ACCELERATING DIFFUSION LARGE LANGUAGE MODELS WITH SLOW FAST SAMPLING: THE THREE GOLDEN PRINCIPLES

Qingyan Wei[1,2] **Yaojie Zhang**[1,2] **Zhiyuan Liu**[1,2] **Dongrui Liu**[3] **Linfeng Zhang**[1,2*]
[1] School of Artificial Intelligence, Shanghai Jiao Tong University
[2] EPIC Lab, Shanghai Jiao Tong University
[3] Shanghai Artificial Intelligence Laboratory
{florawei0506}@gmail.com

ABSTRACT

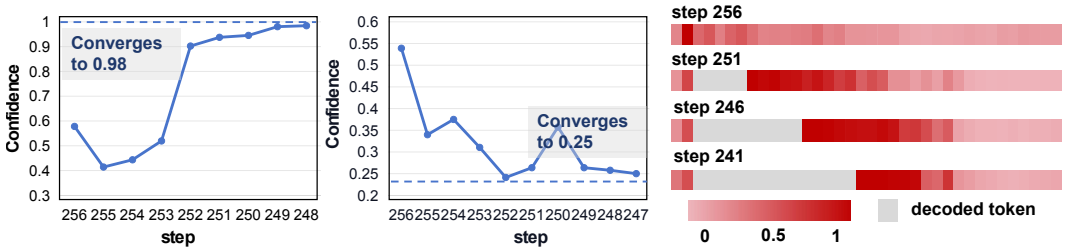
Diffusion-based language models (dLLMs) have emerged as a promising alternative to traditional autoregressive LLMs by enabling parallel token generation and significantly reducing inference latency. However, existing sampling strategies for dLLMs, such as confidence-based or semi-autoregressive decoding, often suffer from static behavior, leading to suboptimal efficiency and limited flexibility. In this paper, we propose **Slow Fast Sampling**, a novel dynamic sampling strategy that adaptively alternates between exploratory and accelerated decoding stages. Our method is guided by three golden principles: *certainty principle*, *convergence principle*, and *positional principle*, which govern when and where tokens can be confidently and efficiently decoded. We further integrate our strategy with dLLM-Cache to reduce redundant computation. Extensive experiments across benchmarks and models show that Slow Fast Sampling achieves up to **15.63**$\times$ speedup on LLaDA with minimal accuracy drop, and up to **34.22**$\times$ when combined with caching. Notably, our approach outperforms strong autoregressive baselines like LLaMA3 8B in throughput, demonstrating that well-designed sampling can unlock the full potential of diffusion LLMs for fast and high-quality generation. Our codes will be released on GitHub.

1 INTRODUCTION

Large Language Models (LLMs) (Zhao et al., 2025) have rapidly become cornerstone technologies in artificial intelligence, demonstrating remarkable capabilities across a diverse range of natural language understanding and generation tasks. However, the prevalent autoregressive nature of most LLMs, where tokens are generated sequentially one after another, introduces significant inference latency, particularly for long sequences. To address this inherent bottleneck, diffusion-based LLMs (Ye et al., 2025; Nie et al., 2025b) have emerged as a promising alternative paradigm. These models are capable of generating multiple tokens in parallel, departing from the strict token-by-token process. This parallel decoding capability offers the distinct advantage of potentially accelerating text generation significantly, positioning diffusion LLMs as a compelling and forward-looking direction for efficient language model inference.

However, current ways of sampling with diffusion LLMs often don't perform as well as they could. Common methods include confidence-based selection (Chang et al., 2022), where a fixed number of tokens with the highest confidence scores are chosen globally for regeneration. Another popular method, semi-autoregressive decoding (Arriola et al., 2025), divides the sequence into fixed blocks and decodes within them. Unfortunately, these methods frequently yield unsatisfactory results (*i.e.,* significant accuracy drop when decoding many tokens in parallel), and are characterized by a static, constant sampling speed throughout the generation process. This lack of flexibility highlights the need for a more dynamic sampling approach: one that can smartly decide how many tokens to sample at each step and where these tokens should be located in the sequence.

*Corresponding author.



(a) Confidence convergence diagram of the 1st token and the 12th token

(b) The Process of Change in the Confidence Map of LLaDA-Base

Figure 1: **Illustration of the Three Golden Principles for Sampling in Diffusion-based LLMs.** (a) Demonstrates the **Convergence Principle**: as decoding proceeds, the confidence values of tokens converge to stable values, either high (e.g., 12th token to 0.98) or low (e.g., 1st token to 0.25). (b) Visualizes the evolution of the **Confidence Map** over 256 diffusion steps on GSM8K of LLaDA Base. High-confidence tokens (in deep red) emerge progressively and are preferentially decoded (**The Certainty Principle**), while selection tends to cluster in contiguous regions (**The Positional Principle**), enabling cache reuse and efficient acceleration.

Motivated by these limitations, we introduce a novel dynamic sampling approach designed to accelerate diffusion-based LLMs, aiming to unlock the real potential of diffusion-based LLMs under high-parallel decoding. As illustrated in Figure 1, our method is guided by three core observations, which we formulate as the **Three Golden Principles** for effective acceleration are as follows:

- **The Certainty Principle:** Tokens exhibiting higher confidence are inherently more determined. Consequently, they are more likely to be decoded correctly early in the process and require less adjustment in subsequent diffusion steps.

