Timestep Embeddings Trigger Collapse in Diffusion Text Generation

Ryota Nosaka
Tokyo University of Science
1424519@ed.tus.ac.jp

Takuya Matsuzaki
Tokyo University of Science
matuzaki@rs.tus.ac.jp

Abstract

Diffusion models have achieved remarkable success in various generative tasks, particularly in image and audio synthesis, which work by iteratively refining random noise into realistic data. Recent studies have highlighted the potential of diffusion models for text generation, but several challenges remain unresolved. One significant issue is that the model begins to degrade a previous sample rather than improve it after a certain timestep in the generation process, resulting in broken text. In this paper, we reveal that timestep embeddings are a principal cause of the collapse problem by analyzing their interactions with word embeddings. Further, we propose two key methods: (a) a simple lightweight word embedding technique that enhances model analyzability as well as learning efficiency; (b) a novel regularization on both word and timestep embeddings. Experimental results demonstrate that our approach effectively mitigates the collapse problem and can lead to a considerable improvement in the quality of generated text.

1 Introduction

Diffusion models are a class of generative models that have achieved state-of-the-art performance in continuous data generation, such as image and audio synthesis (Ho et al., 2020; Song et al., 2021; Kong et al., 2021). The generation process begins by sampling random noise at timestep T, and then progressively *denoising* it toward timestep 0, resulting in realistic data. Several studies have attempted to adapt diffusion models for text generation via word embeddings and have recently demonstrated performance comparable to earlier autoregressive models (Li et al., 2022; Gong et al., 2023; Yuan et al., 2024; Gao et al., 2024), such as GPT-2 (Radford et al., 2019).

However, diffusion-based text generation still faces challenges in ensuring high-quality output. For text generation, the one-step denoising task at timestep t is typically formulated as fully removing noise and reintroducing a smaller amount of noise corresponding to timestep $t-1$. One critical issue is that the model starts failing to perform the full denoising task after a certain timestep, leading to incoherent or grammatically incorrect output (Gao et al., 2024). We refer to it as the *collapse problem*. This phenomenon is counterintuitive, as the denoising task should gradually become easier as the generation process progresses.

To obtain a high-quality sample, a common approach is to generate multiple times and then select the best one using a re-ranking algorithm like Minimum Bayes-Risk (MBR) decoding (Kumar and Byrne, 2004). However, it compromises diversity, which is a key strength of diffusion models. Gao et al. (2024) reported that the collapse problem can be mitigated by their regularization for word embeddings and heuristically modified training and generation processes. This highlights that learning continuous representations of the vocabulary is a core problem, and motivates us to develop a solution that does not rely on heuristically tweaking the diffusion framework.

Usually, a single denoiser model is shared across all timesteps, and learned timestep embeddings are incorporated as signals of timesteps. Although it has long been common practice, the relation between the timestep embeddings and the collapse problem has not been adequately investigated. Meanwhile, it is also widely adopted to use low-dimensional word embeddings combined with additional projection layers. This approach enhances learning efficiency, but increases model complexity; it introduces non-linearity into the way timestep embeddings act on the model.

In this paper, in pursuit of a fundamental solution to preserve the expected behavior of diffusion models—progressively refining data quality—we investigate both word and timestep embeddings and their relationship. Firstly, we introduce a plug-and-

play low-rank word embedding technique. It makes the effect of timestep embeddings more transparent while keeping computational cost low. Secondly, we propose a new metric that reveals that the timestep embedding disrupts word embeddings in addition to the noise. Moreover, based on our metric, we design a novel regularization method to counteract the adverse effects of timestep embeddings within the standard diffusion architecture. Experiments demonstrated the occurrence of the collapse problem and its mitigation through the combination of our two methods.

2 Related Work

2.1 Text Generation with Diffusion Models

Diffusion models are highly powerful generative models. They first intentionally diffuse real data step by step and learn to reverse this process. New realistic data are then generated by sampling from the stationary distribution and iteratively applying the learned denoising operation.

In recent years, research on diffusion-based text generation has been advancing. Broadly, two primary approaches have been proposed: continuous and discrete diffusion language models. Continuous methods generate word embeddings and discretize them to obtain words. Following diffusion image synthesis methods, such as DDPM (Ho et al., 2020) and DDIM (Song et al., 2021), they are based on Gaussian distributions (Li et al., 2022). Discrete methods work directly in word space and define the forward process as sampling from a categorical distribution, such as gradually replacing words with other words or mask tokens (Austin et al., 2021).

At present, diffusion language models are generally known to have limitations in generating high-quality text. In this paper, we focus on how continuous models work, which are built upon the standard diffusion architecture.

2.2 Word Embeddings

In continuous diffusion text generation, the key distinction from image generation lies in the necessity of constructing continuous representations of words in parallel with learning the denoiser model. Gong et al. (2023) reported a significant performance degradation when using fixed pretrained embeddings, implying that word embeddings need to be optimized for diffusion language models.

