Attention-Based Sampler for Diffusion Language Models

Yuyan Zhou[1] **Kai Syun Hou**[1] **Weiyu Chen**[2]* **James Kwok**[1]
[1] Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
[2] Hong Kong Institute of AI for Science, City University of Hong Kong

Abstract

Auto-regressive models (ARMs) have established a dominant paradigm in language modeling. However, their strictly sequential sampling paradigm imposes fundamental constraints on both inference efficiency and modeling flexibility. To address these limitations, diffusion-based large language models (dLLMs) have been proposed, offering the potential for parallel sampling and flexible language modeling. Despite these advantages, current dLLMs sampling strategies rely primarily on token level information, which fails to account for global sequence structure and often yields suboptimal results. In this paper, we study the sampling order selection problem from the perspective of log-likelihood maximization. We show that this problem is NP-hard and propose an optimal sampling-rank-based approximation that makes the objective computationally tractable. We further prove that the tractable objective is optimized by sampling tokens in descending order of their attention-matrix column sums. This finding provides a principled justification for attention-guided sampling and offers a theoretically grounded alternative to greedy search. We instantiate this theoretical insight in a new training-free sampling algorithm, termed Attn-Sampler, and further propose dynamic attention thresholding for practical acceleration. Extensive experiments across multiple benchmarks validate the effectiveness of our proposed method, demonstrating that it achieves superior generation quality while enhancing the sampling parallelism. Code is publicly available at: https://github.com/YuyanZhou1/Attn_Sampling.

1 Introduction

Auto-regressive models (ARMs) [Achiam et al., 2023, Liu et al., 2024, Dubey et al., 2024, Zhou et al., 2024, Wang et al., 2025] provide a principled and widely adopted framework for language modeling. It factorizes the joint distribution of a sequence into a product of conditional probabilities under a fixed left-to-right order. However, this approach requires strictly sequential, token-by-token sampling, imposing fundamental limitations on computational efficiency and modeling flexibility.

Recent works on diffusion large language models(dLLMs) [Nie et al., 2025, Ye et al., 2025, Zhu et al., 2025, Khanna et al., 2025, Wu et al., 2026a] relax this constraint by allowing permutation-based factorizations of the joint distribution. Instead of committing to a fixed left-to-right order, dLLMs define the sequence likelihood through flexible sampling permutations, enabling parallel prediction and order-agnostic generation. From a probabilistic perspective, this raises a central question:

How to choose the tokens' sampling order so as to maximize the likelihood of target sequence?

Existing sampling strategies for dLLMs address this question through token-level greedy search, such as confidence-based [Zhu et al., 2025], margin-based [Kim et al., 2025a] and entropy-

*Corresponding author: weiyu.chen@cityu.edu.hk

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

based sampling algorithm [Ye et al., 2025]. For parallel sampling, they typically rely on simple thresholding [Wu et al., 2026b, Kim et al., 2025b, Ben-Hamu et al., 2025] or top-k selection [Ye et al., 2025]. While effective in practice, these greedy algorithms lack a clear theoretical justification and connection to the log-likelihood maximization of the target sequence. Consequently, these local selection samplers often yield suboptimal sampling trajectories and fail to maximize the log-likelihood of the target sequence.

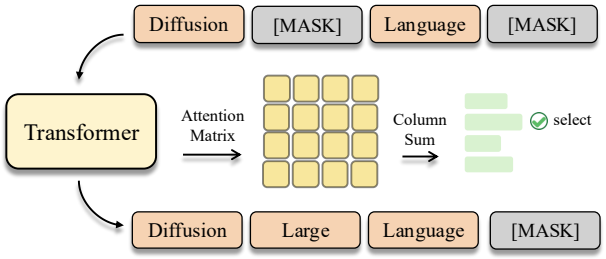

Figure 1: Overview of the Attn-Sampler algorithm. Our approach dynamically determines the sampling order for a masked sequence by leveraging the self-attention mechanism. We compute the column sums of the attention matrix as a proxy for token importance; tokens with higher cumulative attention scores are prioritized and decoded earlier in the sampling process.

In this work, we study the sampling order selection problem from a theoretical perspective. We formulate sampling order selection as an optimization problem that minimizes the gap between a practical permutation-based factorization likelihood and an ideal permutation-independent likelihood (where each token is predicted conditioned on all other tokens in the target sequence). Our main theoretical result shows that this gap is directly related to the token attention scores. However, such a direct optimization is computationally intractable. To alleviate this problem, we propose an optimal approximation to make the objective computationally tractable and theoretically demonstrate that sampling tokens in descending order of their attention matrix column sums can minimize the tractable objective. This provides a formal bridge between the structural properties of self-attention and the log-likelihood maximization problem.

We instantiate this theoretical insight into a novel, training-free sampling algorithm called Attn-Sampler (attention-based sampling). Unlike existing methods that focus on the output probability space, Attn-Sampler utilizes the attention matrix of the transformer model to dynamically determine the most informative tokens to decode at each step. Additionally, we also propose dynamic attention thresholding, an adaptive mechanism that regulates sampling parallelism based on sequence-wide dependencies, leading to higher parallelism.

Our contributions can be summarized as follows:

1. We show that the optimization problem of the sampling order selection is NP-Hard and propose an optimal approximation to make the optimization computationally tractable. We prove that sampling in descending order of attention column sums minimizes the tractable objective, thereby theoretically guaranteeing improved sampling quality.
2. We propose Attn-Sampler, a training-free algorithm that leverages this theoretical insight. Through an adaptive thresholding, it achieves efficient parallel sampling without compromising test accuracy.
3. We theoretically compare existing token-level samplers with our Attn-Sampler and identify the underlying causes of their differing performance in practice.
4. Extensive experiments demonstrate that Attn-Sampler consistently outperforms existing methods across multiple benchmarks in both accuracy and speed, establishing it as a robust new standard for dLLM inference.

2 Preliminary

2.1 Diffusion Large Language Models

Given a sequence $\boldsymbol{x} = (x_0, x_1, \ldots, x_n) \in \mathbb{R}^n$, where x_0 is the start (or context) token and each $x_i \in \mathcal{V}$ belongs to a finite vocabulary $\mathcal{V}$, in auto-regressive models (ARMs), the probability of $\boldsymbol{x}$ is

factorized based on a fixed left-to-right ordering:

$$p(\boldsymbol{x}) = p(x_0) \prod_{i=1}^{n} p\big(x_i \mid x_0, \ldots, x_{i-1}\big).$$

While computationally efficient, this fixed ordering imposes a rigid inductive bias that assumes the sequence structure is strictly unidirectional. However, in many settings, such as complex reasoning and code generation, the most informative tokens do not necessarily occur early in the sequence. Constraining the model to predict x_i solely based on its previous tokens will yield suboptimal learning and generation order, as it disregards dependency structures in the data.

To address this limitation, discrete large language models (dLLMs) [Sohl-Dickstein et al., 2015, Meng et al., 2022, Austin et al., 2021a, Nie et al., 2025] offer the potential for parallel sampling and flexible generation orders.

