff demonstrates strong performance across various Seq2Seq-based conditional generation tasks (e.g., machine translation, paraphrasing, text simplification, and question generation), we note some implementation considerations. The post-training optimization of extrinsic time schedules requires additional sampling iterations, though this overhead is negligible compared to the training time. The time predictor introduces a modest parameter increase to the backbone network.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grant No. 62276245).

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.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation (wmt16). In *First conference on machine translation*, pages 131–198. Association for Computational Linguistics.

Eric Brochu, Matthew W. Hoffman, and Nando de Freitas. 2011. Portfolio allocation for bayesian optimization. *Preprint*, arXiv:1009.5419.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 2–17.

Jiaao Chen, Aston Zhang, Mu Li, Alex Smola, and Diyi Yang. 2023. A cheaper and better diffusion language model with soft-masked noise. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4765–4775, Singapore. Association for Computational Linguistics.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. 2020. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*.

Ting Chen. 2023. On the importance of noise scheduling for diffusion models. *Preprint*, arXiv:2301.10972.

Ting Chen, Ruixiang ZHANG, and Geoffrey Hinton. 2022. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*.

Yunyen Chuang, Hung-Min Hsu, Kevin Lin, Chen-Sheng Gu, Ling Zhen Li, Ray-I Chang, and Hung yi Lee. 2024. Meta-diffu$b$: A contextualized sequence-to-sequence text diffusion model with meta-exploration. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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

Bhuwan Dhingra, Kathryn Mazaitis, and William W. Cohen. 2017. Quasar: Datasets for question answering by search and reading. *Preprint*, arXiv:1707.03904.

Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. 2024. Empowering diffusion models on the embedding space for text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4664–4683, Mexico City, Mexico. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. Diffuseq-v2: Bridging discrete and continuous text spaces for accelerated seq2seq diffusion models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9868–9875.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. 2023. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11575–11596.

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

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

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

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

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural crf model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960.

Rabeeh Karimi Mahabadi, Hamish Ivison, Jaesung Tae, James Henderson, Iz Beltagy, Matthew Peters, and Arman Cohan. 2024. TESS: Text-to-text self-conditioned simplex diffusion. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2347–2361, St. Julian's, Malta. Association for Computational Linguistics.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2020. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter*

of the Association for Computational Linguistics: HLT-NAACL 2004, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Bocheng Li, Zhujin Gao, Yongxin Zhu, Kun Yin, Haoyu Cao, Deqiang Jiang, and Linli Xu. 2024. Few-shot temporal pruning accelerates diffusion models for text generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7259–7269, Torino, Italia. ELRA and ICCL.

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.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528.

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

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR.

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. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Alexander Shabalin, Viacheslav Meshchaninov, Egor Chimbulatov, Vladislav Lapikov, Roman Kim, Grigory Bartosh, Dmitry Molchanov, Sergey Markov, and Dmitry Vetrov. 2025. Tencdm: Understanding the properties of the diffusion model in the space of language model encodings. *Preprint*, arXiv:2402.19097.

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.

Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. 2024. Ar-diffusion: Auto-regressive diffusion model for text generation. *Advances in Neural Information Processing Systems*, 36.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Mingxuan Wang. 2023. Dinoiser: Diffused conditional sequence learning by manipulating noises. *Preprint*, arXiv:2302.10025.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Mingxuan Wang. 2024. Dinoiser: Diffused conditional sequence learning by manipulating noises. *Preprint*, arXiv:2302.10025.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2024. Text diffusion model with encoder-decoder transformers for sequence-to-sequence generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 22–39, Mexico City, Mexico. Association for Computational Linguistics.

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

A Detailed Derivation of the Training Objective of NeoDiff

Let $\boldsymbol{z}$ represent a token embedding and $\boldsymbol{z}_t$ its latent representation at time t, with $\boldsymbol{z}_0 = \boldsymbol{z}$ and $\boldsymbol{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The joint distribution of the forward process is then given by:

$$q(\boldsymbol{z}_{>0}, \tau_{>0}|\boldsymbol{z}_0) := \prod_{t>0} q(\boldsymbol{z}_t, \tau_t|\boldsymbol{z}_0) \quad (8)$$

$$= \prod_{t>0} q(\boldsymbol{z}_t|\boldsymbol{z}_0, \tau_t) q(\tau_t), \quad (9)$$

where

$$q(\boldsymbol{z}_t|\boldsymbol{z}_0, \tau_t) := \mathcal{N}\left(\boldsymbol{z}_t; \sqrt{\bar{\alpha}(\tau_t)}\boldsymbol{z}_0, \bar{\beta}(\tau_t)\mathbf{I}\right),$$

and $\bar{\alpha}(\cdot)$ and $\bar{\beta}(\cdot)$ denote noise schedules with their domains scaled to $[0, 1]$.

