Training-Free Self-Correction for Multimodal Masked Diffusion Models

Yidong Ouyang[1], Panwen Hu[2], Zhengyan Wan[3], Zhe Wang[4], Liyan Xie[5], Dmitriy Bespalov[6], Ying Nian Wu[1], Guang Cheng[1], Hongyuan Zha[7], and Qiang Sun[2,8]

[1]University of California, Los Angeles
[2]Mohamed bin Zayed University of Artificial Intelligence
[3]East China Normal University
[4]University of Virginia
[5]University of Minnesota
[6]Drexel university
[7]The Chinese University of Hong Kong, Shenzhen
[8]University of Toronto

Abstract

Masked diffusion models have emerged as a powerful framework for text and multimodal generation. However, their sampling procedure updates multiple tokens simultaneously and treats generated tokens as immutable, which may lead to error accumulation when early mistakes cannot be revised. In this work, we revisit existing self-correction methods and identify limitations stemming from additional training requirements or reliance on misaligned likelihood estimates. We propose a training-free self-correction framework that exploits the inductive biases of pre-trained masked diffusion models. Without modifying model parameters or introducing auxiliary evaluators, our method significantly improves generation quality on text-to-image generation and multimodal understanding tasks with reduced sampling steps. Moreover, the proposed framework generalizes across different masked diffusion architectures, highlighting its robustness and practical applicability. Code can be found in `https://github.com/huge123/FreeCorrection`.

1 Introduction

Masked diffusion models [2, 13, 30, 22, 1, 26, 32, 29] have recently emerged as a powerful paradigm for text [24, 40, 5] and multimodal generation [37, 38]. By iteratively corrupting data through a masking process and learning to reverse this process, masked diffusion models enable parallel generation and offer an appealing alternative to autoregressive decoding. A growing body of work has shown that this framework scales well with model size and data diversity, and can be seamlessly integrated into large pre-trained architectures.

Despite their empirical success, masked diffusion models suffer from a fundamental limitation that is not yet fully solved. At each reverse step, masked diffusion models update a number of tokens simultaneously, and once a masked token is generated, it is typically treated as immutable in subsequent steps. As a result, errors introduced at early stages of generation may persist and accumulate, leading to mismatches between synthetic distribution and the target data distribution. An explanation of this phenomenon can be established from the perspective of continuous-time Markov chains (CTMCs) [3]. The practical reverse sampling procedure employed by masked

diffusion models is mathematically equivalent to a τ-leaping approximation of the underlying reverse-time CTMC [11]. This approximation marginalizes out exact jump times and performs parallel, approximate updates across all masked dimensions. While τ-leaping enables efficient and highly parallel generation, it is well known in the numerical analysis of CTMCs to introduce path-wise inaccuracies when incorrect jumps occur early and cannot be revised.

In this paper, we re-examine existing approaches that aim to mitigate generation errors via self-correction. Prior work typically relies on either (i) fine-tuning or retraining additional models to explicitly evaluate or re-score generated tokens [32, 42, 15, 28, 14], or (ii) reusing likelihood estimates obtained several steps earlier in the reverse process [33]. While effective in some settings, these approaches often introduce additional training costs and rely on misaligned signals. In contrast, our goal is to develop a training-free correction mechanism that operates directly on the reverse process and remains faithful to the underlying generative dynamics.

Our key insight is that masked diffusion networks already possess useful inductive biases that can be exploited for self-correction without additional training. By carefully analyzing the structure of the reverse process and the model's conditional predictions, we identify more fine-grained design choices that allow the model to reassess and refine uncertain updates during generation. Importantly, our method does not require modifying the pre-trained model or introducing auxiliary evaluators.

We validate our approach through GenEval benchmark [8] on text-to-image generation and VLMEvalKit [6] on several multimodal understanding benchmarks. To the best of our knowledge, this is the first work to study self-correction mechanisms in multimodal masked diffusion models. Our results show that the proposed method significantly improves generation fidelity and semantic alignment, while simultaneously enabling faster sampling without sacrificing output quality. Moreover, the approach generalizes across different masked diffusion architectures, highlighting its robustness and practical applicability. Together with detailed ablation studies on the design of remasking criteria, our findings suggest that using the inductive bias of pre-trained masked diffusion models for remasking opens new avenues for efficient and principled self-correction at inference time.

2 Related Work

Masked Diffusion Models. Masked diffusion models (MDMs) generalize the denoising diffusion paradigm to discrete token spaces by corrupting data with a masking forward process and learning the corresponding reverse dynamics. Early work studied discrete diffusion for discrete variables and established practical training and sampling objectives for masked corruption and parallel decoding [2, 13, 30, 22, 29]. Recent advances have focused on scaling masked diffusion to large language model settings and improving decoding efficiency/quality via better schedules and token-selection rules such as Top-K updates [24, 40, 5].

Beyond pure text, MDMs have also been extended to multimodal generation and unified vision-language modeling by first adopting VQ-VAE [31] to convert images to discrete tokens, and then operating over image and text tokens through large-scale pretraining [37, 38]. These models inherit the key advantage of diffusion-style parallel generation, but also amplify a known issue of discrete parallel sampling: once tokens are filled, they are typically treated as absorbing states, making early mistakes hard to revise.

Self-Correction of Masked Diffusion Models. A growing line of work investigates how to endow MDMs with self-correction so that previously generated tokens can be revised. From a continuous-time Markov chain (CTMC) perspective, several methods adapt predictor-corrector style sampling to reduce discretization errors in the reverse dynamics [3, 18, 7, 4, 27]. Another family of

approaches introduces explicit remasking in the reverse process: tokens can be re-masked according to a schedule and then re-generated, thereby enabling correction during sampling [33, 28].

