Why Mask Diffusion does not Work

WhaleTech.ai

info@whaletech.ai

Abstract

The main advantages of diffusion language models over autoregressive (AR) models lie in their ability to support parallel generation and bidirectional attention, enabling a more controllable generation process. In recent years, open-source mask diffusion language models have emerged, most of which are based on a variant known as absorbing diffusion. However, this paper demonstrates why mask diffusion faces inherent difficulties in achieving parallel generation and bidirectional attention. We also propose the most effective training and inference strategies for mask diffusion.

1 Introduction

Diffusion Language Models (DLMs) are rapidly emerging as a powerful and promising alternative to the dominant autoregressive (AR) paradigm in natural language processing. While AR models excel at a wide range of tasks, their sequential, token-by-token generation process imposes a major bottleneck on inference speed and computational efficiency. In contrast, DLMs generate tokens in parallel through an iterative denoising process. This core mechanism provides inherent advantages in reducing inference latency and naturally incorporating bidirectional context, making DLMs a compelling choice for addressing the trade-off between generation quality and speed.

To adapt diffusion for discrete language data, one of the primary directions has been the development of continuous-space models [1, 2, 3]. This approach first maps discrete tokens into a continuous embedding space where the denoising process is performed. The forward process gradually transforms the token embeddings into noise, and the reverse process learns to invert this corruption, starting from noise and progressively generating a clean embedding. After a denoised embedding is generated, a final rounding step maps it back to a discrete token from the vocabulary.

In recent years, discrete-space DLMs have gained increasing popularity as they define the diffusion process directly on the vocabulary of tokens, avoiding the need for a continuous embedding space during diffusion itself. In this paradigm, the forward process corrupts a sequence by applying a transition matrix at each step, which defines the probability of a token transitioning to any other token [4, 5, 6]. The model then learns to reverse these transitions by predicting the original token distribution given the corrupted sequence.

Masked Diffusion Models represent a modern and highly effective evolution of discrete diffusion language models, forming the foundation for several recent large-scale efforts such as DiffuLLaMA, LLaDA and Dream [7, 8, 9]. The core principle involves a forward process where tokens in a sequence are probabilistically replaced with a special `[MASK]` token [4]. During training, the model learns to predict the original content for these `[MASK]` positions, with the loss computed only over the `[MASK]` tokens. The generation process starts with a fully `[MASK]` sequence. In each iterative step, the model predicts several tokens. This iterative refinement continues until all `[MASK]` tokens are resolved.

In our work, we demonstrate that mask diffusion language models face inherent difficulties in achieving true parallel generation and bidirectional attention, due to the following reasons:

1. The model outputs the conditional marginal distribution for each `[MASK]` token, rather than the joint probability over all `[MASK]` tokens. This implies that parallel sampling cannot be theoretically guaranteed.

2. The distributions over [MASK] tokens distant from the unmasked positions are often smooth, and their most probable token IDs tend to be homogeneous. This essentially implies that, although many of the probabilities produced by the model are correct, they provide little useful information for sampling.
3. For mask diffusion language model, the most reliable and stable generation strategy is likely still the autoregressive (AR) approach, which makes it difficult to effectively leverage bidirectional attention during the generation process.

2 Preliminaries

2.1 Auto-regressive Modeling

Given a token sequence $x^{1:N} = (x^1, x^2, \dots, x^N)$, an autoregressive (AR) language model factorizes the joint probability into a product of conditional probabilities:

$$p(x^{1:N}) = \prod_{n=1}^{N} p(x^n \mid x^{<n}), \tag{1}$$

where $x^{<n} = (x^1, \dots, x^{n-1})$ denotes the prefix before position n. This left-to-right decomposition ensures that each token is generated conditioned on all previously observed tokens.

During training, the standard paradigm is to maximize the log-likelihood of the observed sequence, which is equivalent to minimizing the negative log-likelihood (NLL):

$$\mathcal{L}_{AR} = -\sum_{n=1}^{N} \log p_\theta(x^n \mid x^{<n}), \tag{2}$$

where θ denotes the model parameters. In practice, this loss reduces to the token-level cross-entropy between the predicted distribution and the ground-truth token at each position.