Given $t' = t - \Delta t$, the reverse process is defined as

$$p_\theta(\boldsymbol{z}_{0:1}, \tau_{0:1}) := p_\theta(\boldsymbol{z}_1, \tau_1) \prod_{t'<1} p_\theta(\boldsymbol{z}_{t'}, \tau_{t'} | \boldsymbol{z}_t, \tau_t) \tag{10}$$

$$= p_\theta(\boldsymbol{z}_1, \tau_1) \prod_{t'<1} p_\theta(\boldsymbol{z}_{t'} | \boldsymbol{z}_t, \tau_t, \tau_{t'}) p_\theta(\tau_{t'} | \boldsymbol{z}_t, \tau_t). \tag{11}$$

We further parameterize the distribution of $\boldsymbol{z}_{t'}$ as

$$p_\theta(\boldsymbol{z}_{t'} | \boldsymbol{z}_t, \tau_t, \tau_{t'}) = q(\boldsymbol{z}_{t'} | \hat{\boldsymbol{z}}_0(\boldsymbol{z}_t, \tau_t, t), \tau_{t'}),$$

where $\hat{\boldsymbol{z}}_0$ is the model prediction of $\boldsymbol{z}_0$.

Following Ho et al. (2020), the training objective is derived from the variational lower-bound

$$\mathcal{L}_{\text{VLB}} = \mathbb{E}_q \left[-\log \frac{p_\theta(\boldsymbol{z}_{0:1}, \tau_{0:1})}{q(\boldsymbol{z}_{>0}, \tau_{>0} | \boldsymbol{z}_0)} \right] \tag{12}$$

$$= \mathbb{E}_q \Bigg[-\log \frac{p_\theta(\boldsymbol{z}_1, \tau_1)}{q(\boldsymbol{z}_1, \tau_1 | \boldsymbol{z}_0)} \tag{13}$$

$$+ \sum_{0<t'<1} -\log \frac{q(\boldsymbol{z}_{t'} | \hat{\boldsymbol{z}}_0(\boldsymbol{z}_t, \tau_t, t), \tau_{t'})}{q(\boldsymbol{z}_{t'} | \boldsymbol{z}_0, \tau_{t'})} \tag{14}$$

$$+ \sum_{0<t'<1} -\log \frac{p_\theta(\tau_{t'} | \boldsymbol{z}_t, \tau_t)}{q(\tau_{t'})} \tag{15}$$

$$- \log p_\theta(\boldsymbol{z}_0, \tau_0 | \boldsymbol{z}_{\Delta t}, \tau_{\Delta t}) \Bigg] \tag{16}$$

$$= \mathbb{E}_q \Bigg[\underbrace{\mathbb{KL}(q(\boldsymbol{z}_1, \tau_1 | \boldsymbol{z}_0) \| p_\theta(\boldsymbol{z}_1, \tau_1))}_{\mathcal{L}_1} \tag{17}$$

$$+ \sum_{0<t'<1} \underbrace{\mathbb{KL}(q(\boldsymbol{z}_{t'} | \boldsymbol{z}_0, \tau_{t'}) \| q(\boldsymbol{z}_{t'} | \hat{\boldsymbol{z}}_0, \tau_{t'}))}_{\mathcal{L}_z} \tag{18}$$

$$+ \sum_{0<t'<1} \underbrace{\mathbb{KL}(q(\tau_{t'}) \| p_\theta(\tau_{t'} | \boldsymbol{z}_t, \tau_t))}_{\mathcal{L}_\tau} \tag{19}$$

$$\underbrace{- \log p_\theta(\boldsymbol{z}_0, \tau_0 | \boldsymbol{z}_{\Delta t}, \tau_{\Delta t})}_{\mathcal{L}_0} \Bigg]. \tag{20}$$

Note that $\mathcal{L}_1$ is a constant and can be ignored, and $\mathcal{L}_0$ also becomes negligible when $\Delta t \to 0$. According to prior works (Ho et al., 2020; Li et al., 2022), the term $\mathcal{L}_z$ can be simplified as

$$\mathcal{L}_z = \|\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t, \tau_t, t) - \boldsymbol{z}_0\|^2.$$