Existing self-correction methods mainly differ in whether they require additional training. Some approaches generalize the diffusion process and train new models to support correction and improved reverse transitions [32, 28]. Others estimate per-token quality/confidence with auxiliary networks or by fine-tuning, and then use these estimates to decide which tokens to remask or re-sample [42, 15, 14]. In contrast, our goal is to achieve *training-free* self-correction for *multimodal* MDMs by exploiting the inductive bias already present in pre-trained masked diffusion backbones, without modifying model parameters or introducing external evaluators.

3   Preliminaries

3.1   Formulation of Masked Diffusion Model

Masked diffusion models [2, 13, 30, 22, 1, 26, 32, 29, 24] are characterized by the forward and backward processes. In the forward process, we gradually replace each word in a sentence with a special mask token until all the words are replaced. In the backward process, the model learns to gradually complete the sentence from all mask tokens.

In this work, we follow the Continuous-Time Markov Chain (CTMC) framework used in [3, 7, 4, 20] to model the masked diffusion models. We consider a $\mathcal{D}$-dimensional discrete state space $\mathcal{S}^{\mathcal{D}}$, where $\mathcal{S}$ denotes the finite state space of each dimension. The data distribution is denoted by q_0, with probability mass function $q_0(x)$ over $\mathcal{S}^{\mathcal{D}}$, and a data sample is written as $x_0 \in \mathcal{S}^{\mathcal{D}}$.

Forward Process.   For a given noise level $t \in [0,1]$, the forward process $x_t \sim q_{t|0}\left(x_t|x_0\right)$ is a coordinate-independent masking process via $q_{t|0}(x_t|x_0) = \prod_{d=1}^{\mathcal{D}} q_{t|0}^d(x_t^d|x_0^d)$, where

$$q_{t|0}^d(x_t^d|x_0^d) = (1-\alpha_t)\delta_m(x_t^d) + \alpha_t\delta_{x_0^d}(x_t^d),\ d \in [\mathcal{D}], \tag{1}$$

here α_t is the predefined noise schedule satisfying $\alpha_0 = 1, \alpha_1 = 0$; $\delta_x(z)$ denotes Kronecker delta satisfying $\delta_x(z) = 1$ if $x = z$ and $\delta_x(z) = 0$ if $x \neq z$; and m denotes the one-hot embedding of mask. In other words, for each coordinate i, at forward time step t, x_t^i is replaced by the mask token with probability $1 - \alpha_t$ and remains unchanged otherwise [3].

Reverse Process.   The oracle reverse process of the above forward process, conditional on the initial state x_0, is denoted as $q_{s|t}\left(x_s|x_t, x_0\right) = \prod_{d=0}^{D} q_{s|t}\left(x_s^d|x_t, x_0\right)$ for any $s < t$, where

$$q_{s|t}\left(x_s^d|x_t, x_0\right) = \begin{cases} \delta_{x_t^d}(x_s^d), & \text{if } x_t^d \neq m, \\ \frac{1-\alpha_s}{1-\alpha_t}\delta_m(x_s^d) + \frac{\alpha_s - \alpha_t}{1-\alpha_t}\delta_{x_0^d}(x_s^d), & \text{if } x_t^d = m. \end{cases} \tag{2}$$

Training Objective.   A neural network p_θ parametrized by θ can be trained to approximate the oracle reverse process by minimizing the Kullback–Leibler (KL) divergence between the true reverse conditional distribution $q_{s|t}\left(x_s^d|x_t, x_0\right)$ and $q_{s|t}\left(x_s^d|x_t, p_\theta^d\left(x_t, t\right)\right)$, where $p_\theta^d\left(x_t, t\right)$ denotes the d-th coordinate of the network output. The training objective can be summarized as

$$\mathcal{L}_\theta = \int_0^1 \frac{\alpha_t'}{1-\alpha_t} \underset{\substack{x_0 \sim q_0 \\ x_t \sim q_{t|0}(x_t|x_0)}}{\mathbb{E}} \left[\sum_{d=1}^{D} \delta_m(x_t^d) x_0^{d\top} \log p_\theta^d\left(x_t, t\right)\right] dt.$$

3

Here, $\alpha'_t = \frac{d\alpha_t}{dt}$. In practice, a time-embedding-free architecture for the denoising network, i.e., $p_\theta(x_t, t) = p_\theta(x_t)$, is usually employed as x_t implicitly contains information about t via the number of masked tokens.

Sampling Process. After learning the network $p_\theta(x_t)$, samples can be generated by simulating the reverse process in Eq. (2). Since the reverse dynamics factorize across dimensions, sampling can be performed independently for each coordinate. Specifically, for each masked position d at time t, a transition from the mask token occurs during the interval $(s, t]$ with probability $\frac{\alpha_s - \alpha_t}{1 - \alpha_t}$; conditioned on such a transition, the new token is sampled from the categorical distribution $p_\theta^d(x_t)$.

3.2 The Limitation of Masked Diffusion Model

To understand the limitations of the sampling process in masked diffusion models better, we present a detailed discussion on the coordinate-wise independent, time-inhomogeneous CTMC below.