It is simultaneously essential to consider the discretization. Typically, a rounding distribution is defined and its likelihood is maximized, which facilitates the segregation of word embeddings. Regarding this, Gao et al. (2024) pointed out that the traditional rounding loss is insufficient to procure the desired distribution and proposed the anchor loss. Their method enhances the distinguishability of embeddings than the rounding loss, although the collapse problem still occurs around the final steps of generation.

2.3 Timestep Embeddings

Timestep embeddings play an important role by conditioning the denoiser model on the noise level of input. For continuous diffusion models, a timestep embedding is typically constructed by mapping a timestep to a vector using sinusoidal encoding and transforming it with a multi-layer perceptron (MLP). Particularly in text generation, research on the impact of timestep embeddings remains limited, whereas there are reports in discrete models. For instance, He et al. (2023) tested several embedding methods of timesteps and reported that they significantly impact on the performance. While this pertains to the discrete method, it emphasizes the importance and challenges of timestep embeddings.

3 Preliminaries

3.1 Diffusion Models

Denoising diffusion probabilistic models (DDPMs) (Ho et al., 2020) are the standard architecture of diffusion models. A DDPM consists of two processes: the forward process and the reverse process.

Given a training data sample $\mathbf{z}_0 \sim q(\mathbf{z}_0)$, the forward process gradually adds noise to $\mathbf{z}_0$, transforming it into random noise $\mathbf{z}_T$:

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

where $0 < \beta_1 < \cdots < \beta_T < 1, \alpha_t = 1 - \beta_t$ are hyperparameters called *noise schedule*. Since Gaussian distributions are reproducible, the distribution of $\mathbf{z}_t$ conditioned on $\mathbf{z}_0$ has a closed form for any timestep t:

$$q(\mathbf{z}_t \mid \mathbf{z}_0) = \mathcal{N}\left(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, \bar{\beta}_t I\right) \quad (1)$$

with $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, $\bar{\beta}_t = 1 - \bar{\alpha}_t$. A denoiser model $p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t)$ is trained using $\{\mathbf{z}_t\}_{t=0}^{T}$.

The reverse process generates a new sample $\mathbf{z}_0$ by starting from random noise $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, I)$ and iteratively denoising via $p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t)$.

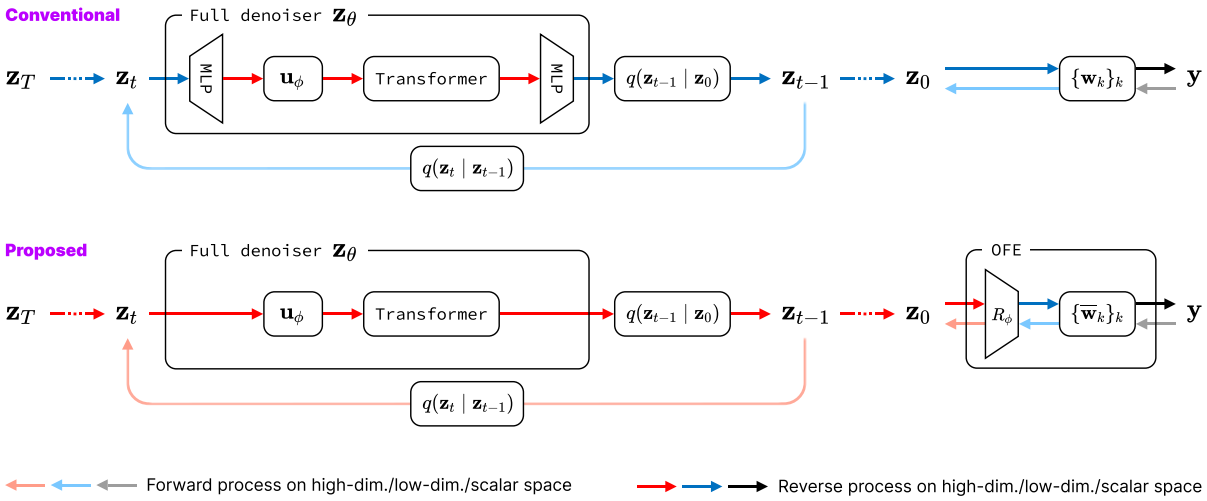


Figure 1: Architecture overview. In the conventional model, the diffusion model (i.e., noisy word embeddings) and Transformer work on spaces of different dimensions and MLPs are unavoidably inserted to bridge them. The proposed model eliminates this discrepancy by consolidating up-/down-projections within the OFE.

3.2 Text Generation with Diffusion Models

Diffusion text generation first generates a sequence of word embeddings $\mathbf{z}_0 = [\mathbf{z}_{0i}]_{i=1}^L$ through denoising and then discretize it into a sequence of words $\mathbf{y} = [y_i]_{i=1}^L$ (Yuan et al., 2024).

Let $\mathbf{w}_k$ be the embedding of the k-th word in the vocabulary ($k \in \{1, 2, ..., V\}$). The forward process begins by sampling each $\mathbf{z}_{0i}$ as follows:

$$q_\phi(\mathbf{z}_{0i} \mid y_i) = \mathcal{N}(\mathbf{z}_{0i}; \mathbf{w}_{y_i}, \beta_0 I)$$

where β_0 is a very small constant.