We also add an anchor loss (Gao et al., 2024) $\mathcal{L}_{\text{anchor}} = \mathbb{E}_q[-\log p_\theta(y | \hat{\boldsymbol{z}}_0(\boldsymbol{z}_t, \tau_t, t))]$ as a regularization term to avoid collapse of the embedding space. Finally, the training objective of the proposed NeoDiff can be written as

$$\mathcal{L} = \mathcal{L}_z + \mathcal{L}_\tau + \mathcal{L}_{\text{anchor}}.$$

B Experimental Settings

B.1 Data Preprocessing

We used byte pair encoding (BPE) (Sennrich et al., 2016) for tokenization. Unlike previous work, we did not employ knowledge distillation (Kim and Rush, 2016) for preprocessing to evaluate our model's performance under more challenging conditions.

B.2 Model Configuration

For our experiments, we set the maximum noise state $s_{\max}$ to 100 and used the *sqrt* schedule for training, optimized schedule for inference. To enhance model performance, we applied self-conditioning (Chen et al., 2022). The transition schedule coefficient k was set to 2, and the maximum truncation time $t_{\max}$ was set to 0.99. Following Gao et al. (2024), we also employed noise rescaling with a degradation score threshold $\mathrm{DGS}_{\mathrm{MAX}}$ of 0.2.

Regarding the model architecture, we adopted the configuration from Gao et al. (2024) for the IWSLT14 De-En, WMT14 En-De, and WMT16 En-Ro datasets. For the QQP, Wiki-Auto, and QT datasets, we used the configuration from Gong et al. (2022) to enable a fair comparison with these models. Detailed settings are presented in Table 9.

B.3 Training and Generation

We trained our models using NVIDIA RTX 3090 24G GPUs on Ubuntu 18.04 with FairSeq 0.12(Ott et al., 2019)(MIT-licensed). For the WMT14 En-De and WMT16 En-Ro datasets, training took nearly 4 days and 2 days, respectively, using 4 GPUs. For the IWSLT14 De-En dataset, training took approximately 1 day using a single GPU. The QQP, Wiki-Auto, and QT datasets each required around 8 hours of training on a single GPU. The training data splits are presented in Table 16.

During generation, we used 20 iteration steps ($K = 20$) without early stopping for the IWSLT14 De-En dataset. For the other datasets, we employed 10 iteration steps without early stopping, which is faster than the 20 steps ($k = 20$) used by Gao et al. (2024) across all datasets. We utilized 2D parallel decoding and selected the best sentence using the minimum Bayes risk (MBR) (Kumar and Byrne, 2004) method based on the BLEU score. The reported results are averaged over 3 runs. The random seed is set to 7.

B.4 Optimized Extrinsic Time Schedule

We propose a systematic approach to optimize the extrinsic time schedule $\mathbf{S} = \{t_1, t_2, ..., t_K\}$, where K denotes the number of diffusion steps and $t_i \in [0, 1]$ with $t_1 < t_2 < ... < t_K$. While previous works (Dhariwal and Nichol, 2021; Chen, 2023) focus on optimizing noise schedules with fixed time steps, we directly optimize the time schedule through Bayesian optimization. Our method extends Li et al. (2024)'s framework from discrete subset selection to continuous optimization over the complete schedule.

At its core, our approach is straightforward: we sample text using different time schedules on the validation set and select the schedule that achieves the highest BLEU score for inference. The optimization process (Algorithm 1) employs Gaussian Process-based Bayesian optimization with the GP-Hedge acquisition function (Brochu et al., 2011). Starting from a uniform time schedule, we iteratively propose candidate schedules using Limited-memory BFGS (Liu and Nocedal, 1989) and evaluate them using BLEU scores on the validation set. This approach enables precise calibration of the time schedule while maintaining the ordering constraint $t_1 < t_2 < ... < t_K$. Following Li et al. (2024), we limit optimization to 100 iterations, keeping the computational overhead negligible compared to model training time(Li et al., 2024). The resulting task-specific schedules demonstrate improved generation quality while maintaining computational efficiency.

C Additional Diversity Analysis on QQP dataset

In this section, we provide a detailed comparison of NeoDiff and Transformer on the QQP task, specifically focusing on multi-candidate generation and inter-sentence diversity. The results presented in Table 8 complement the main paper's Table 4 by offering a deeper look into how diversity metrics evolve with an increasing number of generated samples (b).