- **The Convergence Principle:** As the diffusion process unfolds and tokens are progressively refined, the semantic meaning of many tokens stabilizes, and their associated confidence scores converge towards a steady value. This convergence indicates that these tokens have largely settled into their final form and require minimal further refinement.

- **The Positional Principle:** We observe that even without explicit constraints, the model's sampling preferences often gravitate towards tokens in specific, frequently neighboring or clustered, positions. This inherent positional bias can be strategically exploited. For instance, parts of the sequence can be effectively cached, leading to significant acceleration gains.

Integrating these principles, we propose our **Slow Fast Sampling** approach, which operates in two distinct phases: an **Exploratory Stage** and an **Accelerated Decoding Stage**. In the initial Exploratory Stage, the model performs less constrained decoding. This allows it to freely explore the sequence space. Leveraging the Positional Principle, the model identifies promising regions and predicts a target span where tokens exhibit emerging high Certainty and initial Convergence. Subsequently, the Accelerated Decoding Stage capitalizes on this by rapidly and parallelly decoding these high-certainty, converging tokens within the identified span. This strategic division enables substantial speed-ups by efficiently processing tokens that are already largely determined, focusing computational effort where it's most impactful. As shown in Figure 2, our method achieves a maximum speedup of $15.63\times$ on LLaDA. When further combined with **dLLM-Cache** (Liu et al., 2025), the speedup increases to an impressive $34.22\times$ with only marginal accuracy degradation. Our contributions are threefold:

1. We propose the **Three Golden Principles** based on token certainty, convergence, and positional influence, which critically govern effective and efficient sampling in diffusion-based LLMs.

2. Building on these principles, we introduce **Slow Fast Sampling**, a novel two-stage dynamic strategy specifically designed to leverage these principles for optimal acceleration of diffusion LLM .

3. Through experiments on various benchmarks, we demonstrate that Slow Fast Sampling achieves significant inference acceleration (e.g., up to ***15.63***$\times$ on LLaDA with Slow Fast Sampling alone, and up to ***34.22***$\times$ when combined with dLLM-Cache) without compromising response quality, thereby offering a superior speed-quality trade-off compared to baseline and simpler sampling methods.

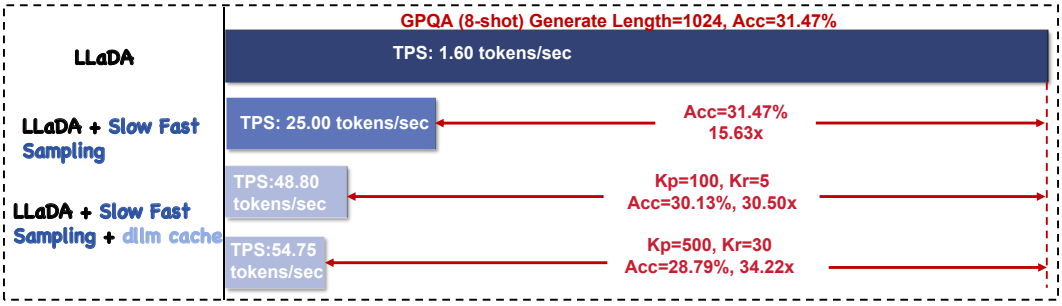


Figure 2: **Throughput and Accuracy Comparison on GPQA (8-shot, Length=1024) with LLaDA and Our Proposed Methods.** We evaluate LLaDA under three settings: (1) vanilla decoding, (2) with our proposed **Slow Fast Sampling**, and (3) **Slow Fast Sampling** further enhanced by **dLLM-Cache**. Compared to the vanilla setting, Slow Fast Sampling alone achieves a **15.63**× speedup while maintaining comparable accuracy. With dLLM-Cache, throughput improves further to **54.75 tokens/sec** (up to **34.22**× speedup), with only minor drops in accuracy. This demonstrates the strong efficiency gains and flexibility enabled by our dynamic strategy.

2 Related Work

2.1 Diffusion Models for Language

Diffusion Models (DMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) have revolutionized generative modeling, particularly in continuous domains like images (Rombach et al., 2022; Peebles & Xie, 2023). However, adapting these models to discrete data such as text presents unique challenges due to its discrete nature. A promising approach in discrete diffusion models involves Masked Diffusion Models (MDMs) (Austin et al., 2021; Lou et al., 2023; Shi et al., 2024; Nie et al., 2025a;b; Hoogeboom et al., 2021; Campbell et al., 2022), which iteratively predict masked tokens based on their context. These advancements have transformed text generation, offering a compelling alternative to autoregressive paradigms in large language models (LLMs). Notable examples include LLaDA (Nie et al., 2025b), an 8B MDM trained from scratch with a bidirectional Transformer, and Dream (Ye et al., 2025), which initializes from pre-trained ARM weights. Both models demonstrate performance comparable to similarly-sized ARMs like LLaMA3 8B (Dubey et al., 2024). Their bidirectional architecture may overcome ARM limitations such as the reversal curse (Berglund et al., 2023), making diffusion a competitive alternative for foundational LLMs.