2.2 Mask Diffusion Modeling

Let $x_0 \in \Delta_K$ be a one-hot token (with the K-th category [MASK] denoted by m). Mask diffusion defines a forward process that interpolates between the clean x_0 and the [MASK] token via a monotone schedule $\alpha_t \in [0, 1]$ ($\alpha_0 = 1, \alpha_1 = 0, t \in [0, 1]$):

$$q(x_t \mid x_0) = \mathrm{Cat}\left(x_t;\ \alpha_t x_0 + (1 - \alpha_t)m\right). \tag{3}$$

For $s < t$, the transition is

$$q(x_t \mid x_s) = \mathrm{Cat}\left(x_t;\ \alpha_{t|s} x_s + (1 - \alpha_{t|s})m\right), \quad \alpha_{t|s} = \frac{\alpha_t}{\alpha_s}. \tag{4}$$

For absorbing diffusion ($\pi = m$), the posterior is

$$q(x_s \mid x_t, x_0) = \begin{cases} \mathrm{Cat}(x_s; x_t), & x_t \neq m, \\ \mathrm{Cat}\left(x_s;\ \frac{(1-\alpha_s)m + (\alpha_s - \alpha_t)x_0}{1-\alpha_t}\right), & x_t = m. \end{cases} \tag{5}$$

We approximate the reverse posterior by a denoiser $f_\theta(x_t, t) \in \Delta_K$:

$$p_\theta(x_s \mid x_t) = \begin{cases} \mathrm{Cat}(x_s; x_t), & x_t \neq m, \\ \mathrm{Cat}\left(x_s;\ \frac{(1-\alpha_s)m + (\alpha_s - \alpha_t)f_\theta(x_t)}{1-\alpha_t}\right), & x_t = m, \end{cases} \tag{6}$$

where $\langle f_\theta(x_t), m \rangle = 0$ [10].

$$\mathrm{KL}(q(x_s \mid x_t, x_0) \,\|\, p_\theta(x_s \mid x_t)) = \begin{cases} \frac{\alpha_s - \alpha_t}{1-\alpha_t} \mathrm{KL}(x_0 \,\|\, f_\theta(x_t)), & x_t = m, \\ 0, & x_t \neq m. \end{cases} \tag{7}$$

ELBO. The variational evidence lower bound is

$$\mathcal{E} = \mathbb{E}_q\big[\log p_\theta(x_0 \mid x_{t(1)})\big] - \sum_{i=2}^{T} \mathbb{E}_q\big[\mathrm{KL}\big(q(x_{s(i)} \mid x_{t(i)}, x_0) \parallel p_\theta(x_{s(i)} \mid x_{t(i)})\big)\big] - \mathbb{E}_q\big[\mathrm{KL}\big(q(x_{t(T)} \mid x_0) \parallel p(x_{t(T)})\big)\big] \,. \tag{8}$$

NELBO (training objective). Equivalently, the training loss as the negative ELBO is

$$\mathcal{L}_{\mathrm{NELBO}} = -\mathbb{E}_q\big[\log p_\theta(x_0 \mid x_{t(1)})\big] + \sum_{i=2}^{T} \mathbb{E}_q\big[\mathrm{KL}\big(q(x_{s(i)} \mid x_{t(i)}, x_0) \parallel p_\theta(x_{s(i)} \mid x_{t(i)})\big)\big] + \mathbb{E}_q\big[\mathrm{KL}\big(q(x_{t(T)} \mid x_0) \parallel p(x_{t(T)})\big)\big] \,. \tag{9}$$

Finally,

$$\mathcal{L} = \sum_{i=1}^{T} \left(-\frac{\alpha_{s(i)} - \alpha_{t(i)}}{1 - \alpha_{t(i)}} \, 1_{\{x_{t(i)} = m\}} \, x_0^\top \log f_\theta(x_{t(i)}) \right), \tag{10}$$

where $s(i+1) = t(i) = \frac{i}{T}$, x_0 is a one-hot vector.

In the continuous time limit where $T \to \infty$ [10], we set a small timestep $\Delta_t = t(i) - s(i) = \frac{1}{T} \in (0, 1)$. The sum over timesteps becomes an integral, and we have a limit

$$\alpha_t' = \frac{\alpha_{t(i)} - \alpha_{s(i)}}{t(i) - s(i)}, \tag{11}$$

and

$$\mathcal{L} = \int_0^1 \frac{\alpha_{t'}}{1 - \alpha_t} \, 1_{\{x_t = m\}} \, x_0^\top \log f_\theta(x_t) \, dt. \tag{12}$$

In practice, given a token sequence $x_0^{1:N} = (x_0^1, x_0^2, \ldots, x_0^N)$ and a time moment $t \in (0, 1]$:

$$\mathcal{L} = \frac{\alpha_{t'}}{1 - \alpha_t} \sum_{j=1}^{N} \left(1_{\{x_t^j = m\}} \, (x_0^j)^\top \log f_\theta(x_t^{1:N}) \right). \tag{13}$$

3 Challenges of Mask Diffusion: A Theoretical Perspective

To better understand the limitations of mask diffusion, this section provides a theoretical analysis of why it struggles to support parallel generation and bidirectional attention.