At timestep t, the one-step denoising is expressed as

$$\begin{aligned} p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t) &= q(\mathbf{z}_{t-1} \mid \mathbf{z}_0 = \mathbf{z}_\theta(\mathbf{z}_t, t)), \\ \mathbf{z}_\theta(\mathbf{z}_t, t) &= \text{Transformer}_\theta(\mathbf{u}_\phi(\mathbf{z}_t, t)), \\ \mathbf{u}_\phi(\mathbf{z}_t, t) &= [\mathbf{z}_{ti} + \mathbf{u}_t]_{i=1}^L . \end{aligned}$$

The full denoiser $\mathbf{z}_\theta$ is a Transformer model (Vaswani et al., 2017). The function $\mathbf{u}_\phi$ fuses a cue of the timestep into noisy word embeddings by adding the timestep embedding. The timestep embedding $\mathbf{u}_t$ is usually parameterized by transforming sinusoidal encoding of the timestep t through an MLP and we follow it. The denoising loss $\mathcal{L}_{\text{denoise}}$ is

$$\begin{aligned} \mathcal{L}_{\text{denoise}} = \mathbb{E}_{\mathbf{y}, \mathbf{z}_{0:T}} \Bigg[\sum_{t=2}^{T} \| \mathbf{z}_0 - \mathbf{z}_\theta(\mathbf{z}_t, t) \|^2 \\ + \| \mathbf{w}_{y_i} - \mathbf{z}_\theta(\mathbf{z}_1, 1) \|^2 + \| \sqrt{\bar{\alpha}_T} \mathbf{z}_0 \|^2 \Bigg]. \end{aligned}$$

In practice, we sample $t \in \{1, 2, ..., T\}$ for each minibatch.

Minimizing only $\mathcal{L}_{\text{denoise}}$ would cause all word embeddings to collapse into a single point because $\mathcal{L}_{\text{denoise}}$ is mainly composed of the mean squared error between $\mathbf{z}_0$ and $\mathbf{z}_\theta(\mathbf{z}_t, t)$. Therefore, we need to promote appropriate segregation among word embeddings. Concretely, we define the distribution for rounding an embedding $\mathbf{w} \in \mathbb{R}^d$ to the k-th word as

$$p_\phi(k \mid \mathbf{w}) = \frac{\exp \mathbf{w}^\top \mathbf{w}_k}{\sum_{\ell=1}^V \exp \mathbf{w}^\top \mathbf{w}_\ell} \tag{2}$$

and the rounding loss $\mathcal{L}_{\text{round}}$ as

$$\mathcal{L}_{\text{round}} = \frac{1}{L} \sum_{i=1}^{L} \mathbb{E}_{\mathbf{y}, \mathbf{z}_0} \left[-\log p_\phi \left(y_i \mid \mathbf{z}_{0i} \right) \right].$$

Another approach is the anchor loss (Gao et al., 2024) that uses the full denoising prediction instead of training data samples:

$$\mathcal{L}_{\text{anchor}} = \frac{1}{L} \sum_{i=1}^{L} \mathbb{E}_{\mathbf{y}, \mathbf{z}_t} \left[-\log p_\phi \left(y_i \mid \mathbf{z}_\theta(\mathbf{z}_t, t)_i \right) \right].$$

In this work, we choose the rounding loss, which is the standard way. The total loss $\mathcal{L}_{\text{total}}$ is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{denoise}} + \mathcal{L}_{\text{round}}.$$

Computing $\mathcal{L}_{\text{round}}$ is expensive because it is performed over the entire target sequence. On the other hand, using low-dimensional word embeddings and a smaller Transformer hurts prediction performance (see §5). Hence, in order to reduce

	d'	d & Hidden Dim.	Feed-Forward Dim.	Layers	Attention Heads
Base	-	768	2048	6	12
Low-d	-	132	2048	6	12
OFE	128	768	2048	6	12

Table 1: Hyperparameters in the exploratory experiments.

computational costs while keeping the capacity of Transformer, it is common practice to use low-dimensional word embeddings and append MLPs for up- and down-projection before $\mathbf{u}_\phi$ and after the Transformer (see Figure 1).

In reverse process, the prediction of $\mathbf{z}_{t-1}$ given the previous prediction of $\mathbf{z}_t$ is sampled by

$$\mathbf{z}_{t-1} = \sqrt{\bar{\alpha}_t}\mathbf{z}_\theta(\mathbf{z}_t, t) + \sqrt{\bar{\beta}_t}\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, I). \tag{3}$$

Then the last output $\mathbf{z}_\theta(\mathbf{z}_1, 1)$ are rounded to words.

The simplest way to perform a sequence-to-sequence task is to use an encoder-decoder Transformer. The source text is input to the encoder, and the noisy target text $\mathbf{z}_t$ to the decoder with non-causal attention.

4 Lightweight Word Embeddings

The conventional MLP-based method for reducing the dimensionality of word embeddings is effective but complicates the interaction between word and timestep embeddings; the MLP first non-linearly transforms noisy word embeddings, and then the timestep embedding is applied. Besides, it may also extract information about timesteps according to the noise level of input, making it difficult to analyze the conditioning by timesteps.