2.2 Acceleration Methods for Diffusion-based LLMs

The high inference latency of diffusion LLMs, primarily due to their iterative denoising process (Nie et al., 2025b; Ye et al., 2025), has spurred research into acceleration techniques. Various strategies have been developed, mainly including caching mechanisms and advanced sampling techniques.

Caching Mechanisms. Feature caching reduces redundant computations by reusing intermediate features across denoising steps. The dLLM-Cache (Liu et al., 2025) combines long-interval prompt caching and short-interval response caching, utilizing a V-verify mechanism to selectively update dynamic tokens, significantly improving inference speed.

Advanced Sampling Techniques. Optimizing the sampling process itself is another major direction for accelerating diffusion LLMs. Low-confidence remasking (Chang et al., 2022; Nie et al., 2025b) prioritizes high-confidence tokens to speed up convergence; semi-autoregressive (Nie et al., 2025b; Arriola et al., 2025) remasking divides sequences into blocks, applying random and low-confidence strategies. Additionally, exact simulation methods for MDMs like the first-hitting sampler (Zheng et al., 2024) have made progress in reducing sampling steps or enhancing per-step efficiency.

However, these sampling methods are often static and lack flexibility. To address this, we propose **Slow Fast Sampling**, a novel dynamic sampling method that achieves more efficient inference acceleration by integrating principles of certainty, convergence, and positional influence.

3 METHODOLOGY

3.1 PRELIMINARY

Inference Process of Diffusion Large Language Models. Diffusion Large Language Models (dLLMs) generate text $\mathbf{y} = (y_1, \ldots, y_L)$ from a prompt $\mathbf{c} = (c_1, \ldots, c_M)$ through an iterative denoising process. The model refines an intermediate state $\mathbf{y}^{(k)} \in \mathcal{T}^L$ over N discrete steps, from $k = N$ to $k = 0$, where $\mathcal{T}$ is the token vocabulary. The process begins with a fully masked sequence:

$$\mathbf{y}^{(N)} = (\underbrace{\texttt{[MASK]}, \ldots, \texttt{[MASK]}}_{L \text{ times}}) \tag{1}$$

where $\texttt{[MASK]} \in \mathcal{T}$ is the special mask token.

At each step $k \in \{N, N-1, \ldots, 1\}$, a mask predictor p_θ estimates the original clean sequence $\mathbf{r}_0 = (r_{0,1}, \ldots, r_{0,L})$ from the noisy state $\mathbf{y}^{(k)}$ and prompt $\mathbf{c}$:

$$P_\theta(\mathbf{r}_0 | \mathbf{c}, \mathbf{y}^{(k)}) \tag{2}$$

An estimate of the clean sequence at step k, $\hat{\mathbf{r}}_0^{(k)}$, is obtained through greedy decoding:

$$\hat{r}_{0,i}^{(k)} = \arg\max_{v \in \mathcal{T}} P_\theta(r_{0,i} = v | \mathbf{c}, \mathbf{y}^{(k)}) \quad \forall i \in \{1, \ldots, L\} \tag{3}$$

Although the predictor p_θ can decode all masked tokens $\texttt{[MASK]}$ in one step, to ensure high-quality generation, dLLM adopts a multi-step decoding process. At each step, the remask strategy refines the tokens. The transition to the next state $\mathbf{y}^{(k-1)}$ is governed by a sampling strategy S:

$$\mathbf{y}^{(k-1)} = S(\hat{\mathbf{r}}_0^{(k)}, \mathbf{y}^{(k)}, k) \tag{4}$$

This iterative denoising process is a sampling procedure. Our work aims to improve the sampling efficiency of dLLM inference, which can be computationally expensive due to the N sequential steps. Using LLaDA (Nie et al., 2025b) as an example, we describe its core sampling strategies. In LLaDA, time steps are defined as $t_k = k/N$ and $t_{k-1} = (k-1)/N$.

LLaDA explores various strategies for the transition function S in Equation 4, which differ mainly in how tokens from $\hat{\mathbf{r}}_0^{(k)}$ update $\mathbf{y}^{(k)}$ to form $\mathbf{y}^{(k-1)}$, especially for positions that were $\texttt{[MASK]}$ in $\mathbf{y}^{(k)}$:

Random Remasking. In this strategy, for each position i:
If $y_i^{(k)} \neq \texttt{[MASK]}$, then $y_i^{(k-1)} = y_i^{(k)}$ (known tokens are preserved).
If $y_i^{(k)} = \texttt{[MASK]}$, then $y_i^{(k-1)}$ is set to $\hat{r}_{0,i}^{(k)}$ with probability $1 - \frac{k-1}{k}$, and remains $\texttt{[MASK]}$ with probability $\frac{k-1}{k}$, ensuring the expected number of masked tokens aligns with the noise schedule.

Low-Confidence Remasking. This deterministic strategy aims to improve sample quality by selectively unmasking tokens. For each position i, if $y_i^{(k)} = \texttt{[MASK]}$, the model predicts $\hat{r}_{0,i}^{(k)}$ and computes its confidence. Specifically, the confidence c_i is given by:

$$c_i = P_\theta(\hat{r}_{0,i}^{(k)} | \mathbf{c}, \mathbf{y}^{(k)}). \tag{5}$$

If $y_i^{(k)} \neq \texttt{[MASK]}$, then $c_i = 1$.