D Efficiency Analysis

D.1 Bayesian Optimization Overhead (WMT14 En-De):

- Training: 505.88 RTX3090 GPU Hours
- Bayesian Optimization: 28.1 RTX3090 GPU Hours (approximately 6% of training time)

Models	K	Speed (sentences/second)	Memory Cost (MB)
Transformer*	n	6.05	-
CMLM*	10	11.80	-
DiffuSeq*	2000	0.06	-
SeqDiffuSeq*	2000	0.05	-
Differmer	20	6.49	2034
NeoDiff	20	5.12	2080

Table 7: Runtime Comparison on IWSLT14 De-En. *: Results from Gao et al. (2024). Others are reproduced.

Model	**b**	**Semantic Faithfulness**	**Fluency**	**Completeness**	**Phrasing Diversity**	**Inter-Sentence div-4**
Transformer	1	83.64	92.56	84.96	**57.19**	**1.000**
Transformer	5	83.70	**94.73**	86.05	54.52	0.686
Transformer	10	83.93	94.64	86.02	54.55	0.561
NeoDiff	1	84.24	88.95	87.83	39.18	**1.000**
NeoDiff	5	85.63	90.69	88.39	41.62	0.684
NeoDiff	10	**87.42**	91.87	**88.79**	45.83	0.631

Table 8: Detailed comparison of NeoDiff and Transformer on the QQP task. Metrics include Semantic Faithfulness, Fluency, Completeness, Phrasing Diversity (single-sample), and Inter-Sentence Diversity (Inter-Sentence Div-4, multi-candidate).

Note: The cost of Bayesian optimization is directly proportional to the amount of data sampled in each iteration. While we used the entire WMT14 validation set, significantly reducing the sample size (e.g., to 20 samples) can drastically lower this overhead to less than 0.1 GPU Hours (Li et al., 2024).

D.2 Runtime Comparison (IWSLT14 De-En)

Table 7 presents a runtime comparison of NeoDiff and several baselines on the IWSLT14 De-En dataset. We measured inference speed (sentences/second) and memory cost (MB). NeoDiff demonstrates competitive inference speed, processing 5.12 sentences per second, which is comparable to Differmer's 6.49 sentences per second. While significantly faster than diffusion-based models like DiffuSeq and SeqDiffuSeq, NeoDiff's speed is lower than the highly optimized Transformer and CMLM models. In terms of memory usage, NeoDiff's 2080 MB consumption is similar to Differmer's 2034 MB.

E Step-wise Generation Examples on IWSLT14 De-En for NeoDiff and Differmer

Table 14 and 15 present a detailed comparison of the translation generation process on IWSLT14 De-En dataset between NeoDiff and Differmer(continuous diffusion model). After incorporating the three aforementioned components(Poisson process, time predictor and optimized schedule), NeoDiff demonstrates more accurate and faster convergence in translation on some sentences compared to Continuous Diffusion Model(Differmer), as illustrated by the step-by-step generation process. Specifically, NeoDiff avoids some of the common pitfalls of diffusion models, such as getting stuck in local optima or generating repetitive phrases.

F Fine-grained Controlled Generation through Token Manipulation

We demonstrate NeoDiff's capability for token-level controlled generation while preserving semantic consistency across translations. Given a source sentence $\mathbf{x}_{\text{src}}$ and its latent representation $\mathbf{z}_0$, we replace a single token to obtain a modified source $\mathbf{x}'_{\text{src}}$. For translation, we initialize the process with $\mathbf{z}_0$ and set $\tau = 1$ only for the modified token position, maximizing noise specifically at that location. This targeted noise application enables precise semantic modifications in the output translation $\mathbf{x}'_{\text{tgt}}$ while preserving the remaining content. As shown in Table 10, NeoDiff achieves localized modifications, whereas baseline methods like Differmer tend to alter substantial portions of the output sentence. This controlled generation capability stems from our fine-grained noise paradigm, enabling token-specific manipulation of the generation process.

Algorithm 1 Extrinsic Time Schedule Calibration via Bayesian Optimization

Require: Trained Diffusion Model M,
Initial Extrinsic Time Schedule $\mathbf{S}_{\text{init}} = \{t_1, t_2, ..., t_K\}$, where $t_i \in \mathbb{R}$ and $0 \leq t_1 < t_2 < ... < t_K \leq 1$,
Optimization iterations n_{iter},
Domain for elements in Extrinsic Time Schedule $\mathcal{D} \subset [0, 1]$,
Source Text T_{src},
Target Text T_{tgt}

Ensure: Optimized Extrinsic Time Schedule $\mathbf{S}_{\text{opt}} = \{t'_1, t'_2, ..., t'_K\}$, where $t'_i \in \mathbb{R}$ and $0 \leq t'_1 < t'_2 < ... < t'_K \leq 1$