This MLP is actually not required if the model has no such dimensional mismatch. To eliminate this entanglement, we propose the *Orthogonally Factorized Embedding* (OFE) technique that employs low-dimensional embeddings while letting both the diffusion model (i.e., noisy word embeddings) and Transformer work on high-dimensional space (see Figure 1). The OFE consists of low-dimensional word embeddings $\{\overline{\mathbf{w}}_k \in \mathbb{R}^{d'}\}_k$ along with a learned column-orthogonal matrix $R_\phi \in \mathbb{R}^{d \times d'}$. Every word embedding $\mathbf{w}_k$ is expressed as

$$\mathbf{w}_k = R_\phi \overline{\mathbf{w}}_k.$$

The rounding distribution can be rewritten as

$$p_\phi(k \mid \mathbf{w}) = \frac{\exp(R_\phi^\top \mathbf{w})^\top \overline{\mathbf{w}}_k}{\sum_{\ell=1}^V \exp(R_\phi^\top \mathbf{w})^\top \overline{\mathbf{w}}_\ell}.$$

Notably, for every word embedding $\mathbf{w}_m = R_\phi \overline{\mathbf{w}}_m$,

$$p_\phi(k \mid \mathbf{w}_m) = \frac{\exp(R_\phi^\top R_\phi \overline{\mathbf{w}}_m)^\top \overline{\mathbf{w}}_k}{\sum_{\ell=1}^V \exp(R_\phi^\top R_\phi \overline{\mathbf{w}}_m)^\top \overline{\mathbf{w}}_\ell} = \frac{\exp \overline{\mathbf{w}}_m^\top \overline{\mathbf{w}}_k}{\sum_{\ell=1}^V \exp \overline{\mathbf{w}}_m^\top \overline{\mathbf{w}}_\ell} \tag{4}$$

since $R_\phi^\top R_\phi = I$. Consequently, $p_\phi(k \mid \mathbf{w})$ reduces to the rounding in the low-dimensional embedding space.

Here, we provide the rationale behind the column-orthogonal constraint. Since the rounding loss promotes appropriate separation of word embeddings as noted in §3.2, the reconstruction of a low-dimensional embedding from a high-dimensional representation, i.e., $R_\phi^\top R_\phi \overline{\mathbf{w}}_m = \overline{\mathbf{w}}_m$ as can be seen in Eq. (4), seems to be naturally acquired due to the rounding loss even when using an unconstrained matrix for R_ϕ. However, we empirically found that this is not the case; in the course of training, $R_\phi^\top R_\phi$ gets close to a scaled identity matrix λI and λ gets larger and larger, and training becomes unstable. We conjecture that the reason of this phenomenon is that λ works similarity to the inverse temperature in a temperature softmax. Therefore, we attempt to remedy this problem by eliminating a scaling ambiguity between R_ϕ and low-dimensional embeddings $\overline{\mathbf{w}}_k$, i.e., the same high-dimensional embedding is obtained with γR_ϕ and $\gamma^{-1}\overline{\mathbf{w}}_k$ for any $\gamma \neq 0$: $\mathbf{w}_k = (\gamma R_\phi)\left(\gamma^{-1}\overline{\mathbf{w}}_k\right)$. For that, we choose to impose column-orthogonality on R_ϕ, thereby constraining its Frobenius norm. In addition, it enables the reconstruction by multiplying $R_\phi^\top$ from the left, which ensures that the high-dimensional rounding $p_\phi(k \mid \mathbf{w})$ is equivalent to the low-dimensional rounding. We expect that this property further contributes to stability.

Because a matrix can be parameterized under the column-orthogonal constraint,[1] we can employ the OFE by simply replacing the existing embedding and rounding functions in a model with those of the OFE.

[1] https://pytorch.org/docs/stable/generated/torch.nn.utils.parametrizations.orthogonal.html

5 Exploratory Experiments

In this section, we demonstrate the collapse problem and confirm that it is not attributable to the dimensionality reduction by comparing the three cases of denoisers shown in Table 1. `Base` does not use low-dimensional embeddings. In `Low-d`, $d = 132$ was chosen as the best feasible alternative to 128 ($= d'$ in `OFE`) conforming with 12 attention heads. We conducted experiments on Quora Question Pairs (DataCanary et al., 2017) and measured the BERTScore (Zhang et al., 2020) of the paraphrased questions generated by the models. Further details are described in the main experiments (§7), including the reason for setting $d' = 128$ in `OFE`.

Figure 2 illustrates the evaluation of intermediate samples during the reverse process, namely the full denoising predictions $\mathbf{z}_\theta(\mathbf{z}_t, t)$ at each timestep t. The collapse problem was observed in all models, including `Base`. Thus, the collapse is not due to the dimensionality reduction techniques.

`Low-d` expectedly exhibited worse performance than `Base`. By contrast, `OFE` substantially outperformed `Low-d` and stood comparison with `Base`. Furthermore, it is noteworthy that `OFE` surpassed `Base` early in generation.

We found that models before convergence hardly exhibit collapse as shown in Figure 3. These findings suggest that continuing training only with the traditional loss makes the vectors $\mathbf{u}_\phi(\mathbf{z}_t, t)$, the input to the Transformer, more difficult to handle.