The target number of unmasked tokens for state $\mathbf{y}^{(k-1)}$ is given by:

$$n_{un} = \lfloor L(1 - t_{k-1}) \rfloor = \lfloor L \left(1 - \frac{k-1}{N}\right) \rfloor. \tag{6}$$

The tokens corresponding to the n_{un} highest confidences are unmasked in $\mathbf{y}^{(k-1)}$, while the remaining positions are set to $\texttt{[MASK]}$.

Semi-Autoregressive Remasking. This strategy is used in LLaDA after Supervised Fine-Tuning. The sequence is divided into blocks, and generation proceeds block by block from left to right. Within each block, the random remasking or the low-confidence remasking strategy is applied iteratively to denoise the block before moving to the next.

The choice of sampling strategy S, along with the number of steps N, significantly affects both the generation quality and latency.

3.2 THREE GOLDEN PRINCIPLES OF DLLMS

Building upon the iterative denoising process outlined in Section 3.1, our empirical analysis of dLLM behavior, particularly with models like LLaDA, has revealed consistent patterns in token generation. These patterns, which we term the **Three Golden Principles**, form the bedrock of our proposed acceleration strategy. They describe how token certainty, convergence, and positional effects interact during the diffusion process, offering key insights into optimizing sampling.

The Certainty Principle: High Confidence Indicates Determination. We observe that tokens predicted with high confidence by the mask predictor p_θ are significantly more likely to be part of the final, correct sequence and tend to remain unchanged in subsequent denoising steps. The confidence for a predicted token $\hat{r}_{0,i}^{(k)}$ at position i and step k is given by $P_\theta(\hat{r}_{0,i}^{(k)}|\mathbf{c}, \mathbf{y}^{(k)})$. As illustrated in Figure 1 (b), at any given step k, a subset of tokens typically exhibits substantially higher confidence scores compared to others. These high-confidence tokens are prime candidates for early "acceptance" or less frequent resampling.

Implication for Acceleration: By prioritizing tokens that quickly reach a high confidence threshold, we can reduce redundant computations on already determined parts of the sequence.

The Convergence Principle: Tokens Stabilize Over Iterations. During the iterative refinement process, individual tokens (both their predicted identity $\hat{r}_{0,i}^{(k)}$ and their confidence $P_\theta(\hat{r}_{0,i}^{(k)}|\mathbf{c}, \mathbf{y}^{(k)})$) undergo a period of fluctuation before settling. As shown in Figure 1 (a), a representative token might initially change its predicted identity and confidence across several early diffusion steps. However, as k decreases, the token's confidence converges are often to a stable value. This convergence signals that the model has formed a consistent belief about the token's identity within its current context.

Implication for Acceleration: Tokens that have demonstrated convergence (i.e., stable identity and confidence over a window of recent steps) are less likely to change. Aggressively decoding can prevent unnecessary re-evaluation, thereby speeding up the process.

The Positional Principle: Decoding Exhibits Regional Preferences. Beyond individual token behaviors, we find that the generation process often exhibits spatial patterns. High-confidence and early-converging tokens do not appear randomly scattered throughout the sequence. Instead, they frequently emerge in contiguous blocks or localized regions, as suggested by Figure 1 (b). This phenomenon might be due to local semantic dependencies or the influence of strongly contextualized parts of the prompt $\mathbf{c}$. For example, after a few initial steps, a particular span of tokens might collectively achieve high confidence and stability, while other regions remain largely masked or uncertain. The model appears to focus its decoding efforts on specific segments at different stages.

Implication for Acceleration: Recognizing these regions allows for targeted decoding. Instead of uniformly processing all tokens, computational resources can be concentrated on regions that are currently most amenable to decoding.

These three principles collectively indicate that a one-size-fits-all, static sampling strategy is inherently inefficient. They motivate a dynamic approach where the number of tokens sampled, their selection criteria, and their locations are adapted based on the evolving state of the generated sequence. Our Slow Fast Sampling methodology, detailed in Section 3.3, is designed to explicitly leverage these observations to achieve significant inference speedups.

3.3 SLOW-FAST SAMPLING

1. Exploratory Stage (Slow Phase): Identifying the Next Stable Region. As illustrated in Figure 3, the primary goal of this stage is to cautiously advance decoding while identifying a promising, stable region for subsequent rapid processing. Starting from s_{cycle} and extending to the end of the full sequence L, this stage operates as follows for a limited number of dLLM steps:

- **Cautious Decoding:** At each dLLM step k within this stage, we perform a conservative decoding update. We select the top-k_{slow} tokens within the current exploratory window $[s_{cycle}, L]$ that exhibit the highest confidence $P_\theta(\hat{r}_{0,i}^{(k)}|\mathbf{c}, \mathbf{y}^{(k)})$, and these are unmasked to form part of $\hat{\mathbf{r}}_0^{(k)}$ (as per Equation 3 for these tokens, followed by Equation 4).