2.1.1 Basics

dLLMs are a class of latent variable models with a forward noising process and a learned reverse denoising process. Given data sample $\boldsymbol{x}$ (also denoted $\boldsymbol{x}^0$ in the following, where the superscript denotes the time step), the forward process progressively corrupts $\boldsymbol{x}^0$ to a sequence of latent variables $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^T$ (with each $\boldsymbol{x}^t \in \mathbb{R}^n$) by iteratively applying noise. The corresponding joint distribution is: $q(\boldsymbol{x}^{1:T}) = q(\boldsymbol{x}^0) \prod_{t=1}^{T} q(\boldsymbol{x}^t \mid \boldsymbol{x}^{t-1})$, where $q(\boldsymbol{x}^t \mid \boldsymbol{x}^s) = \text{Cat}(\boldsymbol{x}^t; \mathbf{Q}_t \boldsymbol{x}^s)$ is the categorical distribution and $\mathbf{Q}_t$ is the transition matrix at time step t. On the other hand, the backward process learns to gradually denoise the masked sequence back to the original data distribution by iteratively predicting masked tokens as t moves from T to 0, with the corresponding joint distribution: $p_{\boldsymbol{\theta}}(\boldsymbol{x}^{0:T}) := p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1} | \boldsymbol{x}^t)$. The model parameter can be optimized by minimizing the negative log-likelihood of the clean data $\boldsymbol{x}^0$. When the model uses the absorbing kernel, learning the denoising process leads to minimizing the Evidence Lower Bound (ELBO) of log likelihood [Nie et al., 2025]: $\mathcal{L}(\boldsymbol{\theta}) = -\mathbb{E}_{\boldsymbol{x}^0, \boldsymbol{\epsilon}, t} \sum_{i=1}^{N} \mathbf{1}_{[x_i^t = \texttt{MASK}]} \log p_{\boldsymbol{\theta}}(x_i^0 \mid \boldsymbol{x}^t)$, where x_i^0 is the ith element of $\boldsymbol{x}^0$.

2.1.2 Sampling Process in dLLMs

The sampling process involves iteratively unmasking tokens to reconstruct the original sequence. Early implementations use uniform sampling [Austin et al., 2021a], where the token to be unmasked is selected at random. However, this ignores the model's varying levels of certainty across different positions. To improve generation quality, research has shifted toward uncertainty-driven heuristics that utilize the model's intrinsic predictive signals to guide the unmasking sequence [Nie et al., 2025, Ye et al., 2025, Kim et al., 2025a, Zhou et al., 2026]. These strategies typically prioritize unmasking based on three metrics. (i) Confidence: Selecting tokens with the highest predicted probability for their most likely label [Nie et al., 2025]. (ii) Margin: Measuring the probability gap between the top two candidates [Kim et al., 2025a]. (iii) Entropy: Quantifying the overall predictive uncertainty across the entire distribution [Ye et al., 2025]. Recent advancements have further refined these heuristics by introducing adaptive samplers, which dynamically adjust the sampling process based on confidence [Wu et al., 2026b], entropy [Ben-Hamu et al., 2025], or KL-divergence [Kim et al., 2025b] to balance generation speed and sampling quality.

Each sampler specifies a generation order, represented by a permutation π. Existing dLLMs typically adopt block-wise [Arriola et al., 2025] or semi-autoregressive sampling [Nie et al., 2025], in which the permutation is defined at each block. Let π denote a permutation of the index set $\{1, \ldots, b\}$ within a block, such that token i is decoded at step $\pi(i)$. A permutation-based chain rule [Kim et al., 2025a, Chen et al., 2020] is used in dLLMs, which factorizes the joint distribution in a block with b tokens as:

$$p(\boldsymbol{x}) = p(x_0) \prod_{i=1}^{b} p\big(x_{\pi(i)} \mid \boldsymbol{x}_{\pi(<i)}\big). \tag{1}$$

Here, $\pi(< i) = \{j \in \{1, \ldots, b\} : \pi(j) < \pi(i)\}$ is the set of token indices decoded before token i, and $\boldsymbol{x}_{\pi(<i)}$ is the corresponding set of tokens. Thus, our goal is to find the permutation which

maximizes the log-likelihood:

$$\pi^* = \arg\max_\pi \log\left(p_{\boldsymbol{\theta}}(x_0)\prod_{i=1}^{b} p_{\boldsymbol{\theta}}(x_{\pi(i)}|\boldsymbol{x}_{\pi(<i)})\right). \tag{2}$$

2.2 One-Layer Transformer with Softmax Attention

In the theoretical analysis of transformers, the one-layer transformer with softmax attention has been commonly used [Huang et al., 2024, Li et al., 2024, Siyu et al., 2024, Vaswani et al., 2017]. Mathematically, given a sequence $\boldsymbol{x} = (x_0, x_1, \ldots, x_n)$ of discrete random variables drawn from distribution p_{data}, each token x_i is first mapped to a d-dimensional embedding $\boldsymbol{e}_i = \text{Embed}(x_i) \in \mathbb{R}^d$. A positional encoding $\boldsymbol{p}_i \in \mathbb{R}^d$ is added to form the input representation $\boldsymbol{h}_i = \boldsymbol{e}_i + \boldsymbol{p}_i$. Given $\{\boldsymbol{h}_i\}_{i=0}^n$, a single-head self-attention layer has three $d \times d$ projection matrices: $\boldsymbol{W}_K, \boldsymbol{W}_Q$, and $\boldsymbol{W}_V$, which calculates the query, key and value, respectively, as:

$$\boldsymbol{q}_i = \boldsymbol{W}_Q\boldsymbol{h}_i, \quad \boldsymbol{k}_i = \boldsymbol{W}_K\boldsymbol{h}_i, \quad \boldsymbol{v}_i = \boldsymbol{W}_V\boldsymbol{h}_i. \tag{3}$$

The output representation $\boldsymbol{z}_i$ is then computed as:

$$\boldsymbol{z}_i = \sum_j \boldsymbol{A}_{ij}\boldsymbol{v}_j, \tag{4}$$

where $\{\boldsymbol{A}_{ij} = \exp(\boldsymbol{q}_i^T\boldsymbol{k}_j/\sqrt{d})/\sum_j \exp(\boldsymbol{q}_i^T\boldsymbol{k}_j/\sqrt{d})\}$ are the attention weights. Finally, $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n]$ is decoded to token probabilities by softmax, as:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \text{Softmax}(\boldsymbol{W}\boldsymbol{Z} + \boldsymbol{b}\boldsymbol{e}^\top), \tag{5}$$

where $\boldsymbol{W} \in \mathbb{R}^{|\mathcal{V}|\times d}$, $\boldsymbol{Z} \in \mathbb{R}^{d\times n}$, $\boldsymbol{b} \in \mathbb{R}^{|\mathcal{V}|}$ and $\boldsymbol{e} \in \mathbb{R}^n$ denotes the all-ones vector. For notational simplicity, we use $\boldsymbol{\theta}$ to denote all the parameters in this model.

3 Proposed Method

In this section, we first formulate the determination of the sampling order in dLLMs as an optimization problem and show that it is NP-hard (Section 3.1). To make the optimization computationally tractable, we propose an optimal rank-only approximation and derive the optimality condition of the resulting tractable objective (Section 3.2). Based on this theoretical result, we propose Attn-Sampler, which determines the sampling order based on the attention matrix (Section 3.3).

3.1 Sampling Order Selection as Optimization

Let $S = \{1, \ldots, b\}$ be the index set of b tokens in a block. On predicting each x_i, the model can only access $\{x_j \mid \pi(j) < \pi(i)\}$. Here, we consider the ideal case where the model can access all the remaining tokens $\boldsymbol{x}_{S\setminus i} = [x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_b]$, with the corresponding likelihood $p_{\boldsymbol{\theta}}(x_0)\prod_{i=1}^b p_{\boldsymbol{\theta}}(x_i \mid \boldsymbol{x}_{S\setminus i})$. Obviously, π^* in (2) is the same as

$$\pi^* = \arg\min_\pi \texttt{PDG}_\pi(\boldsymbol{x}, \boldsymbol{\theta}), \tag{6}$$

where

$$\texttt{PDG}_\pi(\boldsymbol{x}, \boldsymbol{\theta}) = \left[\log\left(p_{\boldsymbol{\theta}}(x_0)\prod_{i=1}^{b} p_{\boldsymbol{\theta}}(x_i \mid \boldsymbol{x}_{S\setminus i})\right) - \log\left(p_{\boldsymbol{\theta}}(x_0)\prod_{i=1}^{b} p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{x}_{\pi(<i)})\right)\right]$$

will be called the *permutation dependency gap*. We interpret the first term on the RHS, $\log\left(p_\theta(x_0)\prod_{i=1}^b p_\theta(x_i \mid \boldsymbol{x}_{S\setminus i})\right)$, as the "ideal" (log) likelihood, where the model can access all non-target tokens when predicting each token. Thus, π^* in (6) finds the optimal permutation that minimizes the gap between this "ideal" likelihood and the permutation-based likelihood (the second term in RHS).