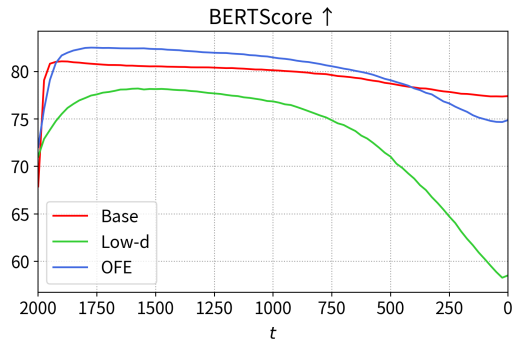


Figure 2: The evaluation of full denoising predictions $\mathbf{z}_\theta(\mathbf{z}_t, t)$ at each timestep t in the exploratory experiments.

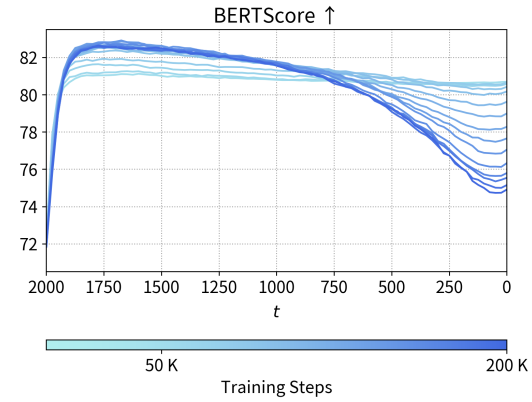


Figure 3: The progression of the collapse problem in `OFE` in the course of training.

6 Study of Timestep Embeddings

The model has been simplified by the OFE, which allows word embeddings to naturally work on the same dimension as timestep embeddings (Figure 1). Thus, we are now able to analyze the relationship between these two embeddings directly. In this section, we first introduce a new metric that measures how timestep embeddings are implicated in the phenomenon that the vectors $\mathbf{u}_\phi(\mathbf{z}_t, t)$ tend to form an undesired structure. We then transform this score to a regularization method to mitigate the collapse problem.

6.1 Analysis of Ambiguity

A denoiser model learns the mapping from a hidden variable $\mathbf{z}_{0i}$ to a word y_i via the rounding loss. In addition, each vector in $\mathbf{u}_\phi(\mathbf{z}_t, t)$, the input of the denoiser, must be properly distinguished from one another, particularly at early diffusion steps.

The forward process is performed by shrinking word embeddings toward the origin and then adding Gaussian noise (Eq. (3); Figure 4). Since the model learns the inverse operation of the forward process, the variance of the full denoising prediction $\mathbf{z}_{0i} = \mathbf{z}_\theta(\mathbf{z}_t, t)_i$ for the input $\mathbf{z}_{ti}$ is expected to decrease as the generation progresses, and hence the sample is gradually determined. However, due to the timestep embedding, the actual input vector $\mathbf{z}_{ti} + \mathbf{u}_t$ may be confused with unrelated words *at different timesteps* (Figure 5).

This hypothesis motivates us to measure which words are now mistaken for which words due to timestep embeddings. Inspired by Kullback-Leibler divergence, we define the *Temporal Ambiguity Score* (TAS) between t and t' as follows:[2]

$$\mathrm{Ambig}\,(t, t') := \frac{1}{V} \sum_{k=1}^{V} \sum_{\ell=1}^{V} \mathbb{E}_{\mathbf{w}_{tk}, \mathbf{w}_{t'k}} \Bigg[p_\phi\left(\ell \mid \mathbf{u}_\phi(\mathbf{w}_{t'k}, t')\right) \log \frac{p_\phi\left(\ell \mid \mathbf{u}_\phi(\mathbf{w}_{t'k}, t')\right)}{p_\phi\left(\ell \mid \mathbf{u}_\phi(\mathbf{w}_{tk}, t)\right)} \Bigg]$$

[2]The rounding p_ϕ always denotes the softmax distribution over unmodified word embeddings, as shown in Eq. (2).

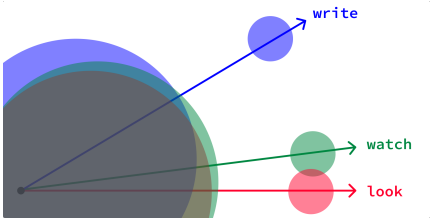


Figure 4: Noise addition at timesteps $0.2T$ and $0.8T$. The circles represent the regions from which $\mathbf{z}_t$ is mostly sampled. The regions move closer to the origin and expand their radii as the forward process progresses.

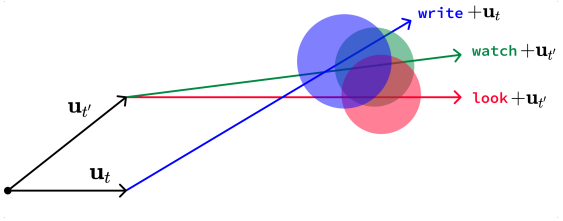


Figure 5: How the timestep embeddings disrupt noisy word embeddings. The noisy "write" at timestep t will be confused with the noisy "watch" and "look" at timestep t'.

where $\mathbf{w}_{tk}$ is a noisy embedding of the k-th word at timestep t. This metric represents how the rounding results of noisy word embeddings are altered by timestep embeddings. It is based on a fact that is peculiar to text generation: unlike in image synthesis, the effect of noise addition lies in whether the rounding yields different words before and after it.