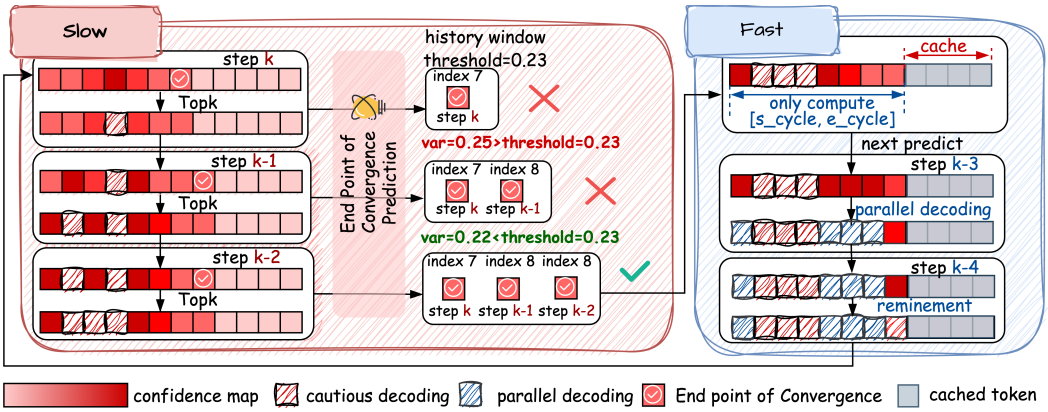


Figure 3: **Overview of the Slow Fast Sampling Pipeline: From Exploratory to Accelerated Decoding.** The method alternates between a **Slow (Exploratory) stage** and a **Fast (Accelerated) stage** for efficient token generation. In the Slow phase (left), the model conducts *cautious decoding* by selecting top-k high-confidence tokens per step while continuously predicting the *End Point of Convergence* and calculating confidence variance across a history window. Once variance drops below threshold (e.g., $0.22 < 0.23$), the corresponding region $[s_{cycle}, e_{cycle}]$ is considered stable. In the Fast phase (right), this stable span is decoded in parallel with aggressive unmasking of high-confidence tokens, while tokens beyond the span are temporarily skipped and their results *cached* for reuse. This alternating structure reduces redundant computation and accelerates decoding while maintaining output quality.

- **End Point of Convergence Prediction:** Concurrently, the model predicts a end point of candidate convergence, $e_{cand}^{(k)}$. This is defined as the furthest token position $i \in [s_{cycle}, L]$ for which the predicted confidence $P_\theta(\hat{r}_{0,i}^{(k)} | \mathbf{c}, \mathbf{y}^{(k)})$ exceeds a minimum confidence threshold τ_{min_conf}. Mathematically:

$$e_{cand}^{(k)} = \max\{i \mid i \in [s_{cycle}, L] \wedge P_\theta(\hat{r}_{0,i}^{(k)} | \mathbf{c}, \mathbf{y}^{(k)}) > \tau_{min\_conf}\} \tag{7}$$

  This $e_{cand}^{(k)}$ represents an estimate of how far into the sequence the model can confidently decode at the current step k.

- **Stability Check:** Throughout the Exploratory Stage, candidate convergence horizons $e_{cand}^{(j)}$ predicted at each dLLM step j are tracked. Let $H_W(k_s) = \{e_{cand}^{(l)} \mid l \in [\max(1, k_s - W_{hist} + 1), k_s]\}$ be the set of candidate horizons in the sliding window of the latest W_{hist} predictions ending at exploratory step k_s. Stability is achieved, concluding this stage, when $\mathrm{Var}(H_W(k_s)) < \sigma_{stable}^2$ for a window where $k_s \geq W_{hist}$. The exploratory stage ends at step k_{final}, defined as:

$$k_{final} = \min\left(\{k_s \mid W_{hist} \leq k_s \leq K_{max} \wedge \mathrm{Var}(H_W(k_s)) < \sigma_{stable}^2\} \cup \{K_{max}\}\right) \tag{8}$$

  If the stability criterion is not met by K_{max} (the maximum allotted exploratory steps), then $k_{final} = K_{max}$. The cycle's convergence horizon e_{cycle}, is then set to the mean of the candidate horizons in the final window $H_W(k_{final})$:

$$e_{cycle} = \mathrm{Mean}(H_W(k_{final})) = \frac{1}{|H_W(k_{final})|} \sum_{e \in H_W(k_{final})} e \tag{9}$$

Once this stability criterion is met, the Exploratory Stage concludes. The endpoint for the subsequent Accelerated Decoding Stage, e_{cycle}, is set to the last recorded candidate horizon, $e_{cand}^{(k)}$. If stability is not achieved within a maximum number of exploratory steps, e_{cycle} can be set to a conservatively determined position or the process might default to a full-sequence cautious decode for that cycle.

2. Accelerated Decoding Stage (Fast Phase): Rapid Parallel Refinement. As shown in the right half of Figure 3, once a stable region $[s_{cycle}, e_{cycle}]$ is identified, this stage aims to rapidly denoise tokens within this span, while efficiently handling tokens outside it:

Table 1: **Performence of LLaDA 8B and Dream 7B with Slow Fast Sampling** on 8 benchmarks.