3.1 Theoretical Setup

For our theoretical analysis, we adopt the following idealized assumptions:

Assumption 1 (`[MASK]` token neutrality). The `[MASK]` token itself carries no intrinsic information (except location information which we don't predict) and can thus be omitted when formulating conditional probabilities. That means the prediction of a `[MASK]` token only relies on unmasked tokens.

Assumption 2 (Length independence). Although performance in practice depends on the length of given `[MASK]` sequence (for example, variations in the reserved length may substantially affect the length of the model's valid output), we abstract away from this factor and assume that diffusion model has already learned the complete data distribution under any given truncation length (equivalently, one may regard the model as reserving an infinitely long sequence of `[MASK]` tokens.). Consequently, the model's output is unaffected by the total sequence length: its prediction on a `[MASK]` token is determined solely by the set of currently unmasked tokens.

3.2 Marginal Distribution Analysis

In this section, we reveal why the model's output should be regarded as a marginal distribution and the distance-dependent features of this distribution.

3.2.1 Weighted Loss Function

We will explain why, from a theoretical perspective, the weights in the weighted loss function are not essential, although they can provide practical benefits in training.

The loss function (Equation 13) essentially measures the model's ability to reconstruct the original data from the noised input at any time step t. The only explicit dependence on t appears in the weighting term (e.g., $\frac{\alpha_{t'}}{1-\alpha_t}$). Theoretically, however, if the model is sufficiently trained (all loss equal to 0), these coefficients are not crucial. The role of the weights is mainly to encourage the model to focus more on cases with small t, while down-weighting the loss at large t, thereby stabilizing the training curve[10].

In fact, the weight comes from Equation 6; Since $f_\theta(x_t)$ has zero components at the positions corresponding to `[MASK]` tokens, we need weighted combination of $f_\theta(x_t)$ and m to approximate $q(x_s|x_t)$. That actually means the model f_θ only focuses on predicting the probabilities of unmasked tokens, one step back to the initial state $(t = 0)$.

Therefore, the model does not care about the exact timestep it is in during inference (unlike diffusion for image generation, timestep matters); at different timesteps, two sequences may well be in the same noise state, leading the model to produce identical outputs.

3.2.2 Model Output as Marginal Distributions

In discrete settings, it is necessary to manually define neighborhoods, as in concrete score matching[11]. Mask diffusion is essentially a single-token transition process, similar to SEDD[5]. Since noise is added independently to each token, The forward process can be regarded as many independent single-token transitions executed in parallel. This, in turn, implies that the reverse denoising process is also fundamentally a single-token transition problem: the model learns how to update one token at a time. However, when multiple tokens are updated simultaneously, there is no guarantee of their mutual coherence. We will demonstrate it in the following example.

A toy experiment Consider a toy dataset consisting of 100 sequences, each of length five tokens. All sequences share the same middle subsequence CD, and the final token E is unique across sequences (i.e., each sequence has a distinct value of E). The first two tokens pair has four cases: 34 sequences begin with AB, 21 with AB', 35 with $A'B$, and 10 with $A'B'$. Apply mask diffusion to this toy dataset and assume the model is sufficiently trained. If the input sequence is `[MASK][MASK]`CD`[MASK]`, what would the model's outputs on the `[MASK]` tokens be?

In the above experiment, the model fails to retain the joint probability of the first token pair. Instead, it only ouput $P(A|CD) = 0.55$ and $P(A'|CD) = 0.45$ at the first `[MASK]` position and $P(B|CD) = 0.69$ and $P(B'|CD) = 0.31$ at the second `[MASK]` position (Under Assumption 1, we omit the explicit form of the `[MASK]` tokens when computing conditional probabilities). If now you decide to predict the first two tokens at once, your choice will be AB while $A'B$ is the most probable pair.