Since exact calculation of $\mathrm{Ambig}\,(t, t')$ is challenging, we estimate it by sampling $(\mathbf{w}_{tk}, \mathbf{w}_{t'k})$ once for each k.

6.2 Regularization for Disambiguation

We consider using the ambiguity score as an objective function. In imitation of the rounding loss, we introduce the *Temporal Disambiguation Loss* using training data samples instead of entire vocabulary:

$$\mathcal{L}_{\text{disambig}} \coloneqq \frac{1}{L}\sum_{i=1}^{L}\sum_{j=1}^{L}\mathbb{E}_{\mathbf{y},\mathbf{z}_t,\mathbf{z}_{t'}\sim q}\Bigg[p_\phi(y_j \mid \mathbf{u}_\phi(\mathbf{z}_{t'i}, t'))\log\frac{p_\phi(y_j \mid \mathbf{u}_\phi(\mathbf{z}_{t'i}, t'))}{p_\phi(y_j \mid \mathbf{u}_\phi(\mathbf{z}_{ti}, t))}\Bigg].$$

Since directly minimizing it is difficult, we further simplify the loss. First, we fix t' at 0, because it is not meaningful to disambiguate embeddings at late diffusion steps, which are distributed almost randomly. Thus, we get

$$\mathcal{L}'_{\text{disambig}} = \frac{1}{L}\sum_{i=1}^{L}\sum_{j=1}^{L}\mathbb{E}_{\mathbf{y},\mathbf{z}_t\sim q}\Bigg[p_\phi(y_j \mid \mathbf{w}_{y_i})\log\frac{p_\phi(y_j \mid \mathbf{w}_{y_i})}{p_\phi(y_j \mid \mathbf{u}_\phi(\mathbf{z}_{ti}, t))}\Bigg].$$

Besides, the model easily learns so that $p_\phi(y_j \mid \mathbf{w}_{y_i}) \approx 1$ if $i = j$, and 0 otherwise. Ignoring noise for training stability, we arrive at the *Simplified Temporal Disambiguation Loss* (STDL):

$$\mathcal{L}^{\text{simple}}_{\text{disambig}} \coloneqq \frac{1}{L}\sum_{i=1}^{L}\mathbb{E}_{\mathbf{y},t\sim q}\big[-\log p_\phi\left(y_i \mid \mathbf{u}_\phi\left(\sqrt{\bar{\alpha}_t}\mathbf{w}_{y_i}, t\right)\right)\big].$$

Note that, from Eq. (1), $\sqrt{\bar{\alpha}_t}\mathbf{w}_{y_i}$ is the expectation of $q(\mathbf{z}_{ti} \mid \mathbf{z}_{0i} = \mathbf{w}_{y_i})$. If we set $\bar{\alpha}_0 = 1$ and ignore the timestep embedding (i.e., $\mathbf{u}_0 = \mathbf{0}$), $\mathcal{L}^{\text{simple}}_{\text{disambig}}$ equals $\mathcal{L}_{\text{round}}$ when $t = 0$. Hence, this is a generalization of the conventional loss associated with rounding. Intuitively, it aims to obtain consistency across all timesteps in the space of $\mathbf{u}_\phi(\cdot, t)$ with respect to the noiseless word embedding space.

7 Main Experiments

7.1 Setup

We set the number of diffusion steps to $T = 2000$ and use the *sqrt* noise schedule (Li et al., 2022), which are widely used. We choose $d' = 128$ to align with baselines that utilize the MLP-based dimensionality reduction method. The output length is adjusted by generating padding tokens. The embedding of the padding token is learned as well as other word embeddings.

`Base` and `OFE` are identical to those used in §5. `OFE+STDL` refers to the model incorporating both OFE and STDL.

The generation proceeds step by step. The results were saved every 25 steps and the final step[3] to reduce the computational cost of evaluation.

Datasets We conduct experiments on two popular sequence-to-sequence tasks: Paraphrasing on Quora Question Pairs (DataCanary et al., 2017) and Text Simplification on Wiki-Auto (Jiang et al., 2020). The data split follows that used for DiffuSeq (Gong et al., 2023).

Metrics The quality of samples is evaluated using BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004) and BERTScore (Zhang et al., 2020)[4]. The diversity among outputs generated from the same

[3] That is, $t = 2000, 1975, \ldots, 25, 1$.

[4] Following DiffuSeq, we use `microsoft/deberta-xlarge-mnli` for computing BERTScore.