First of all, it can be observed that, for each dimension d, the forward masking process admits the following time-varying transition rate:

$$Q_t^{(f)}(x_0^d \to m) = \lambda(t) := -\frac{\alpha'_t}{\alpha_t},$$

since the resulting transition probability satisfies

$$q(x_t^d = x_0^d | x_0^d) = \exp\left(-\int_0^t \lambda(\tau)\, d\tau\right) = \alpha_t,$$

which recovers the predefined noise schedule in Eq. (1).

The reverse process corresponds to the time-reversed CTMC, which progressively transforms the masked state back into samples from the data distribution. The reverse-time transition rate can be computed in two equivalent ways. One approach, adopted in [3, 23], expresses the reverse rate matrix $Q_t^{(r)}$ in terms of the forward rate as,

$$Q_t^{(r)}(x_t \to x_s) = Q_t^{(f)}(x_s \to x_t) \sum_{x_0} \frac{q_{t|0}(x_s|x_0)}{q_{t|0}(x_t|x_0)}\, q_{0|t}(x_0|x_t), x_t \neq x_s. \tag{3}$$

Alternatively, following [7, 4], the reverse transition rate can be obtained via marginalization over the posterior of the data state:

$$Q_t^{(r)}(x_t \to x_s) = \mathbb{E}_{q_{0|t}(x_0|x_t)}\left[Q_t^{(r)}(x_t \to x_s|x_0)\right],$$

where the posterior of the data state is given by $q_{0|t}(x_0 \mid x_t) = \frac{q_{t|0}(x_t|x_0)\, q_0(x_0)}{\mathbb{E}_{q_0(x_0)}[q_{t|0}(x_t|x_0)]}$, and $Q_t^{(r)}(x_t \to x_s|x_0)$ denotes the conditional transition rate of the reverse-time CTMC given the initial state x_0. After the calculation, we get the reverse transition rate:

$$Q_t^{(r)}(x_t \to x_s) = \frac{\alpha'_t}{1 - \alpha_t}\delta_m(x_t)p_{0|t}(x_0 = x_s|x_t).$$

The reverse CTMC could be simulated exactly using Gillespie's Algorithm [9, 10, 36], which samples the next jump time from an exponential distribution and selects a single dimension to update. As pointed out by [3], this is inefficient for large D. In practice, various methods [3, 23] adopt τ-Leaping [11, 36] as a parallel algorithm to update all transitions in a small time interval

4

simultaneously. The sampling process described in Section 3.1 is a first-order approximation with τ-Leaping in sampling of CTMC shown as follows.

The exact transition probability over a finite interval $[s, t]$ is given by integrating the transition rate. Since the CTMC is absorbing once unmasked, only transitions from m to a data token v are nontrivial. Denoting $x_t^d = m$, the transition probability satisfies the following Kolmogorov forward equation for any $\tau \in [s, t]$,

$$\frac{d}{d\tau} q_{\tau|s}(v|m) = Q_\tau^{(r)}(m \to v) = \frac{\alpha_\tau'}{1 - \alpha_\tau} p_{0|\tau}(v|m).$$

The transition probability thus can be written as

$$
\begin{aligned}
q_{s|t}(x_s^d = v \mid x_t^d = m) &= \int_s^t \frac{\alpha_\tau'}{1 - \alpha_\tau} p_{0|\tau}(v \mid m) \, d\tau \\
&\approx p_{0|t}(v \mid x_t, t) \int_s^t \frac{\alpha_\tau'}{1 - \alpha_\tau} \, d\tau \\
&= p_{0|t}(v \mid x_t, t) \log \frac{1 - \alpha_t}{1 - \alpha_s} \\
&\approx p_{0|t}(v \mid x_t, t) \frac{\alpha_s - \alpha_t}{1 - \alpha_t}.
\end{aligned}
$$

Therefore, we show that the sampling process of the masked diffusion model is a first-order approximation of τ-leaping sampling of the underlying reverse-time CTMC.

A characteristic property of τ-leaping is that approximate jumps across multiple dimensions are executed simultaneously. Due to the special design in the masked diffusion model, once a coordinate transitions from the mask token to a data token, it becomes absorbing. Consequently, errors introduced in early parallel jumps cannot be corrected in later steps. To mitigate the error accumulation caused by parallel generation, two main lines of work have been proposed.

The first line of work aims to restrict generation to tokens for which the model has high confidence. In particular, the Top-K strategy [43, 39, 34, 25, 35, 37] is widely adopted in modern masked diffusion models. Instead of treating each position as being generated with probability $(\alpha_s - \alpha_t)/(1 - \alpha_t)$, Top-K decoding selects the positions to be updated based on the highest predicted token probabilities. Top-K margin strategy [16] adopts the difference between the top two predicted probabilities. However, this type of method can only alleviate the error caused by parallel generation.

Another line of work endows masked diffusion models with self-correction capabilities [3, 18, 7, 4, 33, 28, 32, 42, 15, 27], allowing previously generated tokens to be revised in later steps.

3.3 Self-Correction of Masked Diffusion Models

[3, 18, 7, 4] adopt predictor-corrector for CTMC. [33, 28] generalize the reverse transition probability such that a generated token can be remasked according to the schedule σ_t, while [32] generalizes both forward and reverse processes for self-correction, which requires training a new masked diffusion model. [42, 15, 14] define per-token quality as the criterion for remasking, and finetune a pre-trained model for estimation per-token quality. We summarize existing self-correction methods according to whether they require finetuning an existing model or training a new model in Table 1.

4 Proposed Method

In this paper, we aim to investigate whether self-correction can be enabled in pre-trained multimodal masked diffusion models without any additional finetuning or retraining. To achieve training-free

Table 1: Categorization of self-correction for masked diffusion models.