This experiment highlights a problem. Although the model is exposed to many token combinations as supervision signals during training, it only outputs **the marginal distribution of each individual `[MASK]` token**, unlike image generation diffusion, which directly learns the gradient of the log joint probability.

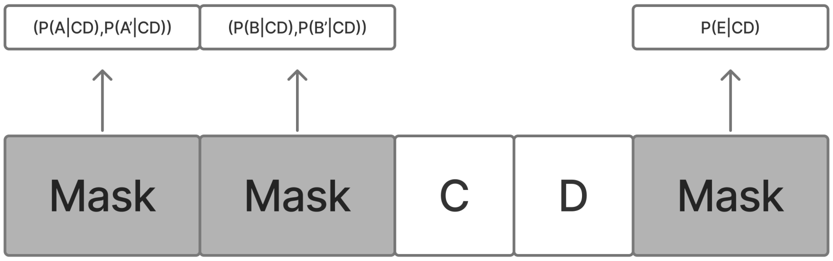

Figure 1: Model Output as Marginal Distributions

As a result, similar to autoregressive models, mask diffusion model cannot guarantee maximization of the joint probability. You may need auxiliary techniques, such as beam search, to find token combinations that maximize the joint probability.

3.2.3 Marginal Distributions as a Function of Distance

In general, the marginal distribution of a `[MASK]` token tends to be sharper when it is closer to unmasked tokens, whereas tokens farther away exhibit smoother distributions. This phenomenon arises because the conditional dependency on observed tokens weakens with distance, leading to greater uncertainty in the predicted probabilities.

This phenomenon can be described through a set of mathematical assumptions.

Assumption 3 . Under autoregressive (AR) conditions, the prediction of the next token is characterized by a Zipfian distribution. In formal mathematical expression, given all preceding unmasked tokens $T_1, T_2, ..., T_n$, the next token prediction $P(T_{n+1}|T_n T_{n-1}...T_1)$ is a Zipfian distrubution [1][12].

Assumption 4 . Given all preceding unmasked tokens $T_{n-1}, T_{n-2}, ...T_1$. Different choices of T_n will lead to noticeable differences in the prediction of next token T_{n+1}. In formal mathematical expression, the peak token IDs of $P(T_{n+1}|T_n T_{n-1}...T_1)$ will be different for any two distinct T_n. Equivalently, the peak token positions constitute a permutation matrix if T_{n+1} and T_n vary within the range of unmasked tokens.

Remark. Both assumptions are in fact rather strong, and they may not necessarily hold locally. However, in the sense of expectation, this trend exists.

Under these assumptions, we can demonstrate the existence of the aforementioned phenomenon.

Given a prompt (which may, for instance, be a user's query), the model reserves an infinitely long sequence of `[MASK]` tokens (Assumption 2); From left to right, we denote them as $T_1, T_2, ..., T_n, ...$, random variables initialized at the `[MASK]` tokens. As previously suggested, the model outputs marginal distributions of all `[MASK]` tokens.

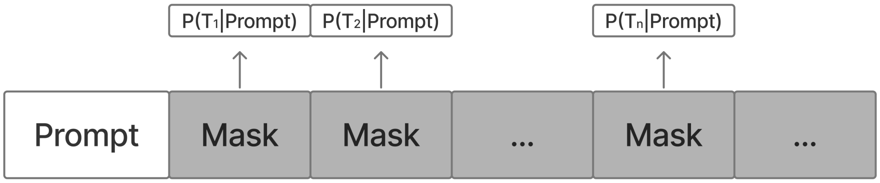

Figure 2: Initial state during inference

Then $P(T_n|T_{n-1}...T_1, prompt)$ is a Zipfian distribution for all $n \geq 1$. We assume that they share the same parameters N and s, thus leading to the same top1 prob ω_1 and top2 prob ω_2 [2].

We know

$$\begin{aligned}&P(T_n|prompt)\\ =&\sum_{T_1}\cdots\sum_{T_{n-2}}\sum_{T_{n-1}} P(T_n|T_{n-1}\cdots T_1, prompt)P(T_{n-1}|T_{n-2}\cdots T_1, prompt)\cdots P(T_1|prompt),\end{aligned} \tag{14}$$

where

$$\begin{aligned}&\sum_{T_{n-1}} P(T_n|T_{n-1}\cdots T_1, prompt)P(T_{n-1}|T_{n-2}\cdots T_1, prompt)\\ \leq&\omega_1 \text{Max} P(T_{n-1}|T_{n-2}\cdots T_1, prompt) + \omega_2(1 - \text{Max}P(T_{n-1}|T_{n-2}\cdots T_1, prompt)) \quad \text{(Assumption 4)}\\ =&(\omega_1 - \omega_2)\text{Max}P(T_{n-1}|T_{n-2}\cdots T_1, prompt)) + \omega_2.\end{aligned} \tag{15}$$