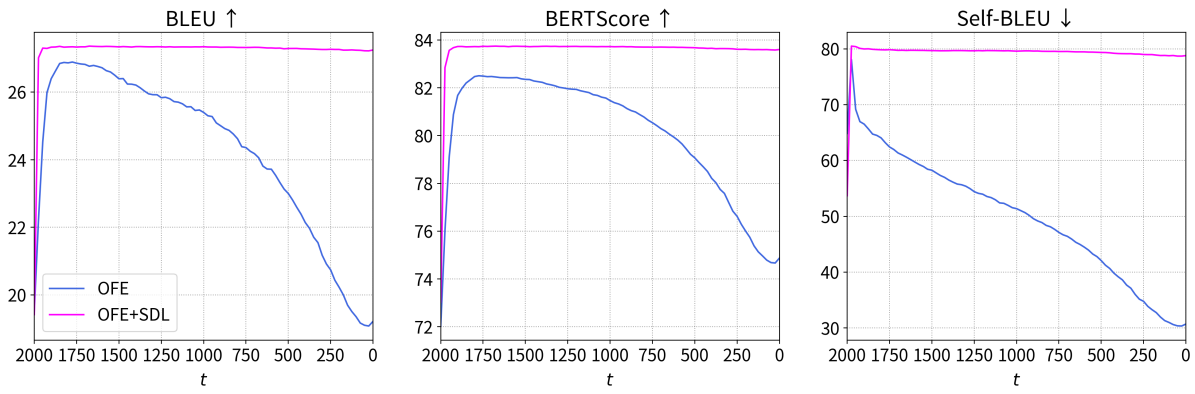
	MBR	BLEU	ROUGE-L	BERTScore	Self-BLEU
DiffuSeq	-	18.29†	52.99†	79.30†	27.32
DiffuSeq	10	24.13	58.80	83.65	-
SeqDiffuSeq	-	23.28	-	82.91	-
SeqDiffuSeq	10	24.34	-	84.00	-
Differmer	-	28.52	60.15	83.80	-
Differmer	10	30.43	61.25	85.02	-
Differmer	20	30.52	61.08	85.02	-
Base	-	22.19 (−2.38)	53.27 (−3.74)	77.39 (−3.66)	59.19 (+ 0.12)
OFE	-	19.21 (−7.68)	50.29 (−9.34)	74.86 (−7.64)	30.67 (+ 0.34)
OFE	10	25.85	57.73	80.64	-
OFE+STDL	-	27.24 (−0.12)	60.29 (−0.11)	83.60 (−0.14)	78.76 (+25.08)
OFE+STDL	10	27.90	60.90	84.20	-
OFE+STDL	20	28.05	61.02	84.30	-

(a) Quora Question Pairs

	MBR	BLEU	ROUGE-L	BERTScore	Self-BLEU
DiffuSeq	-	29.29†	53.13†	77.81†	46.42
DiffuSeq	10	36.22	58.49	81.26	-
SeqDiffuSeq	-	37.09	-	82.11	-
SeqDiffuSeq	10	37.12	-	82.14	-
Differmer	-	40.37	59.56	81.96	-
Differmer	10	40.77	59.86	82.21	-
Differmer	20	40.84	59.88	82.29	-
Base	-	27.85 (−11.68)	50.99 (− 8.78)	72.98 (−9.19)	58.32 (+ 1.43)
OFE	-	30.20 (−12.71)	49.78 (−10.03)	72.70 (−9.38)	48.51 (− 0.00)
OFE	10	39.82	58.15	80.25	-
OFE+STDL	-	41.45 (− 0.12)	59.02 (− 0.07)	81.75 (−0.09)	96.16 (+20.01)
OFE+STDL	10	41.49	59.12	81.84	-
OFE+STDL	20	41.49	62.35	81.84	-

(b) Wiki-Auto

Table 2: Generation qualities and diversities in the main experiments. Difference between the best sample through generation process and the final output are shown in parentheses. † indicates that we evaluated the samples released by the authors. Other baseline results are cited from their paper.



Quora Question Pairs

Figure 6: The evaluation of full denoising predictions $\mathbf{z}_\theta(\mathbf{z}_t, t)$ at each timestep t in the main experiments.

input is a unique strength of diffusion models. To measure this, we employ Self-BLEU (Zhu et al., 2018).

Baselines DiffuSeq (Gong et al., 2023) is a basic diffusion language model. SeqDiffuSeq (Yuan et al., 2024) is an advanced encoder-decoder model with adjusting noise schedule during training. Difformer (Gao et al., 2024) is a model designed to mitigate the collapse problem by improving its training objective and generation process.