Category	References
Training-free	[33]
Training-based or Finetuning-based	[32, 42, 15, 28, 14]

Table 2: Evaluation of text-to-image generation on the GenEval benchmark.

Method	Single	Two	Count	Color	Pos.	Attr.	Overall ↑
Lumina-DiMOO [a]	0.99	0.93	0.85	0.84	0.84	0.71	0.86
Lumina-DiMOO (ReMDM)	**1.00**	**0.94**	0.86	0.87	0.82	0.74	0.87
Lumina-DiMOO (Ours)	0.99	**0.94**	**0.88**	**0.93**	**0.87**	**0.79**	**0.90**

[a] We reevaluate the pre-trained Lumina-DiMOO on the GenEval benchmark for fair comparison. The generation step is 64.

self-correction, we need to revisit the existing training-free framework proposed by [33]. They generalize the reverse process of the masked diffusion model from Eq. (2) to

$$
q_\sigma\left(x_s|x_t, x\right) = \begin{cases} (1 - \sigma_t)\delta_{x_t}(x_s) + \sigma_t\delta_m(x_s), & x_t \neq m, \\ \frac{\alpha_s - (1-\sigma_t)\alpha_t}{1-\alpha_t}\delta_x(x_s) + \frac{1-\alpha_s-\sigma_t\alpha_t}{1-\alpha_t}\delta_m(x_s), & x_t = m. \end{cases}
$$

(4)

That is, once a token is generated, they still have the probability σ_t to be remasked. [33] propose three remasking schedules of σ_t, and find the confidence-based schedule with rescale achieves the best performance. In this schedule, the predicted token probabilities at the time of generation are used as the criterion for remasking. Tokens with lower predicted probabilities are more likely to be remasked. The criterion is reasonable in the sense that the predicted token probabilities $p_\theta^d(x_t)$ are only valid when $x_t^d = m$.

However, this criterion is limited in the sense that we need to evaluate the confidence of each position condition on current x_t rather than previous time step, i.e., the time that the token is generated. Therefore, recent works [42, 15] propose to reevaluate the predicted token probabilities for each generated position at each decoding step, i.e., $p(x_t^d) = p_\theta^d(x_t^d|x_t^{\backslash d})$ where $x_t^{\backslash d}$ denotes changing d-th dimension of x_t to m and $p_\theta^d(x_t^d|\cdot)$ denotes the predicted probability of token x_t^d by the masked diffusion model. [42] propose to train a hollow-transformer for efficient estimation, while [15] propose to finetune the pre-trained masked diffusion model to estimate the predicted token probability. A natural question raised: *Given the inductive bias of the pre-trained masked diffusion model, can we achieve training-free estimation for the predicted token probability?*

4.1 Rationality of Training-free Self-Correction

In this subsection, we use the real experimental results to verify the inductive bias of a pre-trained masked diffusion model. We utilize a state-of-art multimodal masked diffusion model [37]. We aim to investigate whether a pre-trained network is capable of identifying incorrect tokens during the generation process. Specifically, we sample multiple image-text prompt pairs $(z_0^{\text{image}}, z^{\text{prompt}})$ from the Flickr8k dataset [12]. Using a pre-trained language and image tokenizer, we project $(z_0^{\text{image}}, z^{\text{prompt}})$ into the embedding space (x_0, y). We then apply the noising process of masked diffusion models, $q_{t|0}(x_t|x_0)$, to obtain a noisy representation x_t, where $t \sim U(0, 1)$ and $U(0, 1)$ denotes the uniform distribution on the unit interval $(0, 1)$. Next, we randomly select unmasked
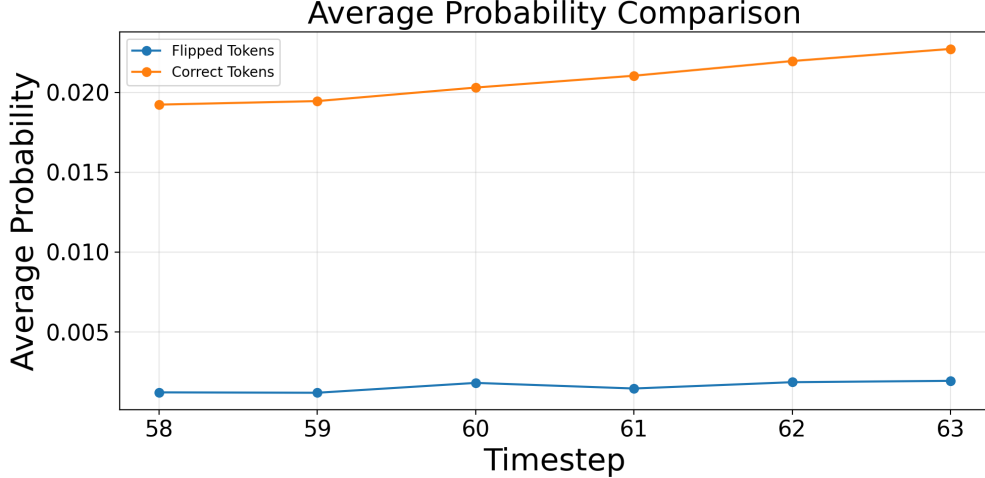
Figure 1: Average predicted probability of flipped tokens and correct tokens over 2000 samples. The x-axis denotes the time steps for generation (64 steps in total for text-to-image generation), while the y-axis denotes the average probability over all flipped positions and the correct position.

tokens and artificially flip their values. These manipulated tokens are fed into the pre-trained network, and we evaluate whether the model can detect the flipped tokens by examining the likelihood assigned to them.