Initialize $\mathbf{S}_{\text{init}} = \{t_1, t_2, ..., t_K\}$ such that t_i are uniformly spaced in $[0, 1]$.
Perform a sampling on T_{src} using diffusion model M and extrinsic time schedule $\mathbf{S}_{\text{init}}$, yielding predicted text T_{pred}.
Compute the BLEU score $\text{BLEU}(T_{\text{tgt}}, T_{\text{pred}})$ using T_{tgt} and T_{pred}.
Initialize the observation set for Bayesian optimization: $O \leftarrow \{(\mathbf{S}_{\text{init}}, \text{BLEU}(T_{\text{tgt}}, T_{\text{pred}}))\}$.
for $i = 1$ to n_{iter} **do**
    Update the Gaussian Process posterior given observations O.
    Generate a candidate set $\mathcal{D}' = \{\mathbf{S}'_1, \mathbf{S}'_2, ..., \mathbf{S}'_N\}$, where each $\mathbf{S}'_j = \{t'_{j1}, t'_{j2}, ..., t'_{jK}\}$ represents a candidate extrinsic time schedule with $t'_{jk} \in \mathcal{D}$ and $0 \leq t'_{j1} < t'_{j2} < ... < t'_{jK} \leq 1$. The candidate set $\mathcal{D}'$ is generated by performing 20 iterations of Limited-memory BFGS (Liu and Nocedal, 1989) with 5 random initial points within $\mathcal{D}^K$.
    Compute the acquisition function value $\alpha_{\text{GP-Hedge}}(\mathbf{S}'_j)$ (Brochu et al., 2011) for all $\mathbf{S}'_j \in \mathcal{D}'$.
    Select the next observation point $\mathbf{S}_i = \arg\max_{\mathbf{S}'_j \in \mathcal{D}'} \alpha_{\text{GP-Hedge}}(\mathbf{S}'_j)$.
    Perform a sampling on T_{src} using M and $\mathbf{S}_i$, yielding predicted text T'_{pred}.
    Compute the BLEU score $\text{BLEU}(T_{\text{tgt}}, T'_{\text{pred}})$ using T_{tgt} and T'_{pred}.
    Update the observation set: $O \leftarrow O \cup \{(\mathbf{S}_i, \text{BLEU}(T_{\text{tgt}}, T'_{\text{pred}}))\}$.
end for
$\mathbf{S}_{\text{opt}} = \arg\max_{(\mathbf{S}, \text{BLEU}) \in O} \text{BLEU}$

Hyper-parameters	WMT14 En-De	WMT16 En-Ro	IWSLT14 De-En	QQP	Wiki-Auto	QT
Architecture						
d_{model}	512	512	512	768	768	768
d_{emb}	128	128	128	128	128	128
d_{ffn}	2048	2048	1024	3072	3072	3072
Heads	8	8	4	12	12	12
Encoder Layers	6	6	6	6	6	6
Decoder Layers	6	6	6	6	6	6
Time Predictor Layers	3	1	1	1	1	1
Activation	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU
Diffusion						
Steps	10	10	20	10	10	10
Training Schedule	*sqrt*	*sqrt*	*sqrt*	*sqrt*	*sqrt*	*sqrt*
Inference Schedule	Optimized	Optimized	Optimized	Optimized	Optimized	Optimized
DGS_{MAX}	0.2	0.2	0.2	0.2	0.2	0.2
Self-Conditioning	✓	✓	✓	✓	✓	✓
Training						
Steps	600K	400K	300K	50K	100K	100K
Batch Size (Tokens)	32K	24K	8K	8K	12K	16K
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Adam β	$(0.9, 0.98)$	$(0.9, 0.98)$	$(0.9, 0.98)$	$(0.9, 0.98)$	$(0.9, 0.98)$	$(0.9, 0.98)$
Weight Decay	0.01	0.01	0.01	0.01	0.01	0.01
Learning Rate	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	2.3×10^{-4}	2×10^{-4}
Warmup	10K	10K	10K	10K	10K	10K
Clip Gradient	1.0	1.0	1.0	1.0	1.0	1.0
Dropout	0.1	0.1	0.3	0.1	0.1	0.1
Length Predict Factor	0.1	0.1	0.1	0.1	0.1	0.1
Label Smoothing	0.1	0.1	0.1	0.1	0.1	0.1
Inference						
Steps	10	10	20	10	10	10
Bayesian Optimization Rounds	100	100	100	100	100	100