[1] $P(x = n) = \frac{n^{-s}}{\sum_{k=1}^{N} k^{-s}}, \quad n = 1, 2, \ldots, N$

[2] $\omega_1 = \frac{1}{\sum_{k=1}^{N} k^{-s}}, \omega_2 = \frac{2^{-s}}{\sum_{k=1}^{N} k^{-s}}$

In addition,

$$\sum_{T_{n-2}}\Big[(\omega_1-\omega_2)\mathrm{Max}P(T_{n-1}|T_{n-2}\cdots T_1,prompt)+\omega_2\Big]P(T_{n-2}|T_{n-3}\cdots T_1,prompt)$$
$$\le(\omega_1-\omega_2)\Big[(\omega_1-\omega_2)\mathrm{Max}P(T_{n-2}|T_{n-3}\cdots T_1,prompt)+\omega_2\Big]+\omega_2. \quad \text{(As Equation 15)} \tag{16}$$

Through iteration,

$$\begin{aligned}&P(T_n|prompt)\\ \le&(\omega_1-\omega_2)^{n-1}\mathrm{Max}P(T_1|prompt)+\omega_2\frac{1-(\omega_1-\omega_2)^{n-1}}{1-(\omega_1-\omega_2)}\\ =&(\omega_1-\omega_2)^{n-1}\omega_1+\omega_2\frac{1-(\omega_1-\omega_2)^{n-1}}{1-(\omega_1-\omega_2)} \quad \text{(Assumption 3)}\\ =&\frac{\omega_2}{1-(\omega_1-\omega_2)}+(\omega_1-\omega_2)^n\frac{1-\omega_1}{1-(\omega_1-\omega_2)}.\end{aligned} \tag{17}$$

As n increases, this upper bound converges to

$$\frac{\omega_2}{1-(\omega_1-\omega_2)}. \tag{18}$$

Although this is a coarse upper-bound estimate, it can still provide us with some intuition: the maximum probability will decay toward a lower bound which is small but not zero.

3.2.4 Experiment

We use LLaDA as the experiment model and set the maximum sequence length to 128. We consider four scenarios; In the first scenario, we design 100 simple questions; in the second, 100 moderately complex mathematics problems; in the third, 100 nonsensical questions; and in the fourth, 100 questions from the domain of humanities and the arts. We recorded the model's average maximum probability at all position of [MASK] tokens. Meanwhile, we set $s = 2.31$, $N = 130000$, under which you will observe the upper-bound curve obtained from our own estimation.