We plot the average likelihood values of the original tokens and their flipped counterparts across different time steps in Figure 1. The results reveal that the pre-trained masked diffusion model will assign significantly higher predicted probabilities to the tokens that make more sense (corrected tokens) than the tokens that does not make sense (flipped tokens). This phenomenon reflected the inductive bias of the pre-trained model on the tokens that are already generated.

In Fig 2, we demonstrate that using the accumulated predicted probabilities could distinguish the wrong tokens well. For each time step after timestep 32, we found the average rank of accumulated probability for the wrong tokens is more and more greater than the average rank evaluated on a single step. Therefore, it inspires us to design a sliding window for self-correction.

4.2   In-depth Design of Self-Correction Criteria

Considering the inductive bias of the pre-trained masked diffusion model, we propose to use the predicted probability $p_\theta^d(x_t)$ for remasking, even when x_t has already been generated. This design allows the model to fully exploit its capabilities throughout the generation process.

Given this formulation, multiple design choices arise for the remasking strategy. We summarize several key aspects below:

- **Remasking rule**: Remasking can be performed either by applying a hard rule (e.g., a Top-K strategy analogous to that used during generation) or by sampling from a multinomial distribution defined by the predicted probability $p_\theta^d(x_t)$.

- **Temporal aggregation:** The remasking decision may depend on an accumulation of the predicted probabilities $p_\theta^d(x_t)$ across generation steps.

- **Top-K margin for remasking:** Whether Top-K margin strategy used in generation can also provide an improvement on remasking

7

Algorithm 1 Training-free Self-Correction for Masked Diffusion Models

---

Require: Pre-trained denoising network $\mathbf{x}_\theta$, number of steps T, noise schedule $\{\alpha_t\}$, remasking schedule $\{\sigma_t\}$, remasking score type Score $\in$ {Current, Cumulated}, remasking rule Rule $\in$ {Deterministic, Stochastic}

Ensure: Sample $\mathbf{x}_0$

  Initialize $\mathbf{x}_T \leftarrow \boldsymbol{m}$

  Initialize per-token score $\mathbf{S} \leftarrow \mathbf{0}$

  for $i = T, T-1, \ldots, 1$ **do**

    $t \leftarrow i/T, \quad s \leftarrow (i-1)/T$

    Retrieve $\alpha_t, \alpha_s, \sigma_t$

    (1) Standard reverse update

    $\hat{\mathbf{x}} \leftarrow \mathbf{x}_\theta(\mathbf{x}_t)$

    Sample $\mathbf{x}_s \sim q_\sigma(\mathbf{x}_s \mid \mathbf{x}_t, \hat{\mathbf{x}})$ {in Eq (4)}

    $\mathbf{x}_t \leftarrow \mathbf{x}_s$

    (2) Training-free likelihood evaluation

    for each position $d \in [\mathcal{D}]$ with $x_t^d \neq m$ **do**

      $\ell_t^d \leftarrow p_\theta^d\left(x_t^d \mid \mathbf{x}_t^{\backslash d}\right)$ {reevaluate likelihood}

      if Score $=$ Current **then**

        $S^d \leftarrow \ell_t^d$

      else

        $S^d \leftarrow S^d + \log(\ell_t^d)$ {cumulated likelihood}

      end if

    end for

    (3) Remasking step

    $K_t \leftarrow \lfloor \sigma_t \cdot D \rfloor$

    if Rule $=$ Deterministic **then**

      $\mathcal{R} \leftarrow \arg\min_{|\mathcal{R}|=K_t}\{S^d\}_{d:x_t^d \neq m}$ {remask K_t lowest-score tokens}

    else

      Sample $\mathcal{R}$ of size K_t with probability $\propto \exp(-S^d)$ over $\{d : x_t^d \neq m\}$

    end if

    for each $d \in \mathcal{R}$ **do**

      $x_t^d \leftarrow m$

    end for

  end for
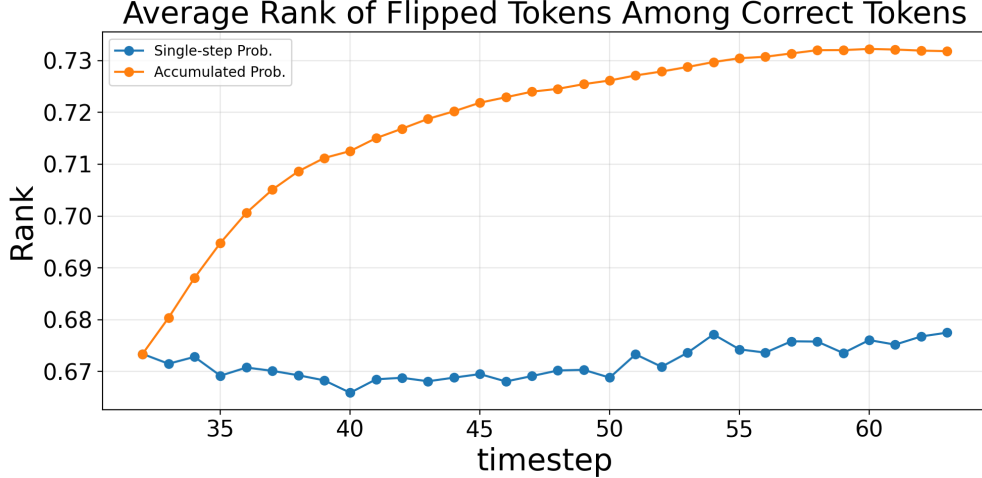
  return $\mathbf{x}_0$

Figure 2: The effectiveness of using accumulated predicted probability. The x-axis denotes the time steps for generation, while the y-axis denotes the average rank of the predicted probabilities of flipped tokens among correct tokens. The larger the rank is, the smaller the probability is.