Table 9: The model architectures and hyper-parameters used in our experiments.

```
Evaluate this translation from {src_lang} to {tgt_lang} (0-100 score):
[Source] {source}
[Reference] {reference}
[Translation] {translation}
Score these aspects STRICTLY IN THIS ORDER:
1. **Accuracy**: Faithfulness to source meaning
2. **Fluency**: Naturalness in target language
3. **Completeness**: Information retention
4. **Creativity**: Handling of ambiguous or open-ended source content

Return ONLY 4 numbers separated by commas, NO text.
```

```
Evaluate this paraphrase generation (0-100 score):
[Original] {source}
[Reference] {reference}
[Paraphrase] {paraphrase}
Score these aspects STRICTLY IN THIS ORDER:
1. **Semantic Faithfulness**: Meaning preservation from original
2. **Fluency**: Naturalness in language
3. **Completeness**: Retention of all information
4. **Phrasing Diversity**: Variation in wording/structure while preserving meaning

Return ONLY 4 numbers separated by commas, NO text.
```

Figure 3: Prompt templates used for LLM-based evaluation. Top: Translation evaluation prompt. Bottom: Paraphrase evaluation prompt.

<src>	und die welt in der wir jetzt leben sieht so aus .
<tgt>	and the world we now live in looks like this .
<src'>	und die welt in der wir jetzt leben sieht anders aus .
<tgt'>	and the world we now live in looks different .
Model	**Generated Content**
NeoDiff **<tgt_pred>**	and the world we live in now , looks like this .
NeoDiff **<tgt'_pred>**	and the world we live in now , looks different .
Differmer **<tgt_pred>**	and the world we're living in now looks like this .
Differmer **<tgt'_pred>**	and the world that we're living in right now , it looks different .
<src>	sein ganzer arbeitsprozess hat sich danach geändert .
<tgt>	and his whole work process changed after that .
<src'>	sein ganzer arbeitsprozess hat sich davon geändert.
<tgt'>	His whole work process changed because of that.
Model	**Generated Content**
NeoDiff **<tgt_pred>**	his whole work process has changed after that .
NeoDiff **<tgt'_pred>**	his whole work process has changed from that .
Differmer **<tgt_pred>**	and his whole work process has changed after that .
Differmer **<tgt'_pred>**	and his whole work process has changed from this .
<src>	der zweite faktor sind die dienste , die wir nutzen .
<tgt>	the second factor is the services we use .
<src'>	der zweite faktor sind die dienste , die wir kennen .
<tgt'>	The second factor is the services we know .
Model	**Generated Content**
NeoDiff **<tgt_pred>**	the second factor is the services that we use .
NeoDiff **<tgt'_pred>**	the second factor is the services we know .
Differmer **<tgt_pred>**	the second factor is the services that we use .
Differmer **<tgt'_pred>**	the second factor is really the services that we meet .

Table 10: Token manipulation example.

Source	Reference	Base Translation	+P Translation
nein war nie eine möglichkeit gewesen .	no had never been an option .	no one has never been a possibility.	no had never been an opportunity.
weiß jemand , was drei sekunden sind ?	does anyone know what three seconds are ?	does anybody know what three seconds?	does anyone know what three seconds are?
sie hatten ein konzept von blauem blut .	they had a concept of blue blood .	they had a idea of blue blood.	they had a concept of blue blood.
und raten sie was wir in dem angriffscode gefunden haben ?	and guess what we found in the attack code ?	and do you guess what we've found in the code of attack?	and guess what we found in the code of attack?
jetzt sehen sie den dalmatiner .	now you see the dalmatian .	now this is the dalmatinan.	now you see the dalmatiner.
denn die kategorien sagen mir , wie ich sie auseinander halten kann .	because the categories tell me how to tell them apart .	because the categories are telling me how i can keep it apart.	because the categories tell me how to keep them apart.
wie konnte es möglich sein , dass wir dies tun ?	how could it be possible that we would do this ?	so how could it possible for us to do this?	how could it be possible that we could do this?
aber es gab immer einen lebenszyklus in ihren präsentationen .	but there was always a life cycle to their presentations .	but there has always been a life cycle in your presentations.	but there was always a life cycle in their presentations.
wir reden zwiespältig davon .	we talk about it ambivalently .	we're talking about it in elessly.	we talk about it continally.
was geschah also jahre danach ?	so what happened years afterward ?	so for years after that, what happened?	so what happened years after that?

Table 11: Additional examples showing the improvements from introducing the Poisson diffusion process on IWSLT14 De-En dataset. The Base model often produces unnatural word ordering and incorrect lexical choices, while +P shows better handling of complex phrases and more natural English constructions.