Let $\boldsymbol{v}_m$ be the value obtained from (3) on a mask token, and $\boldsymbol{v}_j$ the value for token x_j at position j. Define $B = \sup_{j\in S} \|\boldsymbol{v}_j - \boldsymbol{v}_m\|_2$. The following provides an upper bound for $\texttt{PDG}_\pi(\boldsymbol{x}, \boldsymbol{\theta})$. Proof is in Appendix A.1.

Proposition 3.1. *Under first-order approximation,* $\texttt{PDG}_\pi(\boldsymbol{x}, \boldsymbol{\theta}) \leq \sqrt{2}B\|\boldsymbol{W}\|_F \cdot \sum_{j=1}^{b} \sum_{i \in \pi(\leq j)} \boldsymbol{A}_{ij} := \mathrm{UB}(\boldsymbol{\theta}, \boldsymbol{x}, \pi)$*, where* $\boldsymbol{A}$ *is the attention matrix in (4), and* $\|\boldsymbol{W}\|_F$ *is the Frobenius norm of* $\boldsymbol{W}$ *in (5).*

In Proposition 3.1, the term $\sum_{i \in \pi(\leq j)} \boldsymbol{A}_{ij}$ can be equivalently written as $\sum_{i=1}^{b} \mathbf{1}(\pi(i) \leq \pi(j)) \boldsymbol{A}_{ij}$, where $\mathbf{1}(\cdot)$ is the indicator function. This reformulation shows that $\mathrm{UB}(\boldsymbol{\theta}, \boldsymbol{x}, \pi)$ has the same structure as a *weighted linear ordering problem*, which is NP-hard [Schiavinotto and Stützle, 2004]. Importantly, this expression also reveals that, for each token j, $\sum_{i \in \pi(\leq j)} \mathbf{A}_{ij}$ is not determined solely by its own *sampling rank* $\pi(j)$, but implicitly depends on the pairwise ordering structure $\mathbf{1}(\pi(i) \leq \pi(j))$ across token pairs. The computational hardness arises from this pairwise dependency: each cost term depends on the relative order of a token pair (i, j), making the resulting optimization computationally intractable.

3.2 Approximation for Tractable Optimization

To make the optimization problem tractable, we look for a function that only uses each token j's sampling rank $\pi(j)$ to approximate the pairwise term $\sum_{i \in \pi(\leq j)} \mathbf{A}_{ij}$ in expectation, i.e.

$$\min_{g} \mathbb{E}_{\tilde{\pi} | \tilde{\pi}(j) = \pi(j)} \left(\sum_{i \in \tilde{\pi}(\leq j)} \mathbf{A}_{ij} - g(\pi(j)) \right)^2, \tag{7}$$

where the expectation is over all permutations ($\tilde{\pi}$) subject to that token j has the given sampling rank $\pi(j)$.

Proposition 3.2. *For token* j*, the optimal* g *in (7) is:* $g^*(\pi(j)) = \frac{\pi(j)-1}{b-1} \sum_{i=1}^{b} \mathbf{1}(i \neq j) \mathbf{A}_{ij}$.

Proof is in Appendix A.2. Using the above to replace $\sum_{i \in \pi(\leq j)} \boldsymbol{A}_{ij}$ in $\mathrm{UB}(\boldsymbol{\theta}, \boldsymbol{x}, \pi)$, we have

$$\widehat{\mathrm{UB}}(\boldsymbol{\theta}, \boldsymbol{x}, \pi) \equiv \sqrt{2}B\|\boldsymbol{W}\|_F \cdot \sum_{j=1}^{b} \left(\frac{\pi(j) - 1}{b - 1} \sum_{i=1}^{b} \mathbf{1}(i \neq j) \boldsymbol{A}_{ij} \right).$$

Define the "total attention" of token j as $s_j = \sum_{i=1}^{b} \mathbf{1}(i \neq j) \boldsymbol{A}_{ij}$. The following Theorem shows that $\widehat{\mathrm{UB}}(\boldsymbol{\theta}, \boldsymbol{x}, \pi)$ is minimized when tokens are decoded in descending total attention order. Proof is in Appendix A.3.

Theorem 3.1. $\widehat{\mathrm{UB}}(\boldsymbol{\theta}, \boldsymbol{x}, \pi)$ *is minimized when the permutation* π *decodes tokens in descending total attention order.*

3.3 Algorithms

Sequential Sampling. We now present Attn-Sampler (Algorithm 1), a training-free sampling algorithm for dLLMs. Given a prompt $\boldsymbol{c}$ and block size b, we first append b mask tokens at the end of the prompt (step 3). The model performs a forward pass to output the probability matrix $\boldsymbol{p} \in \mathbb{R}^{b \times |\mathcal{V}|}$ and attention matrix $\boldsymbol{A}$ with its diagonal elements removed (step 5). The total attention scores in $\boldsymbol{s} \in \mathbb{R}^b$ are then computed (step 6). Based on Theorem 3.1, the masked tokens are decoded in descending total attention order such that $s_{\pi(1)} \geq s_{\pi(2)} \geq \cdots \geq s_{\pi(b)}$ (step 7). Finally, in steps 8-9, we identify and decode the token corresponding to the maximum total attention.

Parallel Sampling. Current parallel sampling methods [Ben-Hamu et al., 2025, Wu et al., 2026b] employ static strategies on token level information, such as fixed confidence/entropy thresholds, or top-k token selection, to determine which tokens to decode simultaneously. However, they neglect the global structure of the sequence, leading to suboptimal generation performance as will be seen in the experiments (Section 5.2).

To address this issue, we extend the sequential sampling method in Section 3.3 to parallel sampling (Algorithm 2). Specifically, after computation of the total attention scores as in Algorithm 1, we use a probability threshold τ to partition the set of mask tokens into two subsets (step 7): (i) a candidate

Algorithm 1 Attn-Sampler (Sequential).

Input: Prompt $\mathbf{c}$, diffusion language model $f_{\boldsymbol{\theta}}$, block size b.
repeat
  $\mathbf{c} \leftarrow \mathrm{Concat}(\mathbf{c}, b \times \texttt{MASK})$
  while $\texttt{MASK} \in \mathbf{c}$ **do**
    $\boldsymbol{p}, \boldsymbol{A} \leftarrow f_{\boldsymbol{\theta}}(\mathbf{c})$
    $s_i \leftarrow \sum_j \boldsymbol{A}_{ji}$
    $u \leftarrow \arg\max_{i \in \text{masked}} s_i$
    $\hat{\mathbf{x}} \leftarrow \arg\max(\boldsymbol{p}, \text{dim=1})$
    $\mathbf{c}_u \leftarrow \hat{\mathbf{x}}_u$
  end while
until $\langle \texttt{eos} \rangle \in \mathbf{c}$

Algorithm 2 Attn-Sampler (Parallel).

Input: Prompt $\mathbf{c}$, diffusion language model $f_{\boldsymbol{\theta}}$, block size b, threshold τ.
repeat
  $\mathbf{c} \leftarrow \mathrm{Concat}(\mathbf{c}, b \times \texttt{MASK})$
  while $\texttt{MASK} \in \mathbf{c}$ **do**
    $\boldsymbol{p}, \boldsymbol{A} \leftarrow f_{\boldsymbol{\theta}}(\mathbf{c})$
    $s_i \leftarrow \sum_j \boldsymbol{A}_{ji}$
    $\mathcal{I}_{\text{low}} \leftarrow \{i \mid \max p_i < \tau\}$
    $\gamma \leftarrow \max_{j \in \mathcal{I}_{\text{low}}} s_j$
    $\mathcal{U} \leftarrow \{i \mid s_i > \gamma\}$
    $\hat{\mathbf{x}} \leftarrow \arg\max(\boldsymbol{p}, \text{dim=1})$
    $\forall u \in \mathcal{U},\ \mathbf{c}_u \leftarrow \hat{\mathbf{x}}_u$
  end while
until $\langle \texttt{eos} \rangle \in \mathbf{c}$

set, whose tokens have probabilities greater than or equal to τ, and (ii) the non-candidate set with the remaining tokens. In the experiments, τ is set to 0.9. Among tokens in the non-candidate set, we identify the maximum total attention score as a "dynamic threshold" (step 8). Only candidate tokens whose total attention scores exceed this threshold are decoded in the current time step (steps 9-11). This ensures that we only parallelize the sampling of the most important and independent tokens [Wu et al., 2026b], thereby preserving generation quality.