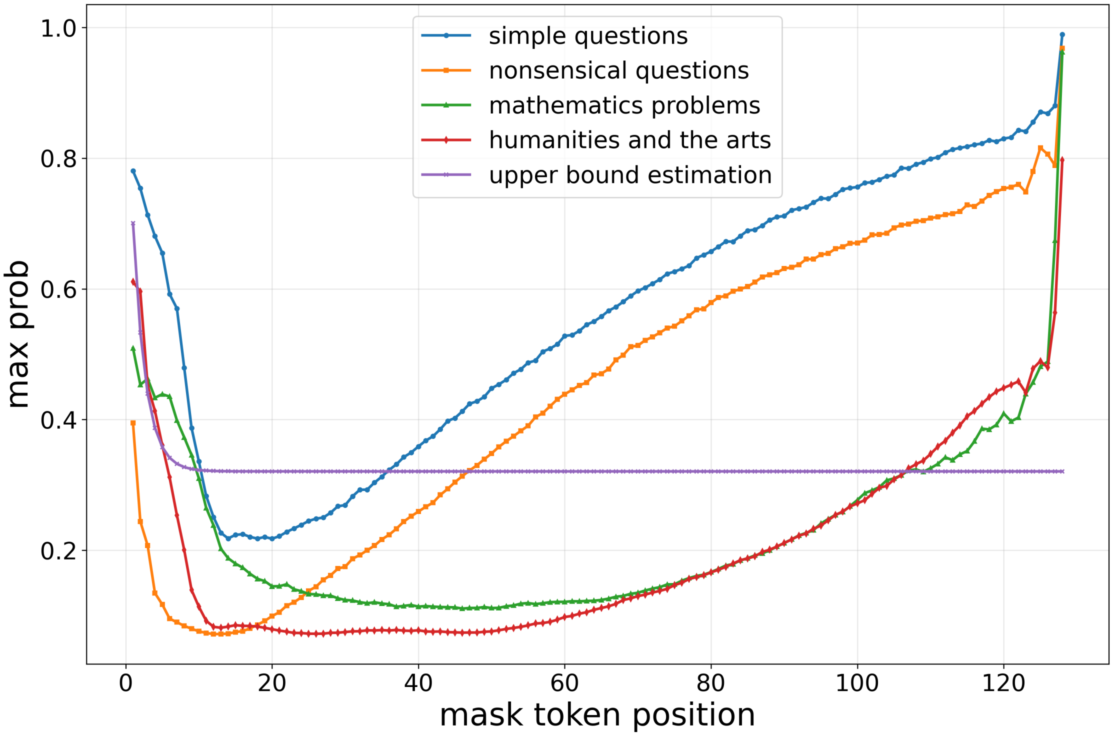

Figure 3: True average max prob and upper bound estimation

Regarding the experimental results, we have made several interesting observations.

Observation 1 (Increasing Maximum Probability). In practice, one may observe that the maximum probability component of [MASK] tokens near the end of a sentence tends to be relatively large. This phenomenon can often be attributed to the presence of special tokens such as end-of-text (or similar tokens indicating sequence termination). If such tokens occur frequently during training, they are more likely to dominate the probability mass during inference. However, this is not necessarily desirable, as the end-of-text token itself carries no substantive meaning, and its premature prediction effectively reduces the space available for meaningful tokens.

Observation 2 (Similarities and Differences). All the curves exhibit a rapid decline within the first 15 tokens, with the differences lying in their lower bounds and the positions at which they begin to rise. When the model exhibits a preference for shorter responses, the upward turning point of the curve emerges earlier. Moreover, when the task is both clear and simple, the lower bound of the probability distribution maybe higher.

Observation 3 (Ground Truth vs. Prediction). Our estimation follows the same trend as the actual behavior in the beginnig, however, the true average maximum probability usually converges to a smaller value (but still noticeably greater than zero). Our predicted curve does not rise afterwards, since our assumption does not account for length constraints or the special end-of-text token.

Overall, a smaller maximum probability in fact implies that the probability distribution becomes smoother and less favorable for sampling.

Homogenization of Distant Mask Predictions Some tokens are function words, which occur very frequently and can combine with many different tokens. In contrast, meaningful tokens occur less frequently and have more limited patterns of combination. This implies that function tokens are more likely to absorb probability mass transferred from the preceding token, which makes them more likely to dominate the probability distribution at positions farther away from unmasked tokens. This explains why, when all [MASK] tokens are predicted simultaneously, positions far from unmasked tokens often collapse to repeated function words such as "," or other high-frequency tokens.

Table 1: Homogenization of Distant Mask Predictions

I	plays	the	have	role	in	perceive	reality	.	perceive
the	.	.	the	and	the	and	and	and	and
and	,	and	and	,	,	and	and	,	,
,	,	,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,	,	,
,	,	,	,	,	,	,	,	endoftext	endoftext
endoftext	endoftext	endoftext	endoftext	endoftext	endoftext	endoftext	endoftext	endoftext	endoftext
endoftext	endoftext	endoftext	endoftext	endoftext	endoftext	endoftext	endoftext	endoftext	endoftext
endoftext	endoftext	endoftext	endoftext	endoftext	endoftext	.	eot_id		

Conjecture. At sufficiently large distances with an infinite length of [MASK] tokens, the distributions become almost identical. With a fixed given length, this near-identical behavior appears in the middle parts of the sequence.

The above characteristics are inevitable phenomena of mask diffusion when the model is sufficiently trained.

3.3 Parallel Sampling

In principle, parallel sampling is infeasible because the assumption of conditional independence does not hold, as also assumed in SEDD (though SEDD allows subsequent adjustments). Due to the distance-dependent behavior of the distribution, marginal conditional probabilities of the initial [MASK] tokens may still exhibit sharp features. Nevertheless, parallel sampling is likely to suffer from reduced joint probability, where unusual token combinations may appear together despite each token being individually probable.