Task	Method	Inference Efficiency		Performance	Method	Inference Efficiency		Performance
		TPS↑	Speed(TPS)↑	Score↑		TPS↑	Speed(TPS)↑	Score↑
Mathematics & Science								
GSM8K	LLaDA Base	4.55	1.00×	69.83	Dream Base	8.16	1.00×	77.02
	+ Sampling	14.57$_{+10.02}$	3.20×$_{+2.20}$	69.59$_{-0.27}$	+ Sampling	17.15$_{+8.99}$	2.10×$_{+1.10}$	76.50$_{-0.52}$
GPQA	LLaDA Base	3.31	1.00×	31.47	Dream Base	5.43	1.00×	35.93
	+ Sampling	16.36$_{+13.05}$	4.94×$_{+3.94}$	31.91$_{+0.44}$	+ Sampling	16.56$_{+11.13}$	3.05×$_{+2.05}$	35.94$_{+0.01}$
Math	LLaDA Base	5.14	1.00×	30.16	Dream Base	8.48	1.00×	38.68
	+ Sampling	11.27$_{+6.13}$	2.19×$_{+1.19}$	29.64$_{-0.52}$	+ Sampling	23.00$_{+14.52}$	2.71×$_{+1.71}$	38.24$_{-0.44}$
General Tasks								
MMLU-pro	LLaDA Base	9.16	1.00×	23.30	Dream Base	14.97	1.00×	24.14
	+ Sampling	23.14$_{+13.98}$	2.53×$_{+1.53}$	23.85$_{+0.55}$	+ Sampling	22.80$_{+7.83}$	1.52×$_{+0.52}$	22.91$_{-1.23}$
MMLU	LLaDA Base	5.02	1.00×	62.11	Dream Base	8.46	1.00×	72.61
	+ Sampling	16.81$_{+11.79}$	3.35×$_{+2.35}$	66.56$_{+4.45}$	+ Sampling	18.43$_{+9.97}$	2.18×$_{+1.18}$	75.13$_{+2.52}$
BBH	LLaDA Base	4.04	1.00×	44.97	Dream Base	6.93	1.00×	51.83
	+ Sampling	21.19$_{+17.15}$	5.24×$_{+4.24}$	44.60$_{-0.37}$	+ Sampling	28.14$_{+21.21}$	4.06×$_{+3.06}$	50.55$_{-1.28}$
Code								
MBPP	LLaDA Base	4.98	1.00×	40.80	Dream Base	8.92	1.00×	54.20
	+ Sampling	13.32$_{+8.34}$	2.67×$_{+1.67}$	41.00$_{+0.20}$	+ Sampling	29.07$_{+20.15}$	3.26×$_{+2.26}$	54.60$_{+0.40}$
HumanEval	LLaDA Base	11.24	1.00×	31.71	Dream Base	11.49	1.00×	54.26
	+ Sampling	35.46$_{+24.22}$	3.15×$_{+2.15}$	33.54$_{+1.83}$	+ Sampling	25.38$_{+13.89}$	2.21×$_{+1.21}$	52.43$_{-1.83}$

- **Out-of-Span Caching:** For tokens outside the identified span (i.e., positions $i > e_{cycle}$), if their current predicted confidence is low (e.g., below τ_{min_conf}), their predicted values $\hat{r}_{0,i}^{(k)}$ from one dLLM step within this stage are computed and then cached. These cached values can be reused in subsequent dLLM steps for these positions, provided they remain outside an active decoding span, thus saving redundant computations.
- **In-Span Parallel Decoding:** Within the span $[s_{cycle}, e_{cycle}]$, an aggressive parallel decoding is attempted. All tokens $y_i^{(k)} = \texttt{[MASK]}$ for $i \in [s_{cycle}, e_{cycle}]$ for which the predicted confidence $P_\theta(\hat{r}_{0,i}^{(k)} | \mathbf{c}, \mathbf{y}^{(k)})$ exceeds the high certainty threshold τ_{high_conf}, i.e.,

$$P_\theta(\hat{r}_{0,i}^{(k)} | \mathbf{c}, \mathbf{y}^{(k)}) > \tau_{high\_conf} \tag{10}$$

  are unmasked by setting their value to the corresponding prediction $\hat{r}_{0,i}^{(k)}$. This update is performed simultaneously for all such qualifying tokens in a single conceptual step to form $\hat{\mathbf{r}}_0^{(k)}$.
- **Fallback Top-k Refinement:** If the number of tokens meeting the τ_{high_conf} criterion within the span is insufficient to make substantial progress (e.g., less than one), we revert to a more conservative update for this dLLM step within the span. Specifically, we select the top-k_{fast} tokens within $[s_{cycle}, e_{cycle}]$ based on confidence and unmask them. This ensures steady progress even when widespread high certainty is not yet achieved.

After the Accelerated Decoding Stage completes, the starting position for the next cycle's Exploratory Stage is updated to $s_{cycle} \leftarrow e_{cycle}$. This cyclical process of exploration and accelerated decoding continues until the entire sequence is generated. This Slow Fast approach dynamically adapts the decoding focus and intensity, leveraging the Certainty and Positional principles to identify promising regions and the Convergence principle to stabilize them efficiently.

4 Experiment

4.1 Experiment Settings

Implementation Details To evaluate the effectiveness of our proposed dynamic sampling approach Slow Fast Sampling, we conducted experiments on representative dLLMs: LLaDA 8B and Dream 7B, focusing on measuring the inference acceleration across various benchmarks. All experiments were conducted on NVIDIA RTX 4090 GPUs.