4 Analysis of Different Samplers

In this section, we analyze the relationships between traditional samplers and the proposed attention-based sampler. In particular, confidence-based sampler selects the token that has the highest predicted probability [Nie et al., 2025], while entropy-based sampler selects the token that minimizes predictive uncertainty [Ye et al., 2025]. Specifically, let $\boldsymbol{p}_i$ be the predictive probability distribution for token i, and $p_i = \max(\boldsymbol{p}_i)$ the corresponding confidence (top probability). Confidence-based sampling selects $i^* = \arg\max(p_i)$, while entropy-based sampling selects token $i^* = \arg\min \mathcal{H}(\boldsymbol{p}_i)$, where $\mathcal{H}(\boldsymbol{p}_i)$ is the predictive entropy.

First, we show that the proposed Attn-Sampler select the same token as confidence-based sampler under certain conditions. First, we introduce a few symbols. On predicting each x_i, let $\boldsymbol{z}_i$ be the output from (4) when the model can only access $x_{\pi(<i)}$. Analogous to Section 3.1, we also consider the ideal case where the model can access all remaining tokens $\boldsymbol{x}_{S \setminus i}$, and let the corresponding output from (4) be $\boldsymbol{z}_i^*$. Define the difference $\boldsymbol{\epsilon}_i = \boldsymbol{z}_i - \boldsymbol{z}_i^*$.

Proposition 4.1. *Confidence-based sampling selects the same token as Attn-Sampler under a first-order Taylor approximation, when (i)* $\|\boldsymbol{\epsilon}_i\|_2 = h\left(\sum_{j \in \mathcal{M}} \boldsymbol{A}_{ij}\right)$*, where* h *is a monotonically increasing function, and* $\mathcal{M}$ *is the set of all mask tokens; (ii)* $\sum_j \boldsymbol{A}_{ji} = g\left(\sum_{j \notin \mathcal{M}} \boldsymbol{A}_{ij}\right)$*, where* g *is a monotonically increasing function; (iii) the direction of* $\boldsymbol{\epsilon}_i$ *aligns with the negative gradient of confidence w.r.t.* $\boldsymbol{z}$*; and (iv) In the ideal case, the predicted confidence is 1, and the gradient of the confidence score of token i with respect to* $\boldsymbol{z}_i^*$ *becomes approximately constant.*

Proof is in Appendix A.4. Next, the following Proposition shows that entropy-based sampling select the same token as confidence-based sampling.

Proposition 4.2. *Entropy-based sampling selects the same token* i^* *as that selected by confidence-based sampling if either (i)* $p_{i^*} = 1$*, and* $\forall i \neq i^*, p_i < 1$*, or (ii)* $p_{i^*} > p_i$ *for all* $i \neq i^*$*, and* $-\mathcal{H}(\boldsymbol{p}_{i^*}) > \log p_i,\ \forall i \neq i^*$*, where* $\mathcal{H}(\boldsymbol{p}_{i^*})$ *denotes the entropy of the predictive distribution at position* i^**.*

Proof is in Appendix A.5. In practice, the assumptions in Propositions 4.1 and 4.2 rarely hold. Consequently, as will be empirically demonstrated in Section 5, confidence-based and entropy-based sampling often yield inferior results.

Table 1: Testing accuracies of various sampling algorithms on different language models and evaluation tasks. The best results are in bold, and the second-best are underlined.

	Sampling method	GSM8K	Math	Humaneval	MBPP	average
Fast-dLLM v2 7B	KLASS	83.24	50.96	54.88	30.69	54.94
	EB-Sampler	58.15	25.18	22.56	15.08	30.24
	Fast-dLLM	82.11	49.22	54.27	30.69	54.07
	Margin Sampler	82.94	49.92	51.22	30.42	53.63
	Entropy Sampler	82.87	<u>51.92</u>	55.49	<u>35.98</u>	56.57
	Confidence Sampler	82.71	50.96	54.27	30.69	54.66
	Attn-Sampler (Parallel)	**84.23**	51.88	**58.54**	<u>35.98</u>	<u>57.66</u>
	Attn-Sampler (Sequential)	<u>84.00</u>	**52.50**	<u>57.93</u>	**36.24**	**57.67**
LLaDA-1.5 8B	KLASS	73.69	38.76	40.85	50.79	51.02
	EB-Sampler	49.51	26.40	42.07	53.18	42.79
	Fast-dLLM	<u>74.91</u>	39.62	40.24	48.68	50.86
	Margin Sampler	72.86	38.82	41.46	49.47	50.65
	Entropy Sampler	73.39	39.50	38.42	49.47	50.20
	Confidence Sampler	74.75	<u>39.64</u>	40.85	48.68	50.98
	Attn-Sampler (Parallel)	<u>74.91</u>	39.62	<u>42.68</u>	**53.97**	<u>52.80</u>
	Attn-Sampler (Sequential)	**74.98**	**39.66**	**43.29**	<u>53.44</u>	**52.84**
Fast-dLLM v2 1.5B	KLASS	61.64	32.22	37.20	27.78	39.71
	EB-Sampler	36.85	13.44	21.95	14.81	21.76
	Margin Sampler	61.49	31.52	37.19	25.13	38.83
	Entropy Sampler	62.55	32.02	37.19	30.16	40.48
	Confidence Sampler	61.56	<u>32.24</u>	37.19	27.78	39.69
	Attn-Sampler (Parallel)	<u>62.62</u>	31.02	**39.63**	**30.96**	<u>41.06</u>
	Attn-Sampler (Sequential)	**62.70**	**32.56**	**39.63**	<u>30.16</u>	**41.26**

5 Experiments

In this section, we validate the effectiveness of the proposed Attn-Sampler. We first compare its predictive accuracy and computational throughput against existing sampling methods, demonstrating that Attn-Samplerachieves state-of-the-art accuracy and a superior throughput-accuracy trade-off. Given that current transformers utilize multi-head attention across multiple layers, we further perform ablation studies on these components to demonstrate that aggregating information across all heads and layers maximizes performance.

5.1 Setup

Models and Hardware. Evaluation is performed on two representative dLLMs: Fast-dLLM v2 (with 1.5B and 7B parameters) [Wu et al., 2026a] and LLaDA-1.5 8B [Zhu et al., 2025]. All experiments are run on a single NVIDIA A6000 GPU.

Benchmark Datasets. We utilize two categories of benchmarks: (i) mathematical reasoning: GSM8K [Cobbe et al., 2021] and MATH [Saxton and Hill, 2019]; (ii) code generation: HumanEval [Chen et al., 2021] and MBPP [Austin et al., 2021b].

5.2 Test Accuracy

We compare the performance of Attn-Sampler against a suite of competitive sampling baselines for dLLMs: (i) top-1 confidence [Nie et al., 2025], (ii) top-1 margin [Kim et al., 2025a], (iii) top-1 negative entropy [Ye et al., 2025], (iv) Fast-dLLM [Wu et al., 2026b], (v) EB-Sampler [Ben-Hamu et al., 2025] and (vi) KLASS Sampler [Kim et al., 2025b].