- **Distributional uncertainty-based remasking**: Inspired by recent work [17], we consider utilizing richer information from the full vocabulary distribution of the predicted probability $p_\theta^d(x_t)$, rather than only relying on Top-K or Top-K margin. Specifically, we use either the KL divergence or the Wasserstein distance between the predicted distributions at consecutive time steps as a criterion to measure the model's confidence at each position. Positions whose generated tokens exhibit larger distributional changes are considered less confident. Accordingly, we remask the Top-K positions with the largest distances during sampling.

We summarize the overall procedure in Algorithm 1. In the following section, we carefully demonstrate the superiority of the proposed framework and provide a detailed comparison between various choices of remasking strategy.

5 Experiments

Experimental Setup   We conduct experiments using state-of-the-art multimodal masked diffusion models [37] and apply our self-correction strategy purely at inference time, without any additional fine-tuning or retraining. To assess the effectiveness of the proposed remasking strategy, we evaluate its performance on both text-to-image generation and multimodal understanding tasks. For text-to-image generation, we adopt the widely used GenEval benchmark [8], which focuses on complex compositional prompts involving object counting, spatial relations, and attribute binding. For multimodal understanding, we report results on MMBench (MMB) [21], SEED-Bench [19], and MMMU [41]. For all multimodal understanding benchmarks, we set the maximum generation length to 256 tokens per question. To obtain an automatic and consistent correctness decision, we adopt GPT-4o-mini as the verifier: given the model response and the benchmark-provided reference, the verifier outputs the judgment. Unless otherwise specified, we use the same backbone, tokenizers, and evaluation protocols as the original model and only change the sampling procedure by inserting the proposed remasking step.

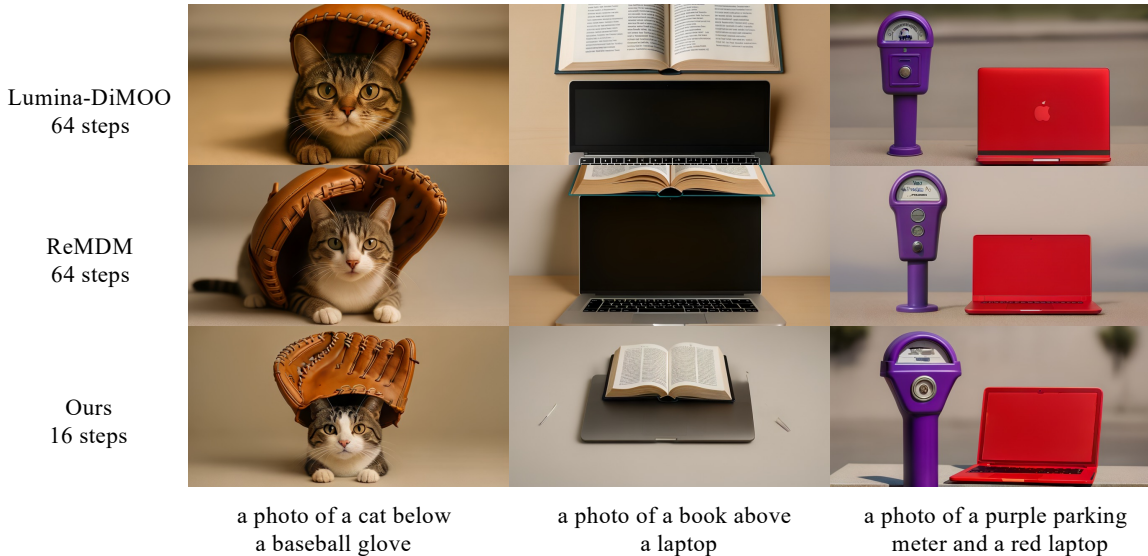|			
Lumina-DiMOO 64 steps			
ReMDM 64 steps			
Ours 16 steps			
	a photo of a cat below a baseball glove	a photo of a book above a laptop	a photo of a purple parking meter and a red laptop

Figure 3: Comparison of generated images of Lumina-DiMOO, ReMDM, and our methods. Our method achieves better quality with less sampling steps.

Table 3: Evaluation of multimodal understanding on several benchmarks.

Model	MMB ↑	SEED ↑	MMMU ↑
Lumina-DiMOO [a]	58.7	71.4	41.4
Lumina-DiMOO (ReMDM)	57.8	**74.3**	43.4
Lumina-DiMOO (Ours)	**60.7**	74.2	**44.0**

[a] We reevaluate the pre-trained Lumina-DiMOO on several multimodal understanding benchmarks for fair comparison. The generation step is 64 and the generated sequence length is 256.

5.1 Overall Effectiveness

We first evaluate the overall effectiveness of the proposed remasking strategy. As shown in Table 2, our method improves GenEval performance over vanilla Lumina-DiMOO as well as existing training-free self-correction approaches [33]. On multimodal understanding shown in Table 3 and Fig 5, we observe consistent gains when enabling our remasking-based self-correction at inference time. Overall, these results support our key claim: the inductive bias of pre-trained masked diffusion models can be exploited to identify and revise low-confidence tokens, leading to improved final generation quality without any additional training.