Given a prompt, we introduce three metrics to evaluate the joint probability of parallel sampling. We assume that we only consider sampling the first few adjacent [MASK] tokens in parallel, as this is the most reliable choice among all parallel sampling strategies.

Metric 1 (The upper bound of joint probability). If you wish to sample the first n [MASK] tokens, and their corresponding maximum probabilities are $p_1, \cdots, p_n$, then the maximum joint probability of the token combination sampled according to these maximum probabilities is min$(p_1, \cdots, p_n)$.

Metric 2 (Independence assumption). If you assume that the selection of each token is mutually independent, then the joint probability is given by $\prod_{i=1}^{n} p_i$.

Metric 3 (The lower bound). If you decide to sample the first n [MASK] tokens according to their maximum probabilities, then in the worst case the joint probability is max$(0, \sum_{i=1}^{n} p_i - (n-1))^3$.

Although these metrics are rather coarse, they can be used to estimate the upper bound of parallel sampling.

Definition. *The PPL of a senquence $(x^1, x^2, \cdots, x^n)$ is defined as $P(x^1, x^2, \cdots, x^n)^{-\frac{1}{n}}$.*

Remark. Approximately speaking, perplexity (PPL) reflects the reciprocal of the average joint probability per token. A larger PPL indicates greater uncertainty for each token, which in turn implies a lower joint probability for the entire sequence.

For example, given a prompt and an infinite sequence of [MASK] tokens; With Equation 17 and Metric 2, assuming $s = 1.05$ and $N = 150000$, we obtain a table showing how approximate values of PPL vary with the number of parallel samples. If the upper bound of PPL is set to 17, then the upper bound on parallel sampling is 6.

Table 2: Approximate Values of PPL

Parallel Samples	PPL
1	9.6618
2	13.1995
3	14.8584
4	15.8570
5	16.4716
6	16.9408
7	17.3027
8	17.6066

The real situation is considerably more complex. Considering Homogenization of Distant Mask Predictions, Parallel sampling may lead to the same token being predicted at two adjacent positions, which in turn causes the joint probability to drop sharply.

Although estimating the upper bound of parallel prediction is challenging-since it depends on different semantic tasks, such a bound does exist and may be smaller than expected.

Remasking techniques Although mask diffusion also learns single-token transitions, it differs from SEDD in that the denoising process fixes the unmasked tokens. While certain remasking techniques exist[13], they can not achieve parallelism, as remasking multiple unmasked tokens simultaneously may cause substantial information loss, nor acceleration, since they require an additional prediction step.

A Random Initialization Strategy Inspired by SEDD One approach is to randomly initialize some [MASK] tokens and reveal others using these sparse signals, then remask the initialized tokens. Repeat many times to fill all [MASK] tokens. This is similar to a SEDD-inspired approximation, but it is inefficient(only the outputs of [MASK] tokens matter) and does not significantly improve speed. In addition, the ratio of randomly initialized tokens is important, as it sets the fraction of information the model fully relies on at each step.

3.4 Generation Order and Training Strategies

3.4.1 Generation Order

The order of generation is crucial, as it can lead to entirely different outputs. We argue that in mask diffusion-if only one token can be predicted at a time and previously unmasked tokens cannot be adjusted-the autoregressive generation order is optimal in general case.

[3] Bonferroni bound

This view is based on empirical intuition: when only one token can be predicted at a time, the best choice is to select the token with the highest marginal probability. From our previous analysis, we see that, in general, the predicted marginal distributions tend to decrease from left to right. In this case, the most reliable strategy is to choose the token with the largest marginal probability, which is typically the leftmost token. However, once the leftmost `[MASK]` token has been chosen, the next token often becomes the one with the largest marginal probability at that step. Thus, the generation process is often nearly autoregressive, with occasional instances of locally non-autoregressive behavior.

In this case, the model does not effectively exploit the advantages of bidirectional attention.

Reverse Order Generation If you sample according to confidence, which is also what we would typically do in practice. Occasionally, the generation process appears to proceed from the end backward. This arises from the influence of the end-of-text token, which is itself the most frequent token. When the sequence length is large, the model may assign the highest confidence to generating an end-of-text token at the final positions. Once one such token is generated, it can in turn increase the likelihood of generating another end-of-text token in the preceding position. Overall, this is not desirable, as it prematurely fixes the length of the sentence.