Table 1 shows the accuracies of various samplers. As can be seen, both sequential sampling and parallel sampling of the proposed Attn-Sampler consistently achieve state-of-the-art results across all the model architectures and benchmarks evaluated. With the Fast-dLLM v2 7B, Attn-Sampler sequential sampling outperforms the top confidence sampler by 3.01% on the average metric. Compared with the strongest baseline (entropy sampler), Attn-Sampler outperforms by 1.1% on average, with a notable +2.44% accuracy gain on HumanEval. Attn-Sampler parallel sampling maintains accuracy levels comparable to those of sequential sampling, and even outperforming sequential sampling on GSM8K and Humaneval datasets. With the different LLaDA-1.5 8B model, Attn-Sampler maintains

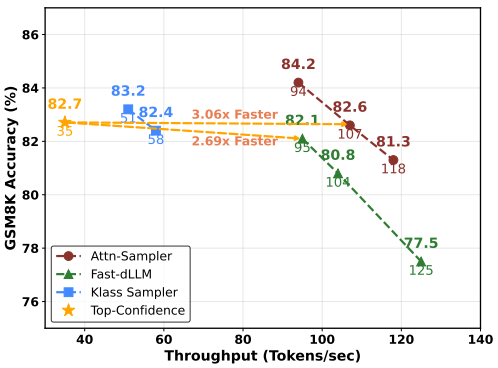

Figure 2: Comparison of throughput vs. accuracy across various parallel sampling methods. Our Attn-Sampler defines a superior Pareto front compared to existing state-of-the-art parallel samplers.

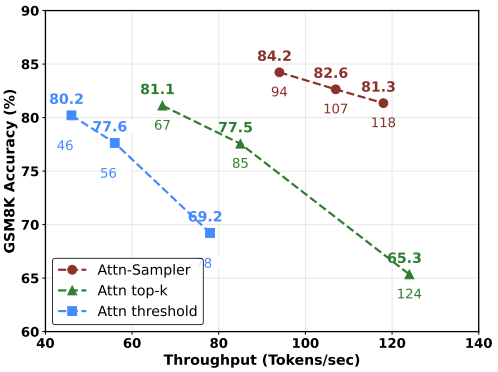

Figure 3: Ablation study of attention sampling strategies. Our adaptive approach outperforms both top-k and static thresholding, achieving a more efficient trade-off between speed and performance.

the highest average accuracy (sequential: 52.84%; parallel: 52.80%), demonstrating robustness across different diffusion scales.

The advantages of Attn-Sampler remain evident at the 1.5B parameter scale. With the Fast-dLLM v2 1.5B, Attn-Sampler achieves a 41.26% average score, surpassing the KLASS and Fast-dLLM baselines by over 1.5 percentage points. This suggests that Attn-Sampler effectively captures the underlying distribution of the diffusion process, even when the base model capacity is limited.

5.3 Inference Speed Comparison

Using the Fast-dLLM v2 model on GSM8K, we evaluate the trade-off between generation throughput (measured in tokens per second (TPS)) and task accuracy of various parallel sampling algorithms (Attn-Sampler, Fast-dLLM, and KLASS). For Attn-Sampler, we vary the threshold τ to $0.9, 0.8$ and 0.7; for Fast-dLLM, we vary the confidence threshold to $0.9, 0.8$ and 0.7. for KLASS, we vary the KL-divergence thresholds to 0.001 and 0.01.

Figure 2 shows the comparison of throughput-accuracy curves. For comparison, we also show the result of top-confidence sampling. As can be seen, Attn-Sampler consistently shows a better Pareto front compared to the baselines. While top-confidence sampling achieves an accuracy of 82.7% with a throughput of 35 TPS, the Fast-dLLM sampler provides a $2.69\times$ speedup (95 TPS) but at a slight drop in accuracy (82.1). In contrast, at the same throughput of 95 TPS, Attn-Sampler achieves a significantly higher accuracy of 84.2%. Furthermore, Attn-Sampler can be configured for even higher efficiency, reaching 107 TPS ($3.06\times$ acceleration) while maintaining 82.6% accuracy, effectively matching the precision of the confidence-based baseline at triple the speed. While the KLASS sampler reaches a competitive accuracy of 83.2%, its throughput is limited to 51 TPS, making it substantially less efficient than our proposed method. These results demonstrate that Attn-Sampler effectively bridges the gap between high-quality reasoning and high-throughput inference.

5.4 Ablation Studies

Dynamic Attention Thresholding In this section, we ablate and validate the effectiveness of our proposed parallel attention sampler. Static thresholding [Wu et al., 2026b] and top-k [Ye et al., 2025] parallelization strategies can improve the throughput of confidence and entropy sampler. However, we show that these strategies are insufficient for the requirements of Attn-Sampler. To evaluate the effectiveness of our proposed dynamic attention thresholding mechanism, we conduct an ablation study using the Fast-dLLM v2 model on the GSM8K dataset. We compare our approach against the two standard strategies: (i) Top-k selection: Decode a fixed number of tokens corresponding to the highest attention scores, where k is set to $2, 3, 4$. (ii) Static thresholding: Masks all attention weights falling below a predefined constant attention threshold, with threshold set as $1.0, 0.9, 0.8$.

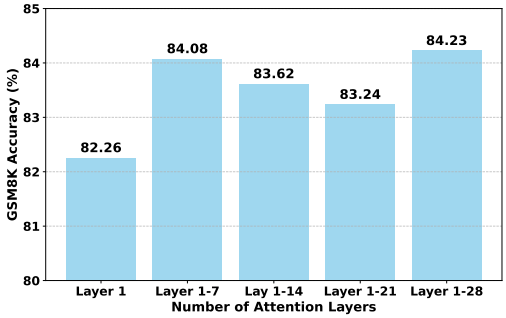

Figure 4: Ablation study of test accuracy using different attention layers.

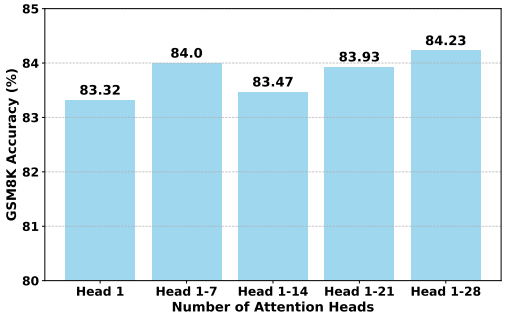

Figure 5: Ablation study of test accuracy using different attention heads.

Figure 3 shows the comparison of different attention based parallel sampling strategies. As can be seen, Attn-Sampler exhibits a significantly more robust Pareto front than either baseline. While both the top-k and static thresholding variants can increase throughput, they suffer from a sharp decline in accuracy as parallelism becomes higher. For example, increasing the throughput of the top-k baseline from 67 TPS to 124 TPS results in an accuracy drop of approximately 15.77% (from 81.12 to 65.35). In contrast, Attn-Sampler maintains a high reasoning performance even at elevated speeds. At 118 TPS, our method still achieves 81.35% accuracy, outperforming the top-k method's accuracy at nearly half the speed (67 TPS). The results suggest that static or fixed-count pruning fails to account for the varying importance of tokens across different diffusion steps. By adaptively scaling the threshold, Attn-Sampler effectively preserves the critical semantic information required for complex tasks like GSM8K while successfully eliminating redundant computations.

Attention Layers and Heads To evaluate the contribution of the internal transformer components to the sampling process, we conduct ablation studies by cumulatively increasing the number of active attention layers and heads. Attention Layers: We first compare the performance of utilizing only the initial layers against different layer subsets up to the full model depth. Figure 4 shows the test accuracies using obtained. As can be seen, relying solely on the first attention layer yields the lowest performance (82.26%), whereas performance improves significantly when extending the scope to the first 7 layers (84.08%) and attain highest when using the mean of all the layers. Attention Heads: A similar cumulative study is conducted for attention heads within each layer. Figure 5 shows the ablation study of test accuracy using different attention heads. As can be seen, using only the first head results in a baseline of 83.32%, with performance scaling to 84.0% as the first 7 heads are included. For both layers and heads, the model achieves peak performance (84.23% on GSM8K) only when the full architectural width and depth, specifically the mean of all 28 layers and heads are utilized. These results suggest that while early components capture fundamental features, the high-level semantic information and specialized representations distributed across the entire network are critical for maximizing model performance.