Table 2: **Performance of LLaDA Base with Slow Fast Sampling and dLLM-Cache.**

Task	Method	Inference Efficiency		Performance
		TPS↑	Speed(TPS)↑	Score↑
Mathematics & Science				
GSM8K	LLaDA Base	4.55	1.00×	69.83
	Sampling + Cache	26.99$_{+22.44}$	5.83×$_{+4.83}$	69.60$_{-0.23}$
GPQA	LLaDA Base	3.31	1.00×	31.47
	Sampling + Cache	29.06$_{+25.75}$	8.78×$_{+7.78}$	33.48$_{+2.01}$
Math	LLaDA Base	5.14	1.00×	30.16
	Sampling + Cache	26.50$_{+21.36}$	5.16×$_{+4.16}$	29.42$_{-0.74}$
General Tasks				
MMLU-pro	LLaDA Base	9.16	1.00×	23.30
	Sampling + Cache	33.38$_{+24.22}$	3.64×$_{+2.64}$	25.53$_{+2.23}$
MMLU	LLaDA Base	5.02	1.00×	62.11
	Sampling + Cache	38.42$_{+33.40}$	7.65×$_{+6.65}$	61.20$_{-0.91}$
BBH	LLaDA Base	4.04	1.00×	44.97
	Sampling + Cache	36.04	8.92×$_{+7.92}$	44.81$_{-0.16}$
Code				
MBPP	LLaDA Base	4.98	1.00×	40.80
	Sampling + Cache	27.26$_{+22.28}$	5.47×$_{+3.87}$	39.00$_{-1.80}$
HumanEval	LLaDA Base	11.24	1.00×	31.71
	Sampling + Cache	41.14$_{+29.90}$	3.66×$_{+2.66}$	31.10$_{-0.61}$

Evaluation Metrics We evaluated sampling acceleration and generation quality of Slow Fast Sampling using a set of quantitative metrics. Inference speed is measured in Tokens Per Second (TPS), indicating the average number of tokens generated per second. Generation quality is assessed using task-specific metrics (e.g., accuracy on GSM8K), reflecting the model's performance under inference acceleration.

4.2 Main Results

Performance and efficiency gains across models Table 1 reports throughput and model performance for both LLaDA 8B and Dream 7B, with and without Slow Fast Sampling. These results demonstrate that our method brings significant improvements in inference efficiency and achieves lossless acceleration in most cases. In our experiments, the key hyperparameters of Slow Fast Sampling, τ_{min_conf} and τ_{high_conf}, were set to 0.1 and 0.85, respectively. The remaining hyperparameters in our method were set as follows: maximum exploratory steps $K_{max} = 8$, sliding window size $W_{hist} = 2$, and stable variance threshold $\sigma^2_{stable} = 1.0$, consistent with the default configuration used in Section 4.2 and ablation studies.

Compatibility with dLLM-Cache Recently, several studies have explored leveraging feature cache to reduce the computational cost of dLLM inference. Our Slow Fast Sampling is highly compatible with existing caching mechanisms and the integration can lead to higher acceleration compared to using either alone. Table 2 compares the performance and inference speed of LLaDA with and without applying our method in combination with dLLM-Cache. The results show that this integration can deliver higher throughput while maintaining comparable model performance.

Comparison with Other Sampling Strategies We compared our method Slow Fast Sampling with three alternative sampling strategies: difussion sampling, diffusion semi-autoregressive sampling and autoregressive (AR) sampling. Diffusion sampling adopts a remasking strategy to iteratively select token to decode in parallel, while AR sampling generates tokens strictly from left to right. The semi-autoregressive approach generates blocks left-to-right and applies the remasking strategy within each block. As shown in Table 3, our method Slow Fast Sampling achieves higher inference efficiency while maintaining competitive generation quality.

Sampling Strategy	TPS↑	Score↑
Autoregressive (AR)	5.25	60.80
Diffusion Sampling	4.55	69.83
Semi-Autoregressive	5.44	66.41
Slow Fast Sampling	9.87	69.59

Table 3: **Comparison of Sampling Strategies** on inference efficiency and generation performance.

Method	TPS↑	Speed(TPS)↑	Accuracy↑
LLaMA3 8B (Dubey et al., 2024)	33.79	1.00×	31.92
LLaDA Base	$1.6_{-32.19}$	1.00×	$31.47_{-0.45}$
+ Sampling	$25.00_{-8.79}$	15.63×	$31.47_{+0.00}$
+ Sampling + Cache ($K_p = 100, K_r = 5$)	$48.80_{+15.01}$	30.50×	$30.13_{-1.34}$
+ Sampling + Cache ($K_p = 500, K_r = 30$)	$54.75_{+20.96}$	34.22×	$28.79_{-3.13}$

Table 4: **Comparison of LLaDA 8B Base with other representative LLMs**. Compared to LLaMA3 8B, LLaDA with Slow Fast Sampling and dLLM-Cache achieves significantly higher throughput (up to +20.96 TPS) while maintaining comparable accuracy.

4.3 Ablation Study

Effect of minimum confidence threshold τ_{min_conf} and high certainty threshold τ_{high_conf}.