Source	Reference	+P Translation	+PT Translation
sie haben ihr telefon gemietet . sie haben es nicht gekauft .	you rented your phone . you didn't buy it .	they've rtended your phone. they didn't buy it.	you rented your phone. you didn't buy it.
ihre familie versammelte sich .	and the family gathered .	her family.	her family gathered.
dunkler urin . dunkel .	dark urine . dark .	dark up. dark.	dark urine. dark.
diese leute verdienen geld .	these guys make money .	these people are earking money.	these people make money.
er ist ganz glücklich darüber , weil er sie getäuscht hat .	he'll be very happy because he's deceived you .	he's very happy about it because he decaked her.	he's very happy about it because he deceied her.
er hatte 20 minuten herrlicher musik gehabt .	he had had 20 minutes of glorious music .	he'd had 20 minutes of god.	he'd had 20 minutes of glorious music.
... es dem crowdsourcing beachtung schenkt .	... paying attention to crowdsourcing .	... it's adghting to the crowdsourcing.	... it gives attention to the crowdsourcing.
er zeigte immer hier hin .	he kept pointing here .	he always showed here.	he always pointed over here.
wenn man es verallgemeinert , passiert folgendes .	if you generalize this , something like this happens .	when you generate it, this is what happens.	when you generalize it, this is what happens.
man konnte manhattan sehen .	you could see manhattan .	see manhattan.	you could see manhattan.

Table 12: Additional examples demonstrating the impact of the time predictor module on IWSLT14 De-En dataset. The examples show how the time predictor enables finer-grained control primarily through word substitutions and better token-level refinements by leveraging information from less-noised tokens to guide the denoising process.

Source	Reference	+PT Translation	Full Translation
so wie es früher eben entsprechend auf dem dorf passierte .	just like it used to happen in the village .	in the same way that happened in the village, it just happened.	just as it used to happen in the village.
darum helfen sie da mit , fragen sie bei den leuten mal nach .	so you're helping out there , just ask the people .	so you can help with there, ask about people.	that's why they help there with, ask people to ask.
noch immer sind wir dem storytelling als informationsvermittlung sehr , sehr stark verhaftet .	what's left is storytelling .	but we're still very sted to storytelling as an information reation, very vivily arrested.	we're still very to storytelling as an information mediation, very, very arrested.
wir wählen jedes jahr einige fellows aus und wir lassen sie mit stadtverwaltungen arbeiten .	we select a few fellows every year and we have them work with city governments .	we choose some fellows every year, and we have them work with city adminicies.	we choose some fellows every year, and we let them work with urban management.
also bot ich einen 10000 $ preis an software für die gewinner .	so i offered a 10,000 dollar prize of software to the winning team .	so i offered a $10,000 price for the winner software.	so i offered a 100,000 price of software for the winners.
dies ist in unserem ganzen land der zweitgrösste abfallfluss amerikas .	this , all over the country , is the second largest waste stream in america .	this is in our entire country, the two-largest waste river of america.	this is the second est waste flow in america's land in our entire country.
wir haben eine art gleichgewicht erreicht .	we have reached a kind of equipoise .	we've reachved some kind of equilibrium.	we've reached some kind of balance.
und das ist aber ganz im anfang .	and that's just the beginning .	and that's just at the very beginning.	and that's at the very beginning.
ich bin überzeugt , dass man irgendwie zur nostalgie , zu wunschdenken hingezogen ist .	i'm convinced that there's some sort of pull to nostalgia , to wishful thinking .	i'm convinced you've been drawn to nostalgia, sort of wokkthinking.	i'm believe that there's kind of moved to nostalgia, you're moved to thinking.
wir haben uns daran gewöhnt , dass dinge linear passieren .	we no longer imagine the thing in images things in images , but codify them through language .	we were used to make things happen to linear.	so we've been used to linear that things happen.

Table 13: Additional examples showing the impact of the optimized schedule on IWSLT14 De-En dataset. These examples demonstrate how the schedule primarily influences the overall sampling trajectory at the sentence level, leading to more natural sentence constructions and better semantic coherence.