6 Conclusion

In this work, we introduced a principled approach to sampling order selection for dLLMs by framing it as a log-likelihood maximization problem. This problem can be formulated as minimizing a "permutation dependency gap". We show that this optimization problem is NP-hard, and propose an optimal sampling-rank-based approximation that makes the objective computationally tractable. We further prove that the resulting tractable objective is optimized by sampling tokens in descending order of their attention-matrix column sums. Based on these findings, we propose Attn-Sampler, a training-free algorithm that uses attention scores to guide sequence generation. By incorporating dynamic thresholding, Attn-Sampler achieves superior generation quality and enhanced parallelism. We further theoretically compare existing token-level samplers with our Attn-Sampler and identify the underlying causes of their differing performance in practice. Our results demonstrate that Attn-Sampler consistently outperforms existing methods, establishing a new, theoretically grounded standard for efficient dLLM sampling.

7 Acknowledgements and Disclosures

This research was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region (Grants 16202523 and 16213126).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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. *arXiv preprint arXiv:2503.09573*, 2025.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 2021a.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021b.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. 2025.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. In *ICML*, 2024.

Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

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

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

Zihao Li, Yuan Cao, Cheng Gao, Yihan He, Han Liu, Jason Klusowski, Jianqing Fan, and Mengdi Wang. One-layer transformer provably learns one-nearest neighbor in context. *NeurIPS*, 2024.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. *NeurIPS*, 2022.

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 *NeurIPS*, 2025.

Grefenstette Saxton and Kohli Hill. Analysing mathematical reasoning abilities of neural models. *arXiv:1904.01557*, 2019.

Tommaso Schiavinotto and Thomas Stützle. The linear ordering problem: Instances, search space analysis and algorithms. *Journal of Mathematical Modelling and Algorithms*, 2004.

Chen Siyu, Sheen Heejune, Wang Tianhao, and Yang Zhuoran. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. In *COLT*, pages 4573–4573, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

Bingning Wang, Haizhou Zhao, Huozhi Zhou, Liang Song, Mingyu Xu, Wei Cheng, Xiangrong Zeng, Yupeng Zhang, Yuqi Huo, Zecheng Wang, et al. Baichuan-m1: Pushing the medical capability of large language models. *arXiv preprint arXiv:2502.12671*, 2025.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. Fast-dllm v2: Efficient block-diffusion llm. In *ICLR*, 2026a.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. In *ICLR*, 2026b.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Yuyan Zhou, Liang Song, Bingning Wang, and Weipeng Chen. Metagpt: Merging large language models using model exclusive task arithmetic. In *EMNLP*, 2024.

Yuyan Zhou, Weiyu Chen, and James Kwok. Diffusion with truncated blocks: Fast and high-quality text generation using truncated block generation. In *Findings of ACL*, 2026.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. In *NeurIPS*, 2025.

A Proofs

A.1 Proof of Proposition 3.1

First, rewrite the optimal permutation π^* in (6) as:

$$\pi^* = \arg\min_{\pi} \sum_{i=1}^{b} \left[-\log p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{x}_{\pi(<i)}) + \log p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{x}_{S\setminus\pi(i)}) \right].$$

For the token at position $\pi(i)$, let $\boldsymbol{z}_{\pi(i)}$ be the representation obtained in (4), and $\boldsymbol{z}^*_{\pi(i)}$ be the representation in the ideal case where the model can access all remaining tokens. From (5),

$$\log p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{x}_{\pi(<i)}) = \log \frac{\exp(\boldsymbol{W}\boldsymbol{z}_{\pi(i)} + \boldsymbol{b})_{x_{\pi(i)}}}{\sum_k \exp(\boldsymbol{W}\boldsymbol{z}_{\pi(i)} + \boldsymbol{b})_k} := \log p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{z}_{\pi(i)}).$$

Let $\boldsymbol{u}_{\pi(i)} \equiv \boldsymbol{W}\boldsymbol{z}_{\pi(i)} + \boldsymbol{b}$. The gradient of $\log p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{z}_{\pi(i)})$ w.r.t. $\boldsymbol{u}_{\pi(i)}$ is:

$$\nabla_{\boldsymbol{u}_{\pi(i)}} \log p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{z}_{\pi(i)}) = -\boldsymbol{p}_{\pi(i)} + \boldsymbol{y}_{x_{\pi(i)}},$$

where $\boldsymbol{p}_{\pi(i)}$ is the predictive probability vector for token $x_{\pi(i)}$ and $\boldsymbol{y}_{x_{\pi(i)}} \in \mathbb{R}^{|\mathcal{V}|}$ denotes the one-hot vector for token $x_{\pi(i)}$. Note that

$$\left\| \nabla_{\boldsymbol{u}_{\pi(i)}} \log p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{z}_{\pi(i)}) \right\|_2^2 = (1 - p_{x_{\pi(i)}})^2 + \sum_{j \neq \pi(i)} p_{x_j}^2 \leq 2,$$

i.e., its gradient norm is bounded and the Lipchitz constant of the log softmax function is $\sqrt{2}$: Given that the Lipchitz constant is $\sqrt{2}$ and $\boldsymbol{u}_{\pi(i)}$ is linear in $\boldsymbol{z}_{\pi}(i)$ with weight $\boldsymbol{W}$, the difference in log-probabilities is bounded by:

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{x}_{S\setminus\pi(i)}) - \log p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{x}_{\pi(<i)}) &= \log p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{z}^*_{\pi(i)}) - \log p_{\boldsymbol{\theta}}(x_{\pi(i)} \mid \boldsymbol{z}_{\pi(i)}) \\
&\leq \sqrt{2} \|\boldsymbol{W}\boldsymbol{z}_{\pi(i)} - \boldsymbol{W}\boldsymbol{z}^*_{\pi(i)}\|_2 \\
&\leq \sqrt{2} \cdot \|\boldsymbol{W}\|_F \cdot \|\boldsymbol{z}_{\pi(i)} - \boldsymbol{z}^*_{\pi(i)}\|_2.
\end{aligned}$$

Next, we analyze the difference in the latent representations. In the Transformer architecture, the hidden state is a weighted sum of value projections $\boldsymbol{v}_j$ mediated by attention weights $\boldsymbol{A}_{ij}$. Let $\mathcal{M}$ be the set of indices for masked tokens. We can express $\boldsymbol{z}_{\pi(i)}$ and $\boldsymbol{z}^*_{\pi(i)}$ as:

$$\boldsymbol{z}_{\pi(i)} = \sum_{j \notin \mathcal{M}} \boldsymbol{A}_{\pi(i)j} \boldsymbol{v}_j + \sum_{j \in \mathcal{M}} \boldsymbol{A}_{\pi(i)j} \boldsymbol{v}_m,$$

$$\boldsymbol{z}^*_{\pi(i)} = \sum_{j \notin \mathcal{M}} \boldsymbol{A}_{\pi(i)j} \boldsymbol{v}_j + \sum_{j \in \mathcal{M}} \boldsymbol{A}_{\pi(i)j} \boldsymbol{v}_j.$$

By subtracting these expressions, the terms corresponding to non-masked tokens cancel out:

$$\|\boldsymbol{z}_{\pi(i)} - \boldsymbol{z}^*_{\pi(i)}\|_2 = \left\| \sum_{j \in \mathcal{M}} \boldsymbol{A}_{\pi(i)j} (\boldsymbol{v}_m - \boldsymbol{v}_j) \right\|_2.$$

