Re-evaluating Confidence Remasking in Masked Diffusion Language Models

Stipe Frković[1,*] **Metod Jazbec**[1,*] **Dan Zhang**[2]
Christian A. Naesseth[1] **Ilija Bogunovic**[3] **Eric Nalisnick**[4]
[1]UvA-Bosch Delta Lab, University of Amsterdam [2]Bosch Center for AI
[3]University of Basel [4]Johns Hopkins University

Abstract

Masked diffusion language models (dLLMs) have recently emerged as a competitive alternative to autoregressive language models, with the promise of faster inference via parallel token generation. A notable limitation of the masked formulation, however, is that once a token has been unmasked it can no longer be revised, leaving dLLMs vulnerable to early sampling mistakes. Confidence-based remasking methods aim to address this in a training-free manner by revising low-confidence tokens, with encouraging early results. In this work, we compare the accuracy–compute Pareto frontiers of three such methods (WINO, Saber, and COVER) to those of confidence-based unmasking alone (Fast-dLLM). We find that under greedy decoding, all three perform on par with Fast-dLLM at commonly used shorter block lengths; gains of remasking appear only at longer block lengths, where confidence-based unmasking is known to degrade. Under stochastic decoding, WINO reduces generation diversity beyond that of unmasking alone, while it helps recover from errors introduced by learned Bernoulli unmasking policies. Our results show that the benefits of confidence-based remasking are highly setting-dependent, underscoring the need for more comprehensive evaluation. Finally, our analyses point to the dLLM, not the remasking rule, as the bottleneck, suggesting that effective remasking may require changes to training, not only to inference.

1 Introduction

Autoregressive (AR) language models have become the dominant paradigm for text generation, but their sequential left-to-right decoding inherently limits inference throughput. Masked diffusion language models (dLLMs; Sahoo et al. 2024, Shi et al. 2024) have recently emerged as a competitive alternative: starting from a fully masked sequence, dLLMs progressively unmask tokens in arbitrary order, enabling parallel decoding within each step. Recent work has scaled dLLMs to billions of parameters [Nie et al., 2025, Ye et al., 2025, Bie et al., 2025, Bethune et al., 2026] and shown that they match AR models on a range of downstream tasks while offering the prospect of substantially more efficient inference [Wu et al., 2025, Ben-Hamu et al., 2025, Kim et al., 2025b, Jazbec et al., 2025].

A well-known limitation of masked dLLMs is that, once a token has been unmasked, the model has no mechanism to revise it [Wang et al., 2025, von Rütte, 2026], even if, later in the generation process and with more context, another token would be a better fit. This stems directly from the absorbing forward process used during training; it makes masked dLLMs susceptible to error propagation and limits their overall performance. To address this, a large body of work has recently attempted to equip masked dLLMs with self-correcting capabilities, either through fine-tuning or continual pre-training

*Equal contribution. Corresponding authors: <stipe.frkovic@student.uva.nl, m.jazbec@uva.nl>

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

[Kim et al., 2025a, Schiff et al., 2026, Chen et al., 2026, Yu et al., 2026, Huang et al., 2025], or post-hoc (training-free) methods [Hong et al., 2026, Dong et al., 2025, Xiang et al., 2026]. The latter are particularly appealing as they can be applied off-the-shelf to any pretrained dLLM. A representative example is WINO [Hong et al., 2026], which complements confidence-based adaptive unmasking (Fast-dLLM; Wu et al. 2025) with a confidence-based remasking step that reverts already-unmasked tokens deemed unlikely under their current surrounding context. Closely related are Saber [Dong et al., 2025], which remasks tokens whose confidence has dropped the most since the previous step, and COVER [Xiang et al., 2026], which keeps, remasks, or replaces unmasked tokens based on leave-one-out confidences approximated via a KV-cache override.

All three methods report gains in accuracy and speed. These gains, however, are reported at individual operating points rather than along an accuracy–compute frontier. WINO and COVER are also compared mainly against high-confidence sampling with a fixed number of tokens per step, rather than against a tuned Fast-dLLM. WINO and Saber are moreover evaluated mainly at large block lengths ($BL \geq 128$), where confidence-based unmasking is known to degrade [Nie et al., 2025]. Finally, all three are evaluated mainly under greedy decoding, although stochastic (i.e., non-greedy) decoding is central to RL post-training of dLLMs [Zhao et al., 2025, Tang et al., 2025] and to `pass@`k evaluation. It thus remains unclear how much of the reported gains comes from remasking itself. In this work, we therefore do not propose a new remasking method. Instead, we ask when training-free, confidence-based remasking actually helps in masked dLLMs.

To this end, we first compare the accuracy–compute Pareto frontiers of WINO, Saber, and COVER with those of Fast-dLLM under greedy decoding, across two dLLMs, four benchmarks, and two block lengths (Section 3.1). At the commonly used shorter block length ($BL = 32$), all three perform on par with Fast-dLLM, while additionally being slower in wall-clock time. Remasking gains appear only at $BL = 128$, where even WINO, the strongest of the three, still trails Fast-dLLM at $BL = 32$. Analyzing WINO in depth, we find that its remasking confidences, computed via auxiliary shadow tokens, closely match oracle leave-one-out confidences. However, most remasked positions are re-unmasked to the same token, so-called *flip-flops* [Xiang et al., 2026] (Section 3.2).

Moving to stochastic generation (Section 3.3), the picture becomes more nuanced. Under stochastic decoding, WINO mitigates some stochasticity-induced errors, improving `pass@1`, but aggravates the diversity collapse recently reported for confidence-based unmasking [Ni et al., 2026, Olausson et al., 2026]. On top of dUltra [Chen et al., 2025], a learned Bernoulli unmasking policy, WINO improves accuracy by $\sim$3.2% on average at only $\sim$2 extra NFEs, a larger gain than on top of Fast-dLLM.

Overall, the benefits of confidence-based remasking are highly setting-dependent. Our analyses further point to the dLLM, not the remasking rule, as the bottleneck, suggesting that effective remasking may require changes to training, not only to inference.

2 Background

We denote a sequence of tokens by $\boldsymbol{x} = (x^1, \ldots, x^L) \in \mathcal{V}^L$, where $\mathcal{V} := \{1, \ldots, V\}$ is the vocabulary and L is the sequence length. Given our focus on masked discrete diffusion, the vocabulary is extended with a special mask token denoted with `M`. Throughout, we use the notation $[L]$ to represent the set $\{1, \ldots, L\}$ for brevity. With *unmasking* we refer to $\texttt{M} \to x$ transitions and with *remasking* to $x \to \texttt{M}$ (where $x \in \mathcal{V} \setminus \{\texttt{M}\}$).

2.1 Masked diffusion language models

Masked diffusion models are a form of absorbing discrete diffusion [Austin et al., 2021a, Hoogeboom et al., 2021] whose noise distribution concentrates on a single absorbing state `M`. Under the continuous-time formulation of Campbell et al. [2022], the forward process independently replaces each token of $\boldsymbol{x}_0 \sim p_{\text{data}}$ with a mask token `M`:

$$p_{t|0}(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \prod_{k=1}^{L} \left(\alpha(t)\mathbf{1}[x_t^k = \texttt{M}] + (1 - \alpha(t))\mathbf{1}[x_t^k = x_0^k]\right)$$

where $\alpha(t)$ is a masking probability governed by a monotonically increasing schedule $\alpha : [0, 1] \to [0, 1]$ with $\alpha(0) = 0$ and $\alpha(1) = 1$. The reverse (generative) process is typically modeled by training

a network q_θ to approximate the true reverse conditional $p_{0|t}$ directly, since its ELBO admits the simple masked-prediction form [Sahoo et al., 2024, Shi et al., 2024, Ou et al., 2025]:

$$-\mathcal{L}(\theta) \triangleq \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_t}\left[\frac{1}{t}\sum_{k=1}^{L}\mathbf{1}[x_t^k = \mathtt{M}]\log q_\theta^k(x_0^k \mid \boldsymbol{x}_t)\right],$$

with $t \sim U[0,1]$, $\boldsymbol{x}_0 \sim p_{\text{data}}$, $\boldsymbol{x}_t \sim p_{t|0}$, and $q_\theta^k(\cdot \mid \boldsymbol{x}_t)$ denoting the model's predicted marginal at position k.

2.2 Unmasking in dLLMs

At test time, the reverse process is discretized and, starting from a fully masked sequence $\boldsymbol{x}_T := \mathtt{M}^L$, new samples are obtained via progressive *unmasking*. Representing the current generation as $\boldsymbol{x}_t \in \mathcal{V}^L$ with still-masked positions $\mathcal{M}_t := \{k \in [L] \mid x_t^k = \mathtt{M}\}$, the next-step generation $\boldsymbol{x}_{t-1}$ is sampled via

$$x_{t-1}^k := \begin{cases} x \sim q_\theta^k(\cdot \mid \boldsymbol{x}_t\,;\tau), & \text{if } k \in \mathcal{U}_t^K, \\ x_t^k\,, & \text{otherwise} \end{cases}$$

where the set $\mathcal{U}_t^K$ of $K \geq 1$ unmasking positions is sampled uniformly at random (without replacement) from $\mathcal{M}_t$ [Zheng et al., 2025, Nie et al., 2025] and the process is repeated until there are no masked positions left ($\mathcal{M}_t = \emptyset$). Here τ denotes a sampling temperature, with higher temperatures yielding more diverse generated samples. In dLLMs, it is common to set $\tau = 0$, which corresponds to greedy token selection.

In practice, confidence-based unmasking is often chosen over random unmasking due to better empirical performance. Two particularly popular instantiations are *high-confidence* sampling [Chang et al., 2022, Nie et al., 2025], where K positions with the highest token confidence[2] $c_t^k \triangleq \max_v q_\theta^k(v \mid \boldsymbol{x}_t)$ are unmasked

$$\mathcal{U}_t^K := \underset{I \subseteq \mathcal{M}_t,\ |I|=K}{\arg\max} \sum_{k \in I} c_t^k \tag{1}$$

and *Fast-dLLM* [Wu et al., 2025], where all positions with token confidence larger than a prespecified threshold λ_1 are unmasked

$$\mathcal{U}_t^\lambda := \{k \in \mathcal{M}_t \mid c_t^k > \lambda_1\} \cup \{\arg\max_{k \in \mathcal{M}_t} c_t^k\} \tag{2}$$

with a fallback to the position with the highest confidence to ensure progress is made in every sampling step. Due to its adaptive nature, Fast-dLLM was empirically shown to push the frontier of parallel-token generation in dLLMs [Wu et al., 2025, Jazbec et al., 2025].

Additionally, confidence-based samplers are often combined with semi-autoregressive (semi-AR; Arriola et al. 2025) sampling, which restricts unmasking to a contiguous block of $BL < L$ tokens at a time and proceeds block-by-block in a left-to-right order (i.e., once all positions in the current block are unmasked, generation moves to the next block). Empirically, semi-AR sampling has proved essential for confidence-based samplers, as using too large a block length BL leads to compromised performance [Nie et al., 2025].

Algorithm 1 WINO sampling algorithm [Hong et al., 2026]. Differences to Fast-dLLM [Wu et al., 2025] unmasking are denoted in blue.

Input: $\boldsymbol{x}_t$, b_t, q_θ, τ, λ_1, λ_2
Output: $\boldsymbol{x}_{t-1}, b_{t-1}$

$\tilde{\boldsymbol{x}}_t \leftarrow [\boldsymbol{x}_t, \mathtt{M}, \ldots, \mathtt{M}] \in \mathcal{V}^{L+BL}$
$\mathcal{B} \leftarrow [b_t \cdot BL : (b_t+1) \cdot BL]$
$\mathcal{M}_t \leftarrow \{k \in [L] \mid x_t^k = \mathtt{M}\}$

$q_t^k \leftarrow q_\theta^k(\cdot \mid \tilde{\boldsymbol{x}}_t)$
$x_0^k \sim q_t^k(\cdot\,;\tau)$
$c_t^k \leftarrow q_t^k(x_0^k)$
$\mathcal{U}_t \leftarrow \{k \in \mathcal{M}_t \cap \mathcal{B} \mid c_t^k > \lambda_1\}$
$\mathcal{U}_t \leftarrow \mathcal{U}_t \cup \{\arg\max_{k \in \mathcal{M}_t \cap \mathcal{B}} c_t^k\}$

$\tilde{c}_t^k \leftarrow q_t^{k_s}(x_t^k), k_s = L + (k - b_t \cdot BL)$
$\mathcal{R}_t \leftarrow \{k \in \mathcal{B} \setminus \mathcal{M}_t \mid \tilde{c}_t^k < \lambda_2\}$
$\mathcal{R}_t \leftarrow \texttt{Loop-guard}(\mathcal{R}_t, \mathcal{U}_{t+1})$

$$x_{t-1}^k := \begin{cases} x_0^k\,, & k \in \mathcal{U}_t \\ \mathtt{M}\,, & k \in \mathcal{R}_t \\ x_t^k\,, & \text{else} \end{cases}$$

$$b_{t-1} := \begin{cases} b_t + 1\,, & \{k \in \mathcal{B} | x_{t-1}^k = \mathtt{M}\} = \emptyset \\ b_t\,, & \text{else} \end{cases}$$

[2] For stochastic sampling, the confidence is defined instead as $c_t^k \triangleq q_\theta^k(x \mid \boldsymbol{x}_t),\ x \sim q_\theta^k(\cdot \mid \boldsymbol{x}_t\,;\tau)$.

Recently, learned Bernoulli unmasking policies have gained traction as a scalable alternative to confidence-based heuristics like Fast-dLLM [Jazbec et al., 2025, Chen et al., 2025]. *dUltra* [Chen et al., 2025], for example, jointly trains an unmasking head during RL post-training that predicts the unmasking probability of each still-masked position at every sampling step:

$$\mathcal{U}_t := \{k \in \mathcal{M}_t \mid b_t^k = 1\}, \quad b_t^k \sim \mathrm{Ber}(s_t^k), \quad s_t^k = \sigma\Big(f_\phi(\boldsymbol{h}_t^{1:L})_k\Big)$$

where f_ϕ is the unmasking head, $\boldsymbol{h}_t^{1:L}$ denotes the dLLM's last-layer hidden states (before the unembedding matrix), and σ is the sigmoid function. Due to the Bernoulli sampling of unmasking decisions, dUltra gives rise to stochastic generation even at $\tau = 0$.

2.3 Confidence-based remasking in dLLMs

In masked dLLMs, once a position has been unmasked, the model has no option to revise it, even if, at a later time step and with additional context, another token would be a better fit. This makes masked models susceptible to early sampling mistakes from which they cannot recover [Wang et al., 2025]. Note that the inability to remask stems from the absorbing (towards `M`) forward process used during training. To overcome this limitation, a growing body of literature has tried to add self-correcting capabilities to masked dLLMs (Hong et al. 2026, Kim et al. 2025a, Huang et al. 2025, Schiff et al. 2026; *inter alia*). While most approaches require some form of retraining or fine-tuning, we focus here on methods that aim to achieve remasking in a post-hoc (training-free) manner.

A representative confidence-based remasking approach is **WINO** [Hong et al., 2026], where the idea of confidence thresholding unmasking [Wu et al., 2025] is applied to remasking. To get informative remasking confidences efficiently, WINO attaches to the current generation $\boldsymbol{x}_t$ an auxiliary block of so-called *shadow* tokens $\tilde{\boldsymbol{x}}_t = [\boldsymbol{x}_t, \mathtt{M}, \ldots, \mathtt{M}] \in \mathcal{V}^{L+BL}$, one per position in the active block b_t. Crucially, each shadow position has the same positional embedding as its corresponding position in the original block but is prevented from attending to the (unmasked) token at the original position via a custom attention mask (see Figure 6). This ensures that the shadow output approximates a re-prediction (starting from `M` instead of the current token x_t^k) of that position given the surrounding context without the cost of an additional forward pass:[3]

$$q_\theta^{k_s}(\cdot \mid \tilde{\boldsymbol{x}}_t) \approx q_\theta^k(\cdot \mid \boldsymbol{x}_{t,-k}) \tag{3}$$

where $k_s = L + (k - b_t \cdot BL)$ is the shadow position corresponding to the k-th original position and $\boldsymbol{x}_{t,-k}$ corresponds to the current generation $\boldsymbol{x}_t$ with the k-th token masked out (i.e., leave-one-out style prediction). Note that original positions do not attend to shadow positions, hence their predictive distribution is not affected by the presence of the shadow block: $q_\theta^k(\cdot \mid \tilde{\boldsymbol{x}}_t) = q_\theta^k(\cdot \mid \boldsymbol{x}_t)$ for $k \leq L$.

All presently unmasked tokens in the active block whose shadow confidence falls below a fixed threshold λ_2 are then remasked

$$\mathcal{R}_t := \{k \in \mathcal{B} \setminus \mathcal{M}_t \mid q_\theta^{k_s}(x_t^k \mid \tilde{\boldsymbol{x}}_t) < \lambda_2\} \tag{4}$$

where $\mathcal{B} := [b_t \cdot BL : (b_t + 1) \cdot BL]$ denotes the set of indices in the active block b_t; see also Algorithm 1. The intuition is that if the current token x_t^k would not likely be predicted under the same surrounding context (captured by the shadow predictive distribution $q_\theta^{k_s}$), the position is set to `M` again so that the model can predict a new token for it in future sampling steps under a new context.

Note that once remasking is allowed, convergence ($\mathcal{M}_t = \emptyset$) is no longer guaranteed: sampling can enter a loop in which a given set of positions is repeatedly unmasked to the same tokens after being remasked. WINO's implementation[4] addresses this with a *loop-guard* mechanism: whenever too many tokens get selected for remasking via confidence thresholding, i.e., $|\mathcal{R}_t| \geq |\mathcal{U}_{t+1}|$, only $K = |\mathcal{U}_{t+1}| - 1$ positions with the smallest shadow confidences are remasked, guaranteeing net progress at every step.

We additionally evaluate two other confidence-based remasking methods. **Saber** [Dong et al., 2025] tracks the dLLM's own (*native*) confidence c_t^k at each unmasked position and remasks the positions whose confidence dropped the most since the previous step. **COVER** [Xiang et al., 2026] approximates leave-one-out confidences $q_\theta^k(\cdot \mid \boldsymbol{x}_{t,-k})$ by overriding the KV cache and is designed to reduce flip-flops.

[3] The approximation is exact for a single-layer transformer; for deeper models, the surrounding tokens' hidden state representations still encode information about x_t^k via self-attention.

[4] `https://github.com/Feng-Hong/WINO-DLLM/blob/main/LLaDA/decoding.py`

3 Re-evaluating confidence-based remasking

We now ask whether confidence-based remasking improves the performance of masked dLLMs. We take the evaluation in the WINO paper [Hong et al., 2026] as a starting point, as it leaves three questions open: i) its main baseline is high-confidence sampling (Equation 1), leaving it unclear whether WINO's gains come from adaptive unmasking (Fast-dLLM) or from remasking itself; ii) it evaluates mainly at large block lengths ($BL \geq 128$ instead of the standard $BL = 32$), where confidence-based unmasking is known to struggle [Nie et al., 2025]; and iii) it studies remasking primarily under greedy decoding ($\tau = 0$), leaving open how remasking performs under stochastic decoding. We address i) and ii) in Section 3.1, where we also include Saber and COVER, analyze WINO's limited gains in Section 3.2, and address iii) in Section 3.3.

Experimental setup We evaluate on two open-source dLLMs, LLaDA-8B-Instruct [Nie et al., 2025] and Dream-v0-Instruct-7B [Ye et al., 2025], and four standard benchmarks: GSM8K [Cobbe et al., 2021], MATH-500 [Hendrycks et al., 2021], HumanEval [Chen et al., 2021], and MBPP [Austin et al., 2021b]. All generations are zero-shot, with a generation length of 256 and, unless stated otherwise, greedy decoding ($\tau = 0$). We trace the Pareto frontiers of Fast-dLLM, WINO, and COVER by varying the unmasking threshold $\lambda_1 \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, and that of Saber, which sets its unmasking threshold adaptively, by varying its remasking budget $\mu \in \{2, 4, 8\}$. For WINO, we set the remasking threshold $\lambda_2 = 0.8$ (other values in Figure 11); for Saber and COVER, we use the official implementations with the default values of all other hyperparameters. To study the role of block length, we vary $BL \in \{32, 128\}$: $BL = 32$ is the current standard setting for LLaDA-style models [Ni et al., 2026, Wu et al., 2025], while $BL = 128$ is the primary setting reported in the WINO paper.

3.1 Confidence-based remasking vs. unmasking alone

We first test whether confidence-based remasking (WINO, Saber, COVER) brings meaningful gains over confidence-threshold unmasking alone (Fast-dLLM; Wu et al. 2025).

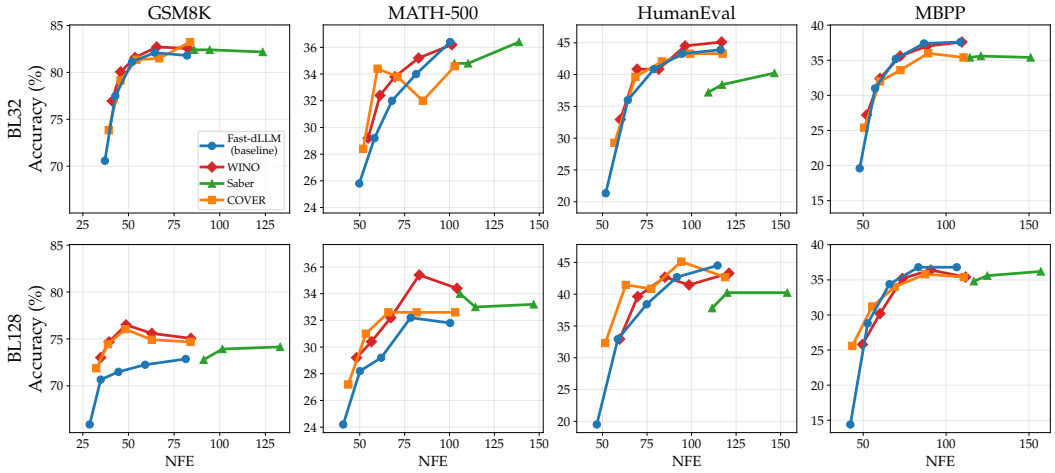


Figure 1: Pareto frontiers of accuracy vs. network function evaluations (NFEs) for remasking methods (WINO, Saber, COVER) and Fast-dLLM on LLaDA-8B-Instruct under greedy decoding ($\tau = 0$), with block lengths $BL = 32$ (top) and $BL = 128$ (bottom); sweep details in Section 3. At $BL = 32$, no remasking method meaningfully improves over Fast-dLLM. At $BL = 128$, WINO gains more than Saber and COVER, but still underperforms Fast-dLLM at $BL = 32$. Results for Dream-v0-Instruct-7B are shown in Figure 7.

Starting with $BL = 32$ (Figure 1), remasking sometimes improves performance (e.g., WINO on LLaDA's HumanEval), but the gains are generally marginal and inconsistent across datasets and along the Pareto frontier. Averaged over the four datasets, WINO improves over Fast-dLLM's top performance by $\sim 0.4\%$ on LLaDA and $\sim 0.5\%$ on Dream (per-dataset values in Table 1), differences that are small relative to the binomial standard errors (Figure 8). Saber changes top performance by

Table 1: Difference in top accuracy, Δ (%), between each remasking method and Fast-dLLM, each evaluated at its optimal point along its Pareto frontier (setup of Figures 1 and 7).

		$BL = 32$				$BL = 128$			
Model	Method	GSM8K	MATH	HEval	MBPP	GSM8K	MATH	HEval	MBPP
LLaDA	WINO	$+0.60$	-0.20	$+1.22$	$+0.00$	$+3.64$	$+3.20$	-1.22	-0.40
	Saber	$+0.30$	$+0.00$	-3.66	-2.00	$+1.29$	$+1.80$	-4.27	-0.60
	COVER	$+1.13$	-1.80	-0.61	-1.60	$+3.18$	$+0.40$	$+0.61$	-1.00
Dream	WINO	$+0.38$	$+0.60$	$+1.22$	$+0.00$	$+0.45$	$+1.60$	$+1.83$	$+2.00$
	Saber	$+1.29$	$+0.00$	-1.22	$+0.00$	$+0.68$	$+0.20$	-2.44	$+1.00$
	COVER	$+0.68$	$+1.80$	$+1.22$	-0.60	-10.39	-5.80	-4.88	-1.60

$\sim -1.3\%$ and $\sim 0.0\%$, and COVER by $\sim -0.7\%$ and $\sim 0.8\%$ (Figures 1 and 7). Since each WINO forward pass extends the sequence with a shadow block, a WINO step is more expensive than a Fast-dLLM step, in both FLOPs and latency. We therefore also compare wall-clock time (Figure 9): on most datasets, all three methods reach their top performance later than Fast-dLLM.

We next turn to $BL = 128$, the main setting of the WINO paper [Hong et al., 2026]. Here WINO yields larger gains over Fast-dLLM (Figure 1), improving top performance by $\sim 1.3\%$ on LLaDA and $\sim 1.5\%$ on Dream on average. In wall-clock time (Figure 9), however, these gains become less consistent, as the cost of its shadow block grows with the block length: at the same unmasking threshold, WINO takes up to $1.7\times$ as long as Fast-dLLM, compared to up to $1.3\times$ at $BL = 32$. On LLaDA's HumanEval and MBPP, Fast-dLLM is even ahead along most of the frontier. Saber and COVER do not consistently share these gains: Saber slightly underperforms Fast-dLLM on both models, and COVER improves by $\sim 0.8\%$ on LLaDA but drops by $\sim 5.7\%$ on Dream (Figure 7); its original paper evaluates Dream only at $BL = 32$.

One possible reason for WINO's gains is that a larger block length gives it more opportunity to correct mistakes over a longer range. However, masked dLLMs are known to degrade under confidence-based samplers as block length grows, since corrupted `EOS` confidences produce degenerate generation orders [Jazbec et al., 2025]. Since WINO at $BL = 128$ remains below Fast-dLLM at $BL = 32$, both in top accuracy and at almost every wall-clock time budget, remasking appears to repair this failure mode rather than raise the underlying dLLM's performance ceiling. Together with its marginal gains at $BL = 32$, this suggests that WINO's improvements may not justify its added computational overhead and implementation complexity.

Takeaway 1. At the commonly used $BL = 32$, none of the three confidence-based remasking methods meaningfully improves over Fast-dLLM, while all add wall-clock overhead. At $BL = 128$, WINO mitigates the degradation of confidence-based unmasking but does not raise the performance ceiling of masked dLLMs, while Saber and COVER show no consistent gains.

3.2 Understanding WINO's limited gains at short block lengths

We now ask why WINO, the strongest of the three methods, gains little at the standard $BL = 32$, and whether the remasking rule or the underlying dLLM is the limiting factor.

Shadow-token approximation We first test whether WINO's design choices limit its gains. Ablating its remasking criterion, loop-guard, shadow attention mask, and max-accept cap does not yield meaningful gains (Appendix B.1), ruling out these choices as the cause. We therefore turn to its core component, the shadow-token approximation of the remasking confidence (Equation 3). Recall that, ideally, remasking would be based on the oracle leave-one-out confidence $q_\theta^k(\cdot \mid \boldsymbol{x}_{t,-k})$. However, computing such oracle confidences requires an additional forward pass for every candidate remasking position at each sampling step. We nevertheless run a WINO variant that remasks based on these oracle confidences (*WINO (Oracle*)*) as a reference. On LLaDA's HumanEval (Figure 2), WINO's shadow confidences match the performance of *WINO (Oracle*)* while being much cheaper (the oracle approach requires $\sim 14\times$ as many NFEs). We also compare top-1 consistency between shadow and oracle predictive distributions in Figure 12, finding it to be above 96%, further validating the

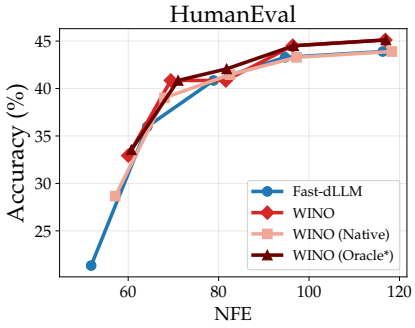


Figure 2: Quality of WINO's shadow-token confidence as an approximation (Equation 3) to the oracle leave-one-out confidence (LLaDA-8B-Instruct, HumanEval, $BL = 32$, $\lambda_2 = 0.8$). *WINO* uses the shadow prediction $q_\theta^{k_s}(\cdot \mid \tilde{\boldsymbol{x}}_t)$; *WINO (Oracle*)* uses the leave-one-out prediction $q_\theta^k(\cdot \mid \boldsymbol{x}_{t,-k})$, incurring one extra forward pass per candidate position; *WINO (Native)* reuses the dLLM's native confidence $q_\theta^k(\cdot \mid \boldsymbol{x}_t)$ at the already-unmasked position. Shadow confidences match oracle performance at a fraction of the cost, whereas the native confidence yields no gain over unmasking alone. *For the oracle, the x-axis shows sampling steps; its actual NFEs are $\sim 14\times$ larger.

shadow-token approximation. Moreover, a WINO variant that remasks based on the dLLM's native confidence $q_\theta^k(\cdot \mid \boldsymbol{x}_t)$ at the unmasked position (*WINO (Native)*) fails to improve over unmasking alone, confirming the necessity of shadow tokens.

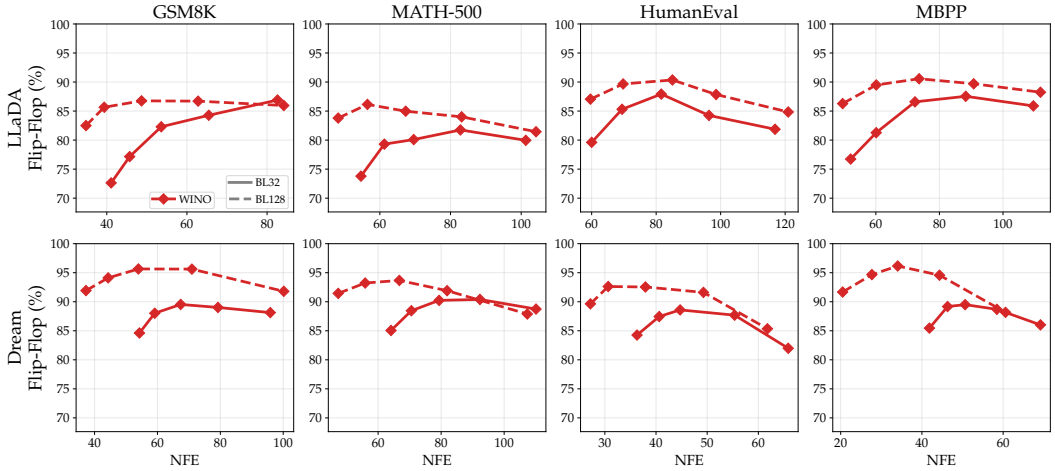


Figure 3: Flip-flop rate along the Pareto frontiers of WINO ($\lambda_2 = 0.8$), defined as the fraction of remasking events in which the position is re-unmasked to the same token [Xiang et al., 2026] (solid: $BL = 32$, dashed: $BL = 128$). $\sim$75% to $\sim$90% of remasked positions on LLaDA-8B-Instruct and $\sim$80% to $\sim$95% on Dream-v0-Instruct-7B are restored to their original token.

Flip-flops Since the shadow-token approximation does not explain WINO's limited gains, we next measure the *flip-flop* rate of WINO, following Xiang et al. [2026]. Concretely, the flip-flop rate is defined as the proportion of remasks that result in no token change, i.e., a position is remasked and then re-unmasked to the same token. As shown in Figure 3, flip-flops occur at high rates for WINO across the entire Pareto frontier: on average, $\sim$75% to $\sim$90% (LLaDA) and $\sim$80% to $\sim$95% (Dream) of remasked positions end up with the same token. Combined with the shadow-token result above, this suggests that WINO *identifies* the same positions as the oracle confidence; at those positions, however, the dLLM rarely *proposes* a different token. Moreover, flip-flop rates are nearly identical for generations with correct and incorrect final answers (Appendix B.3), showing that flip-flops are not specific to failed generations.

Cascading dependencies One possible explanation for high flip-flop rates is that tokens unmasked later were conditioned on x_t^k and still encode it after position k is remasked, pinning the prediction back to x_t^k. To test this, we additionally remask the temporal or spatial neighbors of each flagged position, i.e., tokens unmasked at the same or the next sampling step, or its left and right neighbors (Appendix B.3). Surprisingly, both variants match or degrade accuracy relative to remasking only the flagged position. They also do not reduce the flip-flop rate, and often increase it. Sampling remasked positions stochastically ($\tau = 1$) instead of greedily also does not help, as it only slightly lowers

flip-flop rates and does not improve accuracy (Appendix B.3). This indicates that the bottleneck is neither cascading dependencies nor greedy re-prediction, but the dLLM's predictive distribution at position k itself, which strongly favors the original token.

> **Takeaway 2.** WINO's shadow-token confidence closely approximates the oracle leave-one-out remasking confidence. At the flagged positions, however, the dLLM mostly re-proposes the original token, even when neighboring tokens are also remasked or the token is sampled stochastically. This points to the dLLM, rather than the remasking rule, as the bottleneck.

3.3 Remasking under stochastic generation

Lastly, we study whether confidence-based remasking is beneficial under stochastic generation. Although this setting is largely absent from the current literature on remasking in dLLMs, we believe it is particularly relevant for remasking approaches, as the increased stochasticity may produce more erroneous tokens that could benefit from revision. We consider two variants of stochastic generation: i) stochastic decoding ($\tau > 0$) with deterministic unmasking (Fast-dLLM) and ii) greedy decoding with stochastic unmasking (via learned Bernoulli policies; Jazbec et al. 2025, Chen et al. 2025).

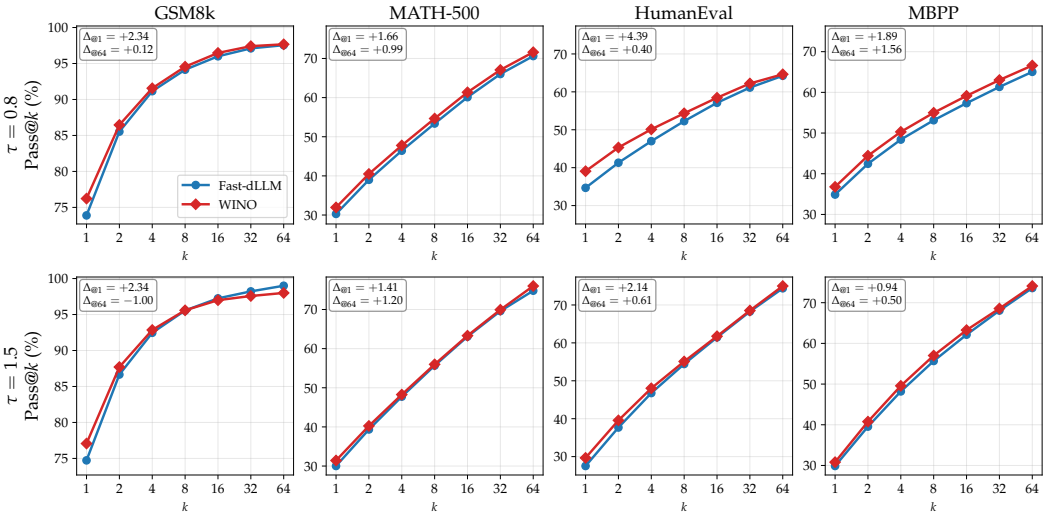


Figure 4: pass@k for WINO and Fast-dLLM under stochastic decoding (LLaDA-8B-Instruct, $BL = 32$, $\lambda_1 = 0.6$, $\lambda_2 = 0.8$) at temperatures $\tau \in \{0.8, 1.5\}$. WINO improves pass@1 by $\sim$2.6% on average at $\tau = 0.8$, but its advantage diminishes as k grows, shrinking to $\sim$0.7% on average at pass@64. This suggests that remasking further constrains generation diversity on top of the diversity collapse already documented for confidence-based unmasking [Ni et al., 2026].

pass@k scaling We compare Fast-dLLM ($\lambda_1 = 0.6$) and WINO ($\lambda_1 = 0.6$) on LLaDA-8B-Instruct with $BL = 32$ under stochastic decoding. Concretely, we consider $\tau = 0.8$, a standard temperature in RL post-training of dLLMs [Zhao et al., 2025, Tang et al., 2025], as well as $\tau = 1.5$, motivated by the observation that dLLMs benefit from higher sampling temperatures than AR LLMs [Ni et al., 2026]. To measure performance, we report the commonly used pass@k metric[5], which tracks the probability that at least one of k generated solutions is correct. As shown in Figure 4, remasking improves pass@1 by $\sim$2.6% on average across the four datasets at $\tau = 0.8$. However, remasking also appears to hurt generation diversity: WINO scales worse than Fast-dLLM as k grows, with gains shrinking to $\sim$0.7% at pass@64 for $\tau = 0.8$. This suggests that confidence-based remasking may further exacerbate the diversity collapse previously observed for confidence-based unmasking [Ni et al., 2026, Olausson et al., 2026], where greedily relying on confidence to decide which positions to sample constrains exploration of the solution space. Consistent with this, WINO lowers the mean

[5]Following Chen et al. [2021], we report the unbiased estimator pass@$k := \mathbb{E}\left[1 - \binom{n-c}{k} / \binom{n}{k}\right]$, where $n \geq k$ is the number of samples generated per problem and c is the number of those that are correct.

token entropy compared to Fast-dLLM (0.61 vs. 0.64 on MATH-500), in part because remasked positions are re-committed with much lower entropy (1.15 → 0.43; Table 3). At $\tau = 1.5$, WINO's improvements shrink further, to $\sim$1.6% and $\sim$0.3% for `pass@1` and `pass@64`, respectively.

> **Takeaway 3.** Under stochastic decoding, WINO improves `pass@1` but exacerbates the diversity collapse of confidence-based unmasking, with gains shrinking as the number of samples k grows.

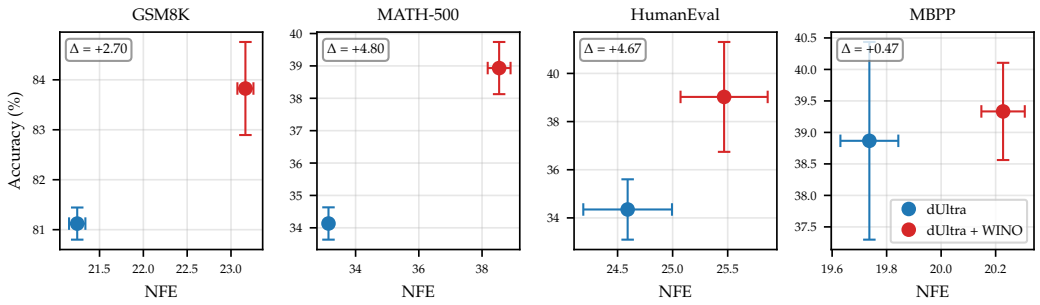


Figure 5: Accuracy and NFEs for dUltra with and without WINO remasking on LLaDA-8B-Instruct ($BL = 32$, $\lambda_2 = 0.8$, $\tau = 0$); means and error bars are over 3 seeds. Adding WINO remasking on top of dUltra's Bernoulli unmasker yields an average accuracy gain of $\sim$3.2% across the four benchmarks at a cost of only $\sim$2 extra NFEs, suggesting that the value of remasking depends on the unmasking strategy.

Bernoulli unmasking We next keep greedy token selection ($\tau = 0$) but make the unmasking decisions stochastic, using the learned Bernoulli unmasking policy of dUltra (Section 2.2). Results comparing dUltra with and without WINO remasking at $BL = 32$ are shown in Figure 5; we report average accuracy and NFEs on test questions over 3 random seeds (with error bars). Adding WINO remasking on top of dUltra improves accuracy by $\sim$3.2% on average across datasets, while increasing NFEs by only $\sim$2 on average. This highlights the potential of confidence-based remasking when paired with stochastic unmasking strategies: it appears effective at correcting low-quality tokens introduced by the randomness of Bernoulli unmasking. Moreover, dUltra produces faster generations than Fast-dLLM (e.g., on GSM8K, it reaches $\sim$80% accuracy in only $\sim$20 NFEs, versus $\sim$50 for Fast-dLLM; see Figure 1), which may further explain WINO's stronger gains here: more aggressive generation yields more tokens in need of revision.

> **Takeaway 4.** When paired with stochastic Bernoulli unmasking, WINO remasking is more effective than with deterministic Fast-dLLM unmasking, suggesting that the added value of confidence-based remasking depends on the unmasking strategy.

4 Related work

Remasking in dLLMs Several recent works explore remasking in masked dLLMs. One direction focuses on train-time interventions: Kim et al. [2025a] fine-tune dLLMs for provable self-correction, Schiff et al. [2026] train models to identify and revise their own errors, Liu et al. [2026] train a separate correction head on top of a frozen dLLM, Bie et al. [2026] jointly train mask-to-token and token-to-token editing objectives at scale, Yu et al. [2026] train dLLMs to verify their own previously generated tokens, and Huang et al. [2025] use post-training to learn when to remask. A related direction is to move away from purely absorbing (masked) diffusion to a uniform one that supports remasking 'naturally': Von Rütte et al. [2025] train hybrid mask+uniform diffusion models, Wang et al. [2026] simplify this with purely uniform transitions, and Chen et al. [2026] extend pretrained masked dLLMs into self-correcting uniform diffusion models. Our work focuses on a complementary, training-free direction.

Post-hoc remasking Several recent methods enable remasking without retraining, typically by applying a confidence-based criterion at inference time. WINO [Hong et al., 2026], the focus of

our re-evaluation, computes remasking confidences via auxiliary shadow tokens that approximate a leave-one-out re-prediction without an additional forward pass. ReMDM [Wang et al., 2025] introduces a remasking sampler that can be applied to pretrained masked diffusion models to enable revisions during generation. Saber [Dong et al., 2025] remasks the unmasked positions whose confidence dropped the most since the previous step. COVER [Xiang et al., 2026] documents the high *flip-flop* rates of confidence-based remasking and reduces them by approximating leave-one-out confidences via a KV-cache override. While these works each report gains in accuracy or speed over static confidence-based unmasking (Equation 1), how much these gains depend on the evaluation setting, such as block length, greedy vs. stochastic decoding, and unmasking strategy, has not been systematically examined, a gap our work aims to fill.

5 Conclusion

To better understand the added value of confidence-based remasking in masked dLLMs, we re-evaluated three recently proposed training-free methods: WINO, Saber, and COVER. Our findings suggest that their benefits depend on the setting. Under greedy decoding at the standard $BL = 32$, all three perform on par with unmasking alone (Fast-dLLM), especially once their wall-clock overhead is accounted for; gains appear only at $BL = 128$, where remasking mitigates the degradation of confidence-based unmasking. Under stochastic decoding, WINO improves `pass@1` at the cost of generation diversity, and it yields larger gains when paired with learned Bernoulli unmasking. Our analyses further point to the dLLM, not the remasking rule, as the bottleneck. Since the impact of remasking is highly setting-dependent, we hope that future work adopts the evaluation framework outlined here to better understand the limits and potential of confidence-based remasking in masked dLLMs.

Limitations and future work While we evaluate three confidence-based remasking methods (WINO, Saber, COVER) on two dLLMs, our failure-mode analyses and the stochastic decoding experiments focus on WINO; future work should apply these analyses to other post-hoc remasking methods, including ReMDM [Wang et al., 2025]. Our results suggest that effective remasking in masked dLLMs likely requires either some form of retraining or a shift to uniform discrete diffusion [von Rütte, 2026, Chen et al., 2026], possibly because masked diffusion training induces a strong "unmask-only" prior ($\texttt{M} \to x$) that post-hoc remasking cannot easily override. Moreover, while we deliberately focused on post-hoc remasking, it would be worthwhile to explore whether the negative impact of remasking on sampling diversity (Figure 4) also holds for approaches that incorporate remasking via retraining or fine-tuning [Schiff et al., 2026, Chen et al., 2026].

Impact statement This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments and Disclosure of Funding

We thank Theo X. Olausson, the Apple MLR Paris team, Guoxuan Xia, and Izabela Kurek for helpful discussions. This project was generously supported by the Bosch Center for Artificial Intelligence. Eric Nalisnick did not utilize resources from Johns Hopkins University for this project.

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. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993. Curran Associates, Inc., 2021a. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/958c530554f78bcd8e97125b70e6973d-Paper.pdf`.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021b. URL https://arxiv.org/abs/2108.07732.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *arXiv preprint arXiv:2505.24857*, 2025.

Louis Bethune, Victor Turrisi, Bruno Kacper Mlodozeniec, Pau Rodriguez Lopez, Lokesh Boominathan, Nikhil Bhendawade, Amitis Shidani, Joris Pelemans, Theo X. Olausson, Devon Hjelm, Paul Dixon, Joao Monteiro, Pierre Ablin, Vishnu Banna, Arno Blaas, Nick Henderson, Kari Noriy, Dan Busbridge, Josh Susskind, Marco Cuturi, Irina Belousova, Luca Zappella, Russ Webb, and Jason Ramapuram. The design space of tri-modal masked diffusion models, 2026. URL https://arxiv.org/abs/2602.21472.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, et al. Llada2. 0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, et al. Llada2. 1: Speeding up text diffusion via token editing. *arXiv preprint arXiv:2602.08676*, 2026.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 28266–28279. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b5b528767aa35f5b1a60fe0aaeca0563-Paper-Conference.pdf.

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*, pages 11315–11325, 2022.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Shirui Chen, Jiantao Jiao, Lillian J Ratliff, and Banghua Zhu. dultra: Ultra-fast diffusion language models via reinforcement learning. *arXiv preprint arXiv:2512.21446*, 2025.

Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. Dmax: Aggressive parallel decoding for dllms. *arXiv preprint arXiv:2604.08302*, 2026.

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.

Yihong Dong, Zhaoyu Ma, Xue Jiang, Zhiyuan Fan, Jiaru Qian, Yongmin Li, Jianha Xiao, Zhi Jin, Rongyu Cao, Binhua Li, et al. Saber: An efficient sampling with adaptive acceleration and backtracking enhanced remasking for diffusion language model. *arXiv preprint arXiv:2510.18165*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Feng Hong, Geng Yu, Yushi Ye, Haicheng Huang, Huangjie Zheng, Ya Zhang, Yanfeng Wang, and Jiangchao Yao. Wide-in, narrow-out: Revokable decoding for efficient and effective DLLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XtLQHlNLxy.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12454–12465. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/67d96d458abdef21792e6d8e590244e7-Paper.pdf.

Zemin Huang, Yuhang Wang, Zhiyang Chen, and Guo-Jun Qi. Don't settle too early: Self-reflective remasking for diffusion language models. *arXiv preprint arXiv:2509.23653*, 2025.

Metod Jazbec, Theo X Olausson, Louis Béthune, Pierre Ablin, Michael Kirchhof, João Monteiro, Victor Turrisi, Jason Ramapuram, and Marco Cuturi. Learning unmasking policies for diffusion language models. *arXiv preprint arXiv:2512.09106*, 2025.

Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z Pan, Hyeji Kim, Sham Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction. *arXiv preprint arXiv:2510.01384*, 2025a.

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. *arXiv preprint arXiv:2502.06768*, 2025b.

Liming Liu, Binxuan Huang, Xin Liu, Bing Yin, and Tuo Zhao. Backplay: Head-only look-back self-correction for diffusion language models. *arXiv preprint arXiv:2601.06428*, 2026.

Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, et al. The flexibility trap: Why arbitrary order limits reasoning potential in diffusion language models. *arXiv preprint arXiv:2601.15165*, 2026.

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. *arXiv preprint arXiv:2502.09992*, 2025.

Theo X Olausson, Metod Jazbec, Xi Wang, Armando Solar-Lezama, Christian A Naesseth, Stephan Mandt, and Eric Nalisnick. A tale of two temperatures: Simple, efficient, and diverse sampling from diffusion language models. *arXiv preprint arXiv:2604.09921*, 2026.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 64972–65009, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a365e37c18fb91af547a2f0012a89e98-Paper-Conference.pdf.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *ICML 2024 Workshop on Efficient and Accessible Foundation Models for Biological Discovery*, 2024. URL https://openreview.net/forum?id=DdU9gP4EXW.

Yair Schiff, Omer Belhasin, Roy Uziel, Guanghan Wang, Marianne Arriola, Gilad Turok, Michael Elad, and Volodymyr Kuleshov. Learn from your mistakes: Self-correcting masked diffusion models. *arXiv preprint arXiv:2602.11590*, 2026.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Xiaohang Tang, Rares Dolga, Sangwoong Yoon, and Ilija Bogunovic. wd1: Weighted policy optimization for reasoning in diffusion language models. *arXiv preprint arXiv:2507.08838*, 2025.

Dimitri Von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. *arXiv preprint arXiv:2503.04482*, 2025.

Dimitri von Rütte. Why diffusion language models are the future, 2026. URL https://dimitri.ml/posts/why-diffusion-language-models-are-the-future/.

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

Linxuan Wang, Ziyi Wang, Yikun Bai, Wei Deng, Guang Lin, and Qifan Song. Generalized discrete diffusion with self-correction. *arXiv preprint arXiv:2603.02230*, 2026.

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, 2025. URL https://arxiv.org/abs/2505.22618.

Yanzheng Xiang, Lan Wei, Yizhen Yao, Qinglin Zhu, Hanqi Yan, Chen Jin, Philip Alexander Teare, Dandan Zhang, Lin Gui, Amrutha Saseendran, et al. Stop the flip-flop: Context-preserving verification for fast revocable diffusion decoding. *arXiv preprint arXiv:2602.06161*, 2026.

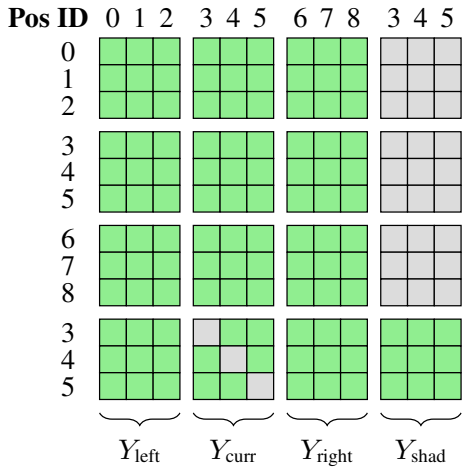
Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Yifan Yu, Yuqing Jian, Junxiong Wang, Zhongzhu Zhou, Donglin Zhuang, Xinyu Fang, Sri Yanamandra, Xiaoxia Wu, Qingyang Wu, Shuaiwen Leon Song, et al. Introspective diffusion language models. *arXiv preprint arXiv:2604.11035*, 2026.

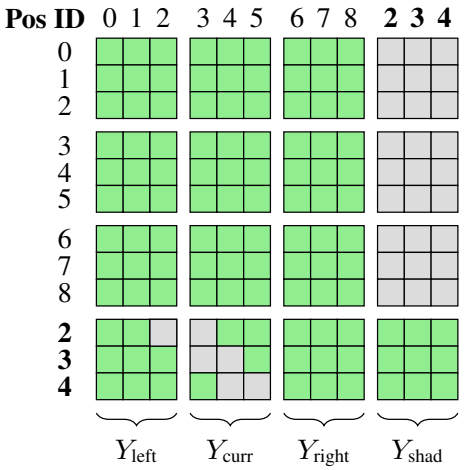
Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*, 2025.

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, April 2025.

A Additional figures



(a) WINO attention mask and position IDs for LLaDA-8B [Nie et al., 2025].



(b) Adaptation for Dream-7B model [Ye et al., 2025].

Figure 6: WINO attention masks and position IDs: LLaDA (a) vs. our adaptation for Dream (b). Note that adaptation is needed since Dream was converted into a dLLM starting from an AR model. Bold indices in (b) highlight the modified position IDs of Y_{shad}. The shadow position IDs were shifted to obtain predictions for the current block, while the masking of main tokens is sub-diagonal and diagonal to prevent information leak due to Dream's shifted predictions.

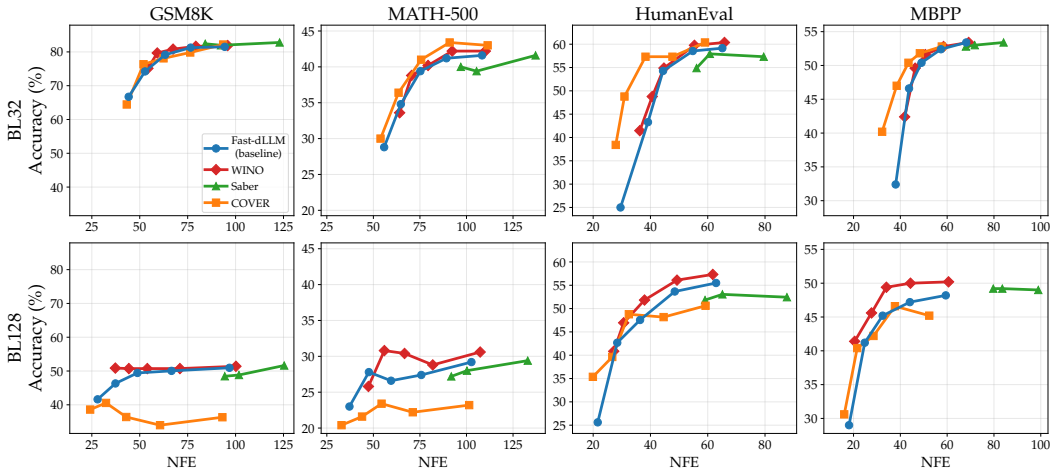
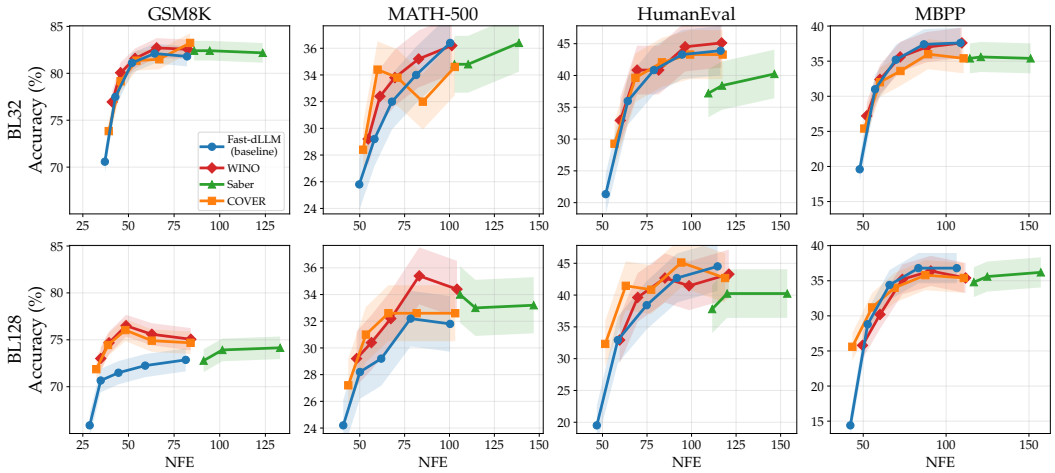
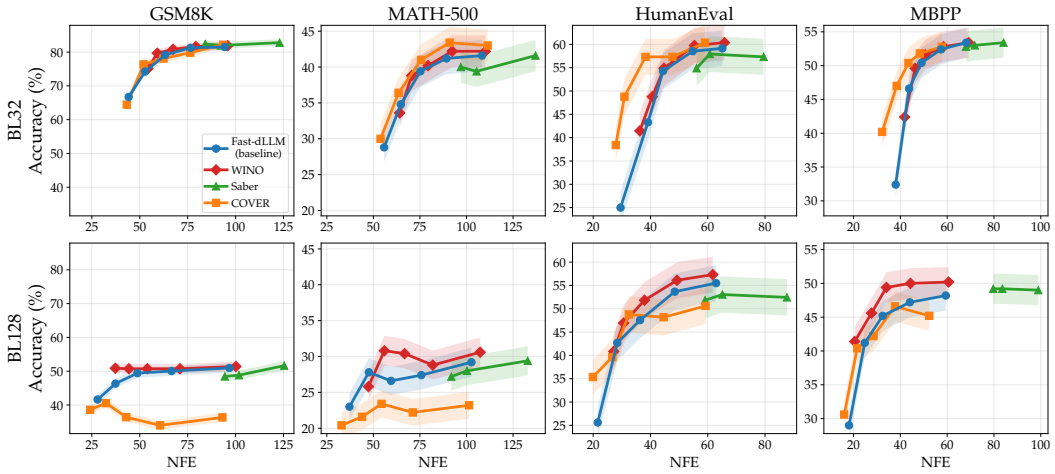


Figure 7: Figure 1 replicated for Dream-v0-Instruct-7B.

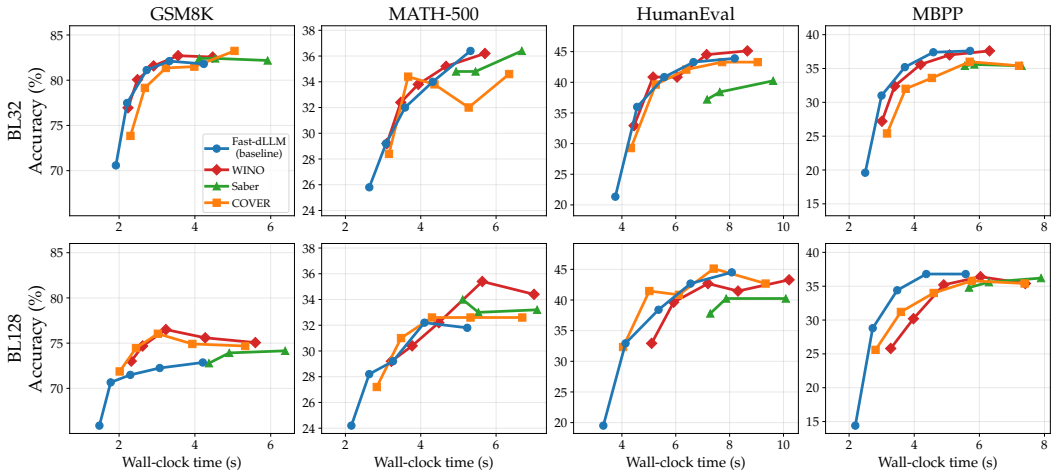


(a) LLaDA-8B-Instruct

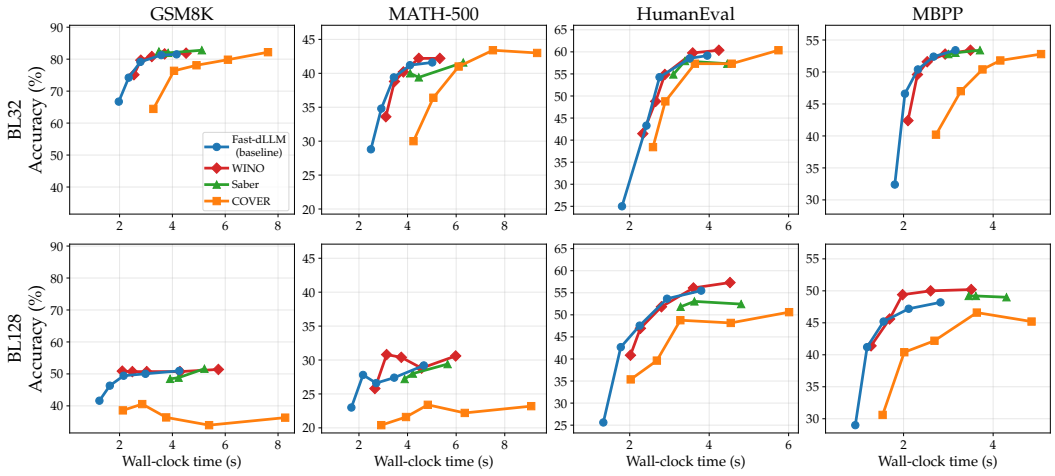


(b) Dream-v0-Instruct-7B

Figure 8: Figures 1 and 7 with shaded bands showing ± 1 binomial standard error of the test-set accuracy, $\sqrt{\hat{p}(1-\hat{p})/n}$, where $\hat{p}$ is the accuracy and n the number of test examples. At $BL = 32$, WINO's average improvements over Fast-dLLM's top performance ($\sim$0.4% on LLaDA and $\sim$0.5% on Dream) are small relative to these standard errors.



(a) LLaDA-8B-Instruct



(b) Dream-v0-Instruct-7B

Figure 9: Figures 1 and 7 replicated using latency (wall-clock time) as an efficiency measure instead of network function evaluations (NFEs). Once their wall-clock overhead is taken into account, the gains of remasking diminish further.

B Additional WINO experiments

B.1 Ablations and tuning

Design choices To ensure that our findings in Section 3.1 are robust across various design choices of WINO, we perform the following ablations and report results in Figure 10:

- **Consistency-based remasking criterion (*Equal*)** Recall that WINO selects positions for remasking via confidence thresholding based on shadow position confidences (Equation 4). To remove the effect of the exact choice of the remasking threshold λ_2, we instead consider here the consistency-based approach where the position is flagged for remasking if the max token under the shadow predictive distribution is not the same as the current token:

$$\mathcal{R}_t := \{k \in \mathcal{B} \setminus \mathcal{M}_t \mid \arg\max_{v \in \mathcal{V}} q_\theta^{k_s}(v \mid \tilde{\boldsymbol{x}}_t) \neq x_t^k\}\,.$$

- **Loop-guard (*MR1*)** To measure the impact of the exact loop-guard implementation, we consider here an alternative implementation where each position is allowed to be remasked at most once.
- **Shadow attention mask (*Diagonal*)** Instead of using a full attention mask for the shadow region as originally proposed in WINO (Figure 6a), we use a diagonal mask which prevents shadow positions from attending to each other.
- **Max-accept (*MA*)** The original WINO implementation imposes a cap on the number of positions that can be unmasked at each step.[6] Since this cap is neither described in the WINO paper nor part of the Fast-dLLM unmasking procedure [Wu et al., 2025], we disable it for the experiments in the main paper and re-enable it here to study its impact.

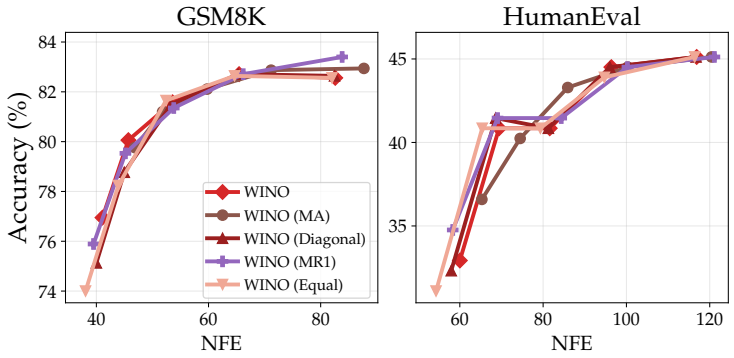


Figure 10: Ablations of four WINO design choices across GSM8K and HumanEval (LLaDA-8B-Instruct, $BL = 32$, $\tau = 0$). The performance across all variants is similar with no variant showing meaningful gains.

Remasking threshold Throughout the main paper, we use the WINO remasking threshold $\lambda_2 = 0.8$, one of the values suggested in the WINO paper [Hong et al., 2026]. To test whether our findings are specific to this choice, we additionally vary $\lambda_2 \in \{0.6, 0.7, 0.8, 0.9\}$ and report results in Figure 11. The Pareto frontiers are largely insensitive to λ_2, and WINO's improvement over Fast-dLLM's top performance at $BL = 32$ remains marginal for every threshold ($\sim$0.5% for $\lambda_2 \in \{0.6, 0.7, 0.8\}$ and $\sim$0.2% for $\lambda_2 = 0.9$, averaged over MATH-500 and HumanEval).

B.2 Shadow-token consistency

To further validate