5.2 Ablation Study on Remasking Design Choices

We perform a comprehensive ablation study to analyze the impact of different remasking design choices described in Section 4.2. Specifically, we compare various remasking rules, temporal aggregation strategies, and uncertainty criteria to better understand their individual contributions in Table 4. In general, "Cumulated likelihood + deterministic remasking" achieves the best performance. It verifies the observation that accumulated predicted probabilities help in identifying the wrong

Table 4: Ablation study on various remasking strategy designs on the GenEval benchmark. "Deterministic remasking" stands for a greedy algorithm that remasks the positions where the predicted probabilities are the smallest K positions, while "stochastic remasking" represents the algorithm that samples the positions to be remasked according to the predicted probabilities. The smaller the predicted probability is, the more likely the position will be remasked.

Method	Single	Two	Count	Color	Pos.	Attr.	Overall ↑
Lumina-DiMOO	0.99	0.93	0.85	0.84	0.84	0.71	0.86
+ Current step likelihood + deterministic remasking	0.95	0.92	0.83	0.87	0.84	0.76	0.86
+ Current step likelihood + stochastic remasking	0.96	0.94	0.86	0.86	0.84	0.75	0.87
+ Cumulated likelihood + deterministic remasking	0.99	0.94	0.88	**0.93**	**0.87**	**0.79**	**0.90**
+ Cumulated likelihood + stochastic remasking	0.99	0.93	0.85	0.88	0.82	0.78	0.87
+ Top K margin	0.99	0.93	**0.90**	0.89	0.86	0.75	0.89
+ KL divergence	0.98	**0.95**	0.85	0.86	0.82	0.72	0.86
+ Wasserstein distance	**1.00**	0.93	0.84	0.88	0.84	0.66	0.86

Table 5: Evaluation of the proposed method on the GenEval benchmark based on MMaDA-8B-MixCoT [38].

Method	Single	Two	Count	Color	Pos.	Attr.	Overall ↑
MMaDA-8B-MixCoT [1] (Top K)	**0.93**	0.47	0.31	0.81	0.16	0.26	0.49
MMaDA-8B-MixCoT (Top K margin)	0.88	0.49	0.29	0.79	0.15	**0.28**	0.48
MMaDA-8B-MixCoT (ReMDM)	0.91	0.59	0.38	0.78	**0.20**	0.19	0.51
MMaDA-8B-MixCoT (Ours)	0.91	**0.67**	**0.39**	**0.84**	0.15	0.18	**0.52**

We reevaluate the pre-trained MMaDA-8B-MixCoT on GenEval benchmarks for fair comparison. The generation step is 64.

token shown in Fig 2. It also verifies the effectiveness of the widely used Top-K strategy. We conduct a careful hyperparameter search for the performance shown in Table 4. The hyperparameter "remasking schedule" σ_t is linked to the number of tokens K_t to be remasked by $K_t = \lfloor \sigma_t \cdot D \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor function. We determined the hyperparameter through the performance on the validation set.

5.3 Remasking for Efficient Sampling

We next study whether the proposed remasking mechanism can serve as a practical acceleration technique at inference time. We vary the number of reverse steps while keeping all other settings fixed, and report the GenEval overall score. As shown in Figure 4, with only 16 sampling steps, our method achieves comparable or better GenEval performance than the 64-step Lumina-DiMOO baseline. These results indicate that selectively revisiting low-confidence positions can mitigate early-step mistakes and enable four times speed up without sacrificing generation quality. The proposed method achieves more improvement against ReMDM, demonstrating the effectiveness of utilizing the inductive bias of the pre-trained model. We also provide some visualization of the generated images in Fig 3.
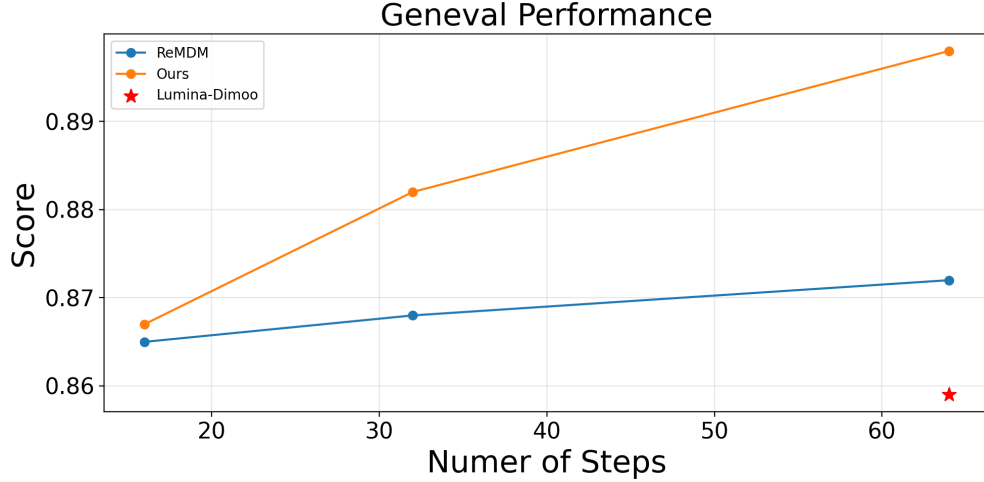
Figure 4: GenEval performance under different inference steps. The x-axis denotes the number of steps for generation, while the y-axis denotes the overall performance on the GenEval benchmark.