Applying the Triangle Inequality and substituting the supremum of the value discrepancy $B = \sup_{j,m} \|\boldsymbol{v}_j - \boldsymbol{v}_m\|_2$, we obtain:

$$\|\boldsymbol{z}_{\pi(i)} - \boldsymbol{z}^*_{\pi(i)}\|_2 \leq \sum_{j \in \mathcal{M}} \boldsymbol{A}_{\pi(i)j} \|\boldsymbol{v}_m - \boldsymbol{v}_j\|_2 \leq B \sum_{j \in \mathcal{M}} \boldsymbol{A}_{\pi(i)j}.$$

In the context of the permutation π, the set of masked tokens $\mathcal{M}$ for the i-th step consists of all tokens that appear later in the permutation, i.e., $\{j \mid \pi(j) \geq \pi(i)\}$. Summing over all i, we get the final

upper bound:

$$\begin{aligned}\mathrm{PDG}_\pi(\boldsymbol{x}) &\le \sqrt{2}\cdot\|\boldsymbol{W}\|_F\cdot B\cdot\sum_{i=1}^{b}\sum_{j:\pi(j)\ge\pi(i)}\boldsymbol{A}_{\pi(i)j}\\ &= \sqrt{2}\cdot\|\boldsymbol{W}\|_F\cdot B\cdot\sum_{i=1}^{b}\sum_{j=1}^{b}\mathbf{1}(\pi(i)\le\pi(j))\boldsymbol{A}_{ij}\\ &= \sqrt{2}\cdot\|\boldsymbol{W}\|_F\cdot B\cdot\sum_{j=1}^{b}\sum_{i\in\pi(\le j)}\boldsymbol{A}_{ij}.\end{aligned}$$

A.2 Proof of Proposition 3.2

For notational simplicity, denote $\sum_{i\in\hat{\pi}(\le j)}\boldsymbol{A}_{ij}$ as $Y_j(\tilde{\pi})$. For any function g, we have

$$\begin{aligned}&\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}\left(Y_j(\tilde{\pi})-g(\pi(j))\right)^2\\ &= \mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}\left(Y_j(\tilde{\pi})-\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}[Y_j(\tilde{\pi})]+\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}[Y_j(\tilde{\pi})]-g(\pi(j))\right)^2\\ &= \mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}\left(Y_j(\tilde{\pi})-\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}[Y_j(\tilde{\pi})]\right)^2+\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}\left(\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}[Y_j(\tilde{\pi})]-g(\pi(j))\right)^2,\end{aligned}$$

where the cross term vanishes because

$$\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}\left[Y_j(\tilde{\pi})-\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}[Y_j(\tilde{\pi})]\right]=0.$$

The first term is independent of g, while the second term is minimized when

$$g(\pi(j))=\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}[Y_j(\hat{\pi})].$$

Therefore,

$$g^*(\pi(j))=\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}[Y_j(\hat{\pi})].$$

It remains to compute $\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}\left[Y_j(\tilde{\pi})\right]$. By definition,

$$Y_j(\tilde{\pi})=\sum_{i=1}^{b}\mathbf{1}(i\in\tilde{\pi}(\le j))\mathbf{A}_{ij}.$$

Conditioned on $\tilde{\pi}(j)=\pi(j)$, each token $i(\neq j)$ is included in $\pi(\le j)$ with probability

$$\Pr(i\in\tilde{\pi}(\le j)\mid\tilde{\pi}(j)=\pi(j))=\frac{\pi(j)-1}{b-1}.$$

Thus,

$$\begin{aligned}\mathbb{E}_{\tilde{\pi}|\tilde{\pi}(j)=\pi(j)}\left[Y_j(\tilde{\pi})\right]&=\sum_{i=1}^{b}\mathbf{1}(i\neq j)\frac{\pi(j)-1}{b-1}\mathbf{A}_{ij}\\ &=\frac{\pi(j)-1}{b-1}\sum_{i=1}^{b}\mathbf{1}(i\neq j)\mathbf{A}_{ij}.\end{aligned}$$

A.3 Proof of Theorem 3.1

First, recall that

$$\widehat{\mathrm{UB}}(\boldsymbol{\theta},\boldsymbol{x},\pi)=\sqrt{2}\|\boldsymbol{W}\|_F\cdot\frac{B}{b-1}\cdot\sum_{j=1}^{b}\left((\pi(j)-1)s_j\right),\tag{8}$$

where $s_j=\sum_{i=1}^{b}\mathbf{1}(i\neq j)\boldsymbol{A}_{ij}$ is the total attention of token j. We first prove that when two tokens decoded at adjacent steps are not in descending total attention order, interchanging their decoding orders can minimize $\widehat{\mathrm{UB}}(\boldsymbol{\theta},\boldsymbol{x},\pi)$.

Recall that $\pi(i)$ is the decoding step of token x_i. Thus, $x_{\pi^{-1}(j)}$ is the token decoded at step j. Consider tokens decoded at steps j and $j+1$, but are not in descending total attention order, i.e.,

$$s_{\pi^{-1}(j)} < s_{\pi^{-1}(j+1)}. \tag{9}$$

Consider the new permutation $\bar{\pi}$ with $\bar{\pi}^{-1}(j+1) = \pi^{-1}(j)$, $\bar{\pi}^{-1}(j) = \pi^{-1}(j+1)$, and $\bar{\pi}^{-1}(k) = \pi^{-1}(k)$ for $k \neq j, j+1$. Note that (8) can be written as

$$\begin{aligned}\widehat{\mathrm{UB}}(\boldsymbol{\theta}, \boldsymbol{x}, \pi) &= \sqrt{2}\,\|\boldsymbol{W}\|_F \frac{B}{b-1} \sum_{r=1}^{b} (r-1)\, s_{\pi^{-1}(r)} \\ &= \sqrt{2}\,\|\boldsymbol{W}\|_F \frac{B}{b-1} \Big(1 \cdot s_{\pi^{-1}(2)} + \cdots + (j-1) \cdot s_{\pi^{-1}(j)} + j \cdot s_{\pi^{-1}(j+1)} + \cdots\Big).\end{aligned}$$

Similarly,

$$\widehat{\mathrm{UB}}(\boldsymbol{\theta}, \boldsymbol{x}, \bar{\pi}) = \sqrt{2}\,\|\boldsymbol{W}\|_F \frac{B}{b-1} \Big(1 \cdot s_{\pi^{-1}(2)} + \cdots + j \cdot s_{\pi^{-1}(j)} + (j-1) \cdot s_{\pi^{-1}(j+1)} + \cdots\Big).$$

Therefore,

$$\begin{aligned}&\widehat{\mathrm{UB}}(\boldsymbol{\theta}, \boldsymbol{x}, \pi) - \widehat{\mathrm{UB}}(\boldsymbol{\theta}, \boldsymbol{x}, \bar{\pi}) \\ &= \sqrt{2}\,\|\boldsymbol{W}\|_F \frac{B}{b-1} \Big((j-1)\, s_{\pi^{-1}(j)} + j\, s_{\pi^{-1}(j+1)} - j\, s_{\pi^{-1}(j)} - (j-1)\, s_{\pi^{-1}(j+1)}\Big) \\ &= \sqrt{2}\,\|\boldsymbol{W}\|_F \frac{B}{b-1} \Big(s_{\pi^{-1}(j+1)} - s_{\pi^{-1}(j)}\Big).\end{aligned}$$

Thus, with (9), swapping the token pair order reduces $\widehat{\mathrm{UB}}(\boldsymbol{\theta}, \boldsymbol{x}, \pi)$.

Next, we prove the theorem by induction:

(i) Base Case: For a sequence of length $n = 2$, the above shows that the tokens should be decoded in descending total attention order.

(ii) Inductive Step: Assume that the hypothesis holds for a sequence of length $n-1$. We now consider a sequence of n tokens. Let token k be the element with the largest total attention. We will show that token k must be first decoded. Assume, to the contrary, that token k is decoded at step $q > 1$. By swapping it with the token decoded at step $q-1$, we can reduce $\widehat{\mathrm{UB}}(\boldsymbol{\theta}, \boldsymbol{x}, \pi)$ because token k has the largest score. Therefore, in the optimal permutation π, token k must be at position 1. Once token k is fixed at the first position, we are left with a sub-problem of the remaining $n-1$ tokens. By our inductive hypothesis, $\widehat{\mathrm{UB}}(\boldsymbol{\theta}, \boldsymbol{x}, \pi)$ for these $n-1$ tokens is minimized when they are arranged in descending total attention order.