Time Step	Differmer Translation	NeoDiff Translation (Ours)
Source: ihr problem ist , dass sie zu wenig haben . **Reference:** their problem is that they have too little .		
0	dete@@ social tious falsche foot ere security madeupword0000 sorry fold says write chri@@	28 lar@@ electricity terms surface ting madeupword0001 madeupword0000 ® gen
1	your problem is that is that they have too little .	their problem is they they have too little .
2	the problem of that is that they have too little .	their problem is that they have too little .
3	the problem of that is that they have too little .	their problem is that they have too little .
4	the problem your you is that they have too little .	their problem is that they have too little .
5	the problem your problem is that they have too little .	their problem is that they have too little .
6	and , your problem is that they have too little .	their problem is that they have too little .
7	now , your problem is that they have too little .	their problem is that they have too little .
8	now , your problem is , they have too little .	their problem is that they have too little .
9	now , your problem is , you have too little .	their problem is that they have too little .
...	...	...
20	now , your problem is , you have too little .	their problem is that they have too little .
Final	now , your problem is , you have too little .	their problem is that they have too little .
Source: denn die kategorien sagen mir , wie ich sie auseinander halten kann . **Reference:** because the categories tell me how to tell them apart .		
0	es clay madeupword0002 ahead jobs involved line madeupword0001 fold <pad> <unk> giving bu@@ <unk> ers sa@@	market@@ madeupword0003 madeupword0001 van mas price gun ba madeupword0000 <unk> 3 ator anima@@ once
1	because the categ@@ ories tell tell me how can can hold them apart .	because the categ@@ ories tell me how to keep them apart .
2	because the categ@@ ories tell tell me how can can hold them apart .	because the categ@@ ories tell me how to keep them apart .
3	because the categ@@ ories are tell me how i can hold it apart .	because the categ@@ ories tell me how to keep them apart .
4	because the categ@@ ories are telling me how i can hold it apart .	because the categ@@ ories tell me how to keep them apart .
5	because the categ@@ ories are telling me how i can hold it apart .	because the categ@@ ories tell me how to keep them apart .
6	because the categ@@ ories are telling me how i can hold it apart .	because the categ@@ ories tell me how to keep them apart .
7	because the categ@@ ories are telling me how i can hold it apart .	because the categ@@ ories tell me how to keep them apart .
8	because the categ@@ ories are telling me how i can hold it apart .	because the categ@@ ories tell me how to keep them apart .
9	because the categ@@ ories are telling me how i can hold it apart .	because the categ@@ ories tell me how to keep them apart .
...	...	...
20	because the categ@@ ories are telling me how i can keep it apart .	because the categ@@ ories tell me how to keep them apart .
Final	because the categories are telling me how i can keep it apart .	because the categories tell me how to keep them apart .

Table 14: Step-by-step generation process of Differmer(continuous diffusion model) and NeoDiff on IWSLT14 De-En dataset(Part 1/2). NeoDiff converges to the correct translation more quickly and accurately.

Time Step	Differmer Translation	NeoDiff Translation (Ours)
Source: ich sprach also einige monate später bei einer konferenz . **Reference:** so i spoke at a conference a couple months after that .		
0	complex positive which affect o went attac@@ care@@ <pad> gers <pad> david fri@@ level	madeupword0001 le@@ leaders news madeupword0003 sta@@ cannot än@@ madeupword0001 spin@@ published mes@@ exhi@@
1	so i i to a few months later at a conference .	so i spoke at at a a few months later .
2	so i i to a few months later at a conference .	so i spoke at a conference a few months later .
3	so i i about a few months later at a conference .	so i spoke at a conference a few months later .
4	so i i about a few months later at a conference .	so i spoke at a conference a few months later .
5	so i i talking a few months later at a conference .	so i spoke at a conference a few months later .
6	so i i talking a few months later at a conference .	so i spoke at a conference a few months later .
7	so i i talking a few months later at a conference .	so i spoke at a conference a few months later .
8	so i was talking a few months later at a conference .	so i spoke at a conference a few months later .
9	so i was talking a few months later at a conference .	so i spoke at a conference a few months later .
...	...	...
20	so i was talking a few months later at a conference .	so i spoke at a conference a few months later .
Final	so i was talking a few months later at a conference .	so i spoke at a conference a few months later .

Table 15: Step-by-step generation process of Differmer(continuous diffusion model) and NeoDiff on IWSLT14 De-En dataset(Part 2/2). NeoDiff converges to the correct translation more quickly and accurately.

Splits	WMT14 En-De	WMT16 En-Ro	IWSLT14 De-En	QQP	Wiki-Auto	QT
Training	4,500,966	608,319	160,215	144,715	677,751	116,953
Validation	3,000	1,999	7,282	2,048	2,048	2,048
Test	3,003	1,999	6,750	2,500	5,000	10,000

Table 16: The dataset splits used in our experiments.