Which one is the correct caption of this image?

A: People in a horse drawn buggy on a city street.

B: A fire hydrant with a pair of eye stickers making a face on it.

C: a large food truck is parked on the side of the street

D: Neither one of these people had a good flight.

Lumina-DiMOO: Long response repeat the question several times.

ReMDM: Long response repeat the question several times.

Ours: To determine the correct caption for the image,
we need to carefully observe the details present in the image.
The image shows a fire hydrant with a pair of eye stickers on it.

Figure 5: Illustration of the performance on multimodal understanding.

5.4 Generalization Across Backbone Models

Finally, we evaluate whether our remasking strategy is tied to a specific backbone or can generalize across masked diffusion architectures. In Table 5, we evaluate the performance of the proposed method on MMaDA-8B-MixCoT [38] for text-to-image generation. We observe consistent improvements when enabling the same inference-time self-correction rule. This suggests that our method primarily leverages generic inductive biases of masked diffusion models, highlighting the robustness of our approach.

6 Conclusion

In this work, we provided an explanation for the error accumulation phenomenon caused by parallel and irreversible token updates. Then, we proposed a model-agnostic training-free self-correction framework that operates directly at inference time. Unlike prior approaches that rely on additional training, auxiliary evaluators, or misaligned likelihood, our method leverages the inductive biases of pre-trained masked diffusion networks for self-correction. Extensive experiments on text-to-image generation and multimodal understanding benchmarks demonstrate that our approach consistently improves generation quality, while also enabling faster sampling without sacrificing output quality. The observed gains generalize across different masked diffusion architectures, highlighting the robustness and practical applicability of the proposed method.

References

[1] Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *ICLR*, 2025.

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

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

[4] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *International Conference on Machine Learning*, 2024.

[5] Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and Bowen Zhou. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *ArXiv*, abs/2510.06303, 2025.

[6] Haodong Duan, Junming Yang, Yu Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiao wen Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, Dahua Lin, and Kai Chen. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024.

[7] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *NeurIPS*, 2024.

[8] Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *NeurIPS Track on Datasets and Benchmarks*, 2023.

[9] Daniel T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22:403–434, 1976.

[10] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81:2340–2361, 1977.

[11] Daniel T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, 115:1716–1733, 2001.

[12] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[13] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forr'e, and Max Welling. Argmax flows and multinomial diffusion: Towards non-autoregressive language models. *NeurIPS*, 2021.

[14] Zemin Huang, Yuhang Wang, Zhiyang Chen, and Guo-Jun Qi. Don't settle too early: Self-reflective remasking for diffusion language models. *ArXiv*, abs/2509.23653, 2025.

[15] Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z. Pan, Hyeji Kim, Sham M. Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction. *ArXiv*, abs/2510.01384, 2025.

[16] Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *ICML*, 2025.

[17] Seo Hyun Kim, Sunwoo Hong, Hojung Jung, Young Ju Park, and Se-Young Yun. Klass: Kl-guided fast inference in masked diffusion models. *NeurIPS*, 2025.

[18] José Lezama, Tim Salimans, Lu Jiang, Huiwen Chang, Jonathan Ho, and Irfan Essa. Discrete predictor-corrector diffusion models for image synthesis. In *International Conference on Learning Representations*, 2023.

[19] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal LLMs with generative comprehension. *CVPR*, 2024.

[20] Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, T. Jaakkola, and Rafael G'omez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising. *ICLR*, 2025.

[21] Yuanzhan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? *ArXiv*, abs/2307.06281, 2023.

[22] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2023.

[23] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024.

[24] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *NeurIPS*, 2025.

[25] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *International Conference on Learning Representations*, 2025.

[26] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *ICLR*, 2025.

[27] Yonghyun Park, Chieh-Hsin Lai, Satoshi Hayakawa, Yuhta Takida, and Yuki Mitsufuji. Jump your steps: Optimizing sampling schedule of discrete diffusion models. *ICLR*, 2025.

[28] Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. *ArXiv*, abs/2502.03540, 2025.

[29] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. *NeurIPS*, 2024.

[30] Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. *ICLR*, 2023.

[31] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Neural Information Processing Systems*, 2017.

[32] Dimitri von Rutte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Scholkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. *ICML*, 2025.

[33] Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *NeurIPS*, 2025.

[34] Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. *ICML*, 2024.

[35] Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforcement learning framework for diffusion large language models. 2026.

[36] Darren J. Wilkinson. Stochastic modelling for systems biology. 2006.

[37] Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Ke Wang, Yibin Wang, Jinbin Bai, Qian Yu, Dengyang Jiang, Yuandong Pu, Haoxing Chen, Le Zhuo, Junjun He, Gen Luo, Tian-Xin Li, Ming Hu, Jin Ye, Shenglong Ye, Bo Zhang, Chang Xu, Wenhai Wang, Hongsheng Li, Guangtao Zhai, Tianfan Xue, Bin Fu, Xiaohong Liu, Yu Qiao, and Yihao Liu. Lumina-dimoo: An omni diffusion large language model for multi-modal generation and understanding. *ArXiv*, abs/2510.06308, 2025.

[38] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025.

[39] Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *ICLR*, 2025.

[40] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *ArXiv*, abs/2508.15487, 2025.

[41] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9556–9567, 2023.

[42] Yixiu Zhao, Jiaxin Shi, Lester Mackey, and Scott W. Linderman. Informed correctors for discrete diffusion models. *NeurIPS*, 2025.

[43] Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *COLM*, 2024.