A.4 Proof of Proposition 4.1

Let $\ell(\boldsymbol{z}_i^*)$ be the confidence of token i in the ideal prediction case, and $\ell(\boldsymbol{z}_i)$ as the confidence in the real prediction case. We first expand the of a perturbed token using a first-order Taylor expansion:

$$\ell(\boldsymbol{z}_i) = \ell(\boldsymbol{z}_i^*) + \nabla \ell(\boldsymbol{z}_i^*)^\top \boldsymbol{\epsilon}_i + O(|\boldsymbol{\epsilon}_i|^2), \tag{10}$$

In the worst-case scenario, the error $\boldsymbol{\epsilon}_i$ is oriented in the direction of the gradient, i.e., $\boldsymbol{\epsilon}_i = -c \nabla \ell(\boldsymbol{z}_i^*)$ for some $c > 0$. Substituting this into (10), and dropping the $O(|\boldsymbol{\epsilon}_i|^2)$ term,

$$\ell(\boldsymbol{z}_i) = \ell(\boldsymbol{z}_i^*) - c \|\nabla \ell(\boldsymbol{z}_i^*)\|_2 \cdot \|\boldsymbol{\epsilon}_i\|_2.$$

By Assumption (iv), maximizing $\ell(\boldsymbol{z}_i)$ is equivalent to minimizing the error magnitude $\|\boldsymbol{\epsilon}_i\|_2$:

$$\arg\max_i \ell(\boldsymbol{z}_i) = \arg\min_i \|\boldsymbol{\epsilon}_i\|_2.$$

By Assumption (i) in Proposition 4.1, $\|\boldsymbol{\epsilon}_i\|_2$ is a monotonically increasing function of the total attention on masked tokens. Therefore:

$$\arg\min_i \|\boldsymbol{\epsilon}_i\|_2 = \arg\min_i \sum_{j \in \mathcal{M}} \boldsymbol{A}_{ij}.$$

Since the total row sum of the attention matrix is normalized (i.e., $\sum_{j\in\mathcal{M}} \boldsymbol{A}_{ij} + \sum_{j\notin\mathcal{M}} \boldsymbol{A}_{ij} = 1$),

$$\arg\min_i \sum_{j\in\mathcal{M}} \boldsymbol{A}_{ij} = \arg\max_i \sum_{j\notin\mathcal{M}} \boldsymbol{A}_{ij}.$$

Finally, applying assumption (ii), since g is monotonically increasing,

$$\arg\max_i \sum_{j\notin\mathcal{M}} \boldsymbol{A}_{ij} = \arg\max_i g\left(\sum_{j\notin\mathcal{M}} \boldsymbol{A}_{ij}\right) = \arg\max_i \sum_{j=1}^{n} \boldsymbol{A}_{ji}.$$

This establishes the equivalence between the confidence-based selection and Attn-Sampler.

A.5 Proof of Proposition 4.2

Let $\boldsymbol{p}_i$ be the predictive distribution for token i, and define

$$p_i = \max_j [\boldsymbol{p}_i]_j, \qquad n_i = -\mathcal{H}(\boldsymbol{p}_i).$$

Confidence-based sampling selects

$$i_{\text{conf}} = \arg\max_i p_i,$$

while entropy-based sampling selects

$$i_{\text{ent}} = \arg\max_i n_i.$$

Case (i). Suppose $p_{i^*} = 1$ and $p_i < 1$ for all $i \neq i^*$. Since $p_{i^*} = 1$, the distribution $\boldsymbol{p}_{i^*}$ is deterministic. Therefore,

$$\mathcal{H}(\boldsymbol{p}_{i^*}) = 0, \qquad n_{i^*} = 0.$$

For any $i \neq i^*$, since $p_i < 1$, the distribution $\boldsymbol{p}_i$ is not deterministic. Hence,

$$\mathcal{H}(\boldsymbol{p}_i) > 0, \qquad n_i = -\mathcal{H}(\boldsymbol{p}_i) < 0.$$

Thus,

$$n_{i^*} > n_i, \qquad \forall i \neq i^*,$$

which implies

$$i_{\text{ent}} = i^*.$$

Moreover, since $p_{i^*} = 1$ and $p_i < 1$ for all $i \neq i^*$, we also have

$$i_{\text{conf}} = i^*.$$

Therefore,

$$i_{\text{ent}} = i_{\text{conf}} = i^*.$$

Case (ii). Suppose $p_{i^*} > p_i$ for all $i \neq i^*$, and

$$-\mathcal{H}(\boldsymbol{p}_{i^*}) > \log p_i, \qquad \forall i \neq i^*.$$

We first show that, for any token i,

$$-\mathcal{H}(\boldsymbol{p}_i) \leq \log p_i.$$

Since $p_i = \max_j [\boldsymbol{p}_i]_j$, we have

$$[\boldsymbol{p}_i]_j \leq p_i, \qquad \forall j.$$

Therefore,

$$\log \frac{1}{[\boldsymbol{p}_i]_j} \geq \log \frac{1}{p_i}, \qquad \forall j.$$

Using the definition of entropy,

$$\begin{aligned}
\mathcal{H}(\boldsymbol{p}_i) &= \sum_j [\boldsymbol{p}_i]_j \log \frac{1}{[\boldsymbol{p}_i]_j} \\
&\geq \sum_j [\boldsymbol{p}_i]_j \log \frac{1}{p_i} \\
&= \log \frac{1}{p_i}.
\end{aligned}$$

Thus,

$$-\mathcal{H}(\boldsymbol{p}_i) \le \log p_i.$$

That is,

$$n_i \le \log p_i.$$

For any $i \neq i^*$, the assumption gives

$$n_{i^*} = -\mathcal{H}(\boldsymbol{p}_{i^*}) > \log p_i.$$

Combining this with $n_i \le \log p_i$, we obtain

$$n_{i^*} > n_i, \qquad \forall i \neq i^*.$$

Hence,

$$i_{\text{ent}} = \arg\max_i n_i = i^*.$$

Meanwhile, since $p_{i^*} > p_i$ for all $i \neq i^*$, confidence-based sampling also selects

$$i_{\text{conf}} = \arg\max_i p_i = i^*.$$

Therefore,

$$i_{\text{ent}} = i_{\text{conf}} = i^*.$$

This completes the proof.

B Broader impacts

This work improves the inference efficiency of dLLMs, which may enhance the accessibility and energy efficiency of large language models by reducing computational cost. Our work is primarily methodological and does not introduce new application-specific deployment scenarios. While it may influence downstream research and applications, we do not identify any direct negative societal impacts beyond the general risks associated with broader deployment of language models.

C Limitation

This work studies the sampling order selection problem in Transformer-based masked diffusion language models and introduce the Attn-sampler algorithm. One of the limitation is that current Attn-sampler cannot be directly extended to uniform diffusion. In uniform diffusion, the decoding process is more complex, since each token may be updated multiple times throughout sampling. Our Attn-Sampler is based on an optimal permutation, which relies on the one-pass decoding structure of masked dLLMs. Extending the framework to uniform diffusion is an interesting direction, and we will leave it for future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.

- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix C of the main paper.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All assumptions and proofs are provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Algorithms 1 and 2 in the main paper describe the proposed method in detail. Details of setup are provided in the experiment section.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code will be made publicly available after the paper is accepted.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Discussed in Setup section of the main paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiments are deterministic under the fixed evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: Experiments compute resources are included in Experiment section of the main paper.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research presented in this paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are included in appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use publicly available models and datasets, and cite their original creators or owners throughout the paper. Their licenses and terms of use are reviewed and respected.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.