The core confidence thresholds, τ_{min_conf} and τ_{high_conf}, dictate the behavior of our Slow Fast pipeline. As shown in Figure 4, the minimum confidence threshold τ_{min_conf} defines the candidate region in the exploratory stage. A moderate value of $\tau_{min\_conf} = 0.1$ is optimal, as lower values lead to unstable regions and higher values create overly conservative, inefficient regions. The high certainty threshold τ_{high_conf} directly manages the speed-quality trade-off during the accelerated stage. A higher value leads to more cautious and accurate generation at the cost of speed (TPS). We select $\tau_{high\_conf} = 0.85$, which achieves a near-peak GSM8K score without a drastic reduction in inference speed, striking an effective balance.

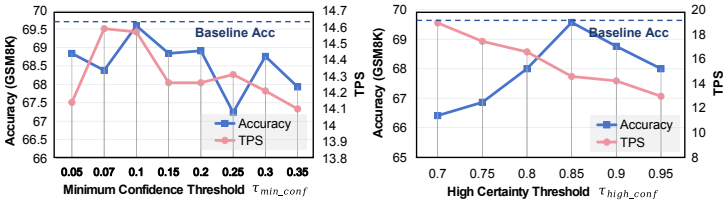


Figure 4: **Effect of Confidence Thresholds.** τ_{min_conf} controls the exploratory range, and τ_{high_conf} balances accuracy and speed during fast decoding. Other hyperparameters follow the default settings in Section 4.2.

Effect of Hyperparameters in the Stability Check. The stability check, which transitions the model from the slow to the fast phase, is governed by three hyperparameters whose effects are shown in Figure 5. The maximum number of exploratory steps, K_{max}, is crucial for allowing the convergence horizon to stabilize; we find $K_{max} = 8$ provides sufficient exploration for high-quality generation without incurring excessive overhead. This is complemented by the sliding window size, W_{hist}. Since prediction changes and convergence occur rapidly, we find a smaller window of $W_{hist} = 2$ is effective, striking a strong balance between maintaining quality and maximizing inference speed. Finally, a strict stable variance threshold of $\sigma^2_{stable} = 1.0$ ensures that the accelerated phase is only triggered for genuinely stable regions, solidifying the reliability of our method.

Outperforming Autoregressive LLMs in Inference Speed. As shown in Table 2, when equipped with our Slow Fast Sampling and dLLM-Cache, LLaDA Base not only accelerates significantly over its default setting but also **outperforms the autoregressive LLaMA3 8B model** Dubey et al. (2024) in inference speed (54.75 vs. 33.79 TPS), while maintaining comparable accuracy. This demonstrates that diffusion-based LLMs, with proper sampling and optimization, can surpass traditional autoregressive models in both efficiency and practicality.

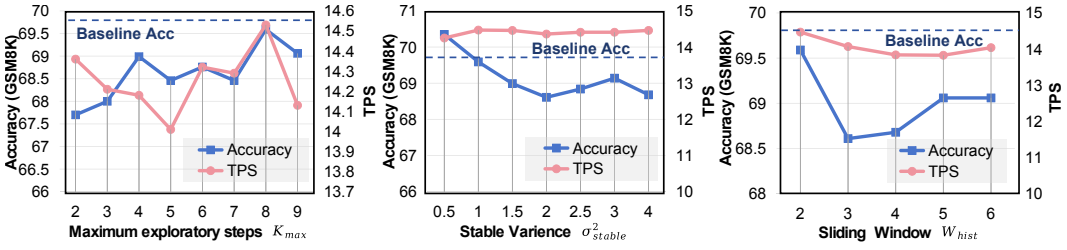


Figure 5: **The sensitivity study on hyper-parameters in the stability-Check.** Accuracy and TPS vary with K_{max}, σ^2_{stable}, and W_{hist}. The chosen defaults ($K_{max} = 8$, $\sigma^2_{stable} = 1.0$, $W_{hist} = 2$) offer strong speed-quality trade-offs.

5 CONCLUSION

In this work, we present **Slow Fast Sampling**, a dynamic and principled approach to accelerate diffusion-based language models. By leveraging three key observations: token certainty, convergence, and positional bias, we design a two-stage decoding pipeline that adaptively balances exploration and efficient parallel decoding. Extensive experiments across benchmarks demonstrate that our method not only significantly improves inference speed (up to **15.63**× and up to **34.22**× combined with dLLM-Cache), but also maintains strong generation quality, outperforming even autoregressive LLMs in throughput. We believe this work marks an important step toward making diffusion LLMs practical and competitive in real-world deployment.

REFERENCES

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. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*, 2023.

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

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022.

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 preprint arXiv:2407.21783*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyan Wei, Shaobo Wang, and Linfeng Zhang. dllm-cache: Accelerating diffusion large language models with adaptive caching, 2025. URL https://github.com/maomaocun/dLLM-cache.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text, 2025a. URL https://arxiv.org/abs/2410.18514.

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, 2025b. URL https://arxiv.org/abs/2502.09992.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. Simplified and generalized masked diffusion for discrete data. *arXiv preprint arXiv:2406.04329*, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b, 2025. URL https://hkunlp.github.io/blog/2025/dream.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2025. URL https://arxiv.org/abs/2303.18223.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.