7.2 Generation Quality and Diversity

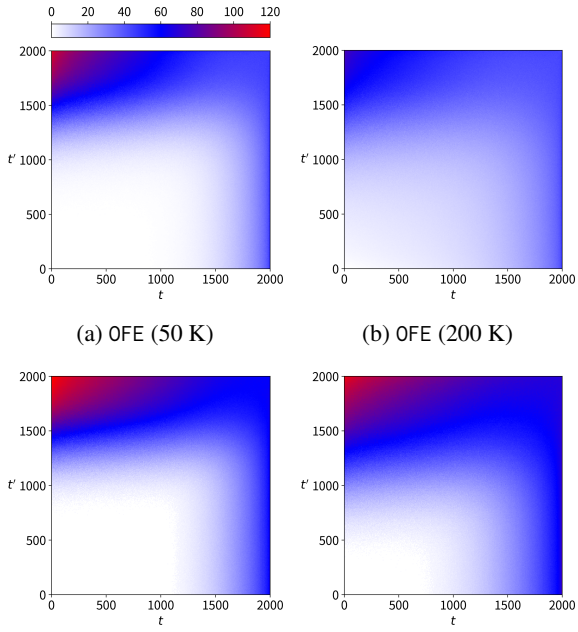
Table 2 shows the evaluation of the final outputs. As we saw in §5, the collapse problem occurred in `Base` and `OFE`, where BLEU, ROUGE-L and BERTScore deteriorated in the course of sampling. Their final results were roughly comparable performance to DiffuSeq as expected, since DiffuSeq is largely equivalent to the conventional model in Figure 1. However, `OFE`'s BERTScore is lower than DiffuSeq across all datasets. It suggests that the OFE may cause more severe collapse than the MLP-based method. `OFE+STDL` showed essentially comparable performance to Difformer, although it occasionally underperformed Difformer when combined with MBR.

`OFE+STDL` without MBR was substantially better than `OFE` with MBR. It is remarkable that the STDL achieves better performance without prolonged generation times, the drawback of MBR.

Figure 6 presents the assessment of intermediate outputs in the reverse process. In contrast to `OFE`, `OFE+STDL` consistently maintained generation quality throughout the reverse process. Moreover, on Quora Question Pairs, the STDL not only suppressed the collapse but also entirely improved sample quality.

The diversity unfortunately fell with STDL. However, as shown in Figure 6, the improvement in Self-BLEU of `OFE`'s output progresses in tandem with a decline in BLEU and BERTScore. This suggests that the diversity previously reported in diffusion language models may actually be an illusion arising from corrupted samples. Even if the collapse is partially tolerated, since Self-BLEU should ideally be comparable to BLEU, further improvements are required for text generation to achieve genuine diversity.

As a side note, employing MBR is also likely to reduce diversity. Comparing MBR = 10 and 20 of Difformer on Quora Question Pairs and `OFE+STDL` on Wiki-Auto, although the sequence-level similarity to the reference data measured by BLEU or ROUGE-L increased, the semantic similarity indicated by BERTScore remains unchanged.



(a) `OFE` (50 K) (b) `OFE` (200 K)

(c) `OFE+STDL` (50 K) (d) `OFE+STDL` (500 K)

Quora Question Pairs

Figure 7: Temporal Ambiguity Score of the checkpoints at the training steps indicated in parentheses.

	Checkpoint Step
`Base`	500 K
`OFE`	200 K
`OFE+STDL`	500 K

Quora Question Pairs

Table 3: The training step at which the evaluation checkpoint was saved.

7.3 Ambiguity across Timesteps

As observed in §5, the collapse problem in `OFE` becomes more severe as training progresses. To analyze this phenomenon, we compare the TAS between checkpoints that do and do not exhibit collapse. Figure 7 depicts the TAS for these checkpoints of `OFE` and `OFE+STDL`. In the early stage of training, the ambiguity is low for small timesteps and high for large timesteps as expected. However, as training of `OFE` advances, the ambiguity increases for small timesteps while decreasing for large timesteps. This suggests that the conventional loss function excessively focuses on constructing timestep embeddings for high noise levels while

neglecting those for small noise levels. By contrast, in OFE+STDL, the TAS remains relatively stable throughout training, preserving the distinguishability of the non-collapsing embeddings space.

7.4 Training Efficiency

Table 3 presents the training steps corresponding to the evaluation checkpoints. OFE was trained faster than Base, even though the former achieved performance comparable to the latter. We also observed that the STDL led to slower convergence.

8 Conclusion

In this work, we investigated the collapse problem through careful observation of the emergence of the phenomenon and the lens of how timestep embeddings influence word embeddings. To address this challenge, we proposed a principled dimensionality reduction technique and a regularization method that acts on both embeddings. Our methods are simple and easy to employ, yet dramatically mitigate the collapse problem. In addition, we revisited generation diversity of diffusion language models, and suggested its intrinsic difficulty.

9 Limitations

One limitation of our study is that it remains unclear whether the STDL mitigates only the collapse problem or may also reduce the generation of desirable diversity. That said, our findings call for further investigation into timestep embeddings for diffusion models and diversity in text generation.

Acknowledgments

This work was partially supported by KIOXIA Corporation.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured Denoising Diffusion Models in Discrete State-Spaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993. Curran Associates, Inc.

DataCanary, hilfialkaff, Lili Jiang, Meg Risdal, Nikhil Dandekar, and tomtung. 2017. Quora Question Pairs. Kaggle.

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. Association for Computational Linguistics.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models. In *International Conference on Learning Representations*.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing 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. Association for Computational Linguistics.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. 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. Association for Computational Linguistics.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2021. 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*, pages 169–176. Association for Computational Linguistics.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-LM Improves Controllable Text Generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 4328–4343. Curran Associates, Inc.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A Benchmarking Platform for Text Generation Models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100. Association for Computing Machinery.