Random Generation Random-order generation often produces fluent text. However, it can only generate one token at a time; otherwise, it becomes difficult to ensure coherence among tokens. Moreover, it is difficult to guarantee that the joint probability of such sampling results is maximized.

3.4.2 Training Strategies

In mask diffusion, during training we randomly mask a relatively long sequence and attempt to reconstruct it. This requires our training to cover the $2^n - 1$ possible scenarios that a sentence may face, which is quite difficult to achieve. However, once we determine that inference is carried out in an almost autoregressive (AR) manner, most of the situations covered during training will not actually occur at inference time, thereby introducing substantial redundancy in training.

We introduce the optimal inference and training method, which is well aligned with the current generation approach.

Semi-AR Generation in Small Blocks During inference, we define a small block (e.g., of size 4 or 8) and generate within it using a diffusion process (or parallel sampling with the block size as the upper bound). Once the predictions for the block are completed, we proceed to the next small block, until the model outputs the end-of-text token. This essentially corresponds to semi-autoregression[3], but with smaller blocks; the reason is that `[MASK]` tokens far from the given condition are difficult to exploit, making large block sizes nearly ineffective.

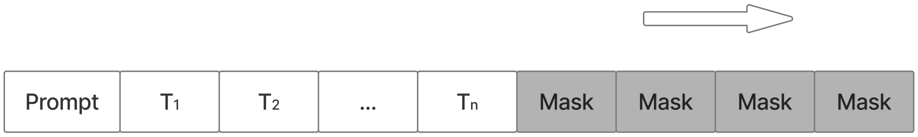

Figure 4: Semi-AR Generation (Block size=4)

Blockwise Reverse-Order Training During training, we replace a block of tokens with `[MASK]` tokens in a backward order, starting from the end of the sequence. The prediction of the tokens within this block serves as the supervision signal. Afterward, we remove the content of this block and move one block further toward the beginning, masking the corresponding tokens. This process continues until the remaining sequence is shorter than a block; if there are still residual tokens, we allocate a block from the leftmost side and let the model generate it unconditionally.

This training scheme better aligns with our generation approach and is more efficient, as each block only needs to be trained for $2^4 - 1$ cases (if block size = 4), which is easier to achieve.

4 Conclusion

In this work, we have provided a theoretical and empirical analysis of the limitations of mask diffusion language models: their predictions collapse to marginal distributions, parallel sampling lacks joint coherence, and the generation process is essentially autoregressive, which misaligns with the training procedure.

We also proposed a training and inference framework that better aligns with current mask diffusion language models, although it does not fundamentally enhance their ability for parallel generation or the effective use of bidirectional attention.

Our findings suggest that future work should investigate diffusion approaches that enable genuine parallel generation and effective bidirectional attention, while remaining computationally efficient in training.

For more information, please visit our website:

`https://whaletech.ai/`

WhaleTech.ai Team

Core Contributors:

Haocheng Sun

Cynthia Xin Wen

Edward Hong Wang

Research (alphabetically by first name):

B.Y. Wei

J.Y. Shi

T. Zhou

Z.X. Yang

Operation (alphabetically by first name):

J.F. Liu

X.W. Zhang

References

[1] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in neural information processing systems*, volume 35, pages 4328–4343, 2022.

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

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

[4] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rian Van Den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in neural information processing systems*, volume 34, pages 17981–17993, 2021.

[5] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024.

[6] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *35th Conference on Neural Information Processing Systems*, pages 1–12, 2021.

[7] Shitao Gong, Shailesh Agarwal, Yilun Zhang, Junkai Ye, Lin Zheng, Mike Li, Chen An, Peize Zhao, Wei Bi, and Jia Han. Scaling diffusion language models via adaptation from autoregressive models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[8] Siyin Nie, Fuzhao Zhu, Zhelun You, Xi Zhang, Jiacheng Ou, Jixuan Hu, Jizhou Zhou, Yixuan Lin, Ji-Rong Wen, and Chen Li. Large language diffusion models, 2025.

[9] Junkai Ye, Ziyu Xie, Lin Zheng, Jian Gao, Zikai Wu, Xing Jiang, Zhaocheng Li, and Lingpeng Kong. Dream 7b. `https://hkunlp.github.io/blog/2025/dream`, 2025.

[10] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, 2024.

[11] Chenlin Meng, Jiaming Song, Kristy Choi, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. In *Advances in Neural Information Processing Systems*, 2022.

[12] Steven T. Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5):1112–1130, 2014.

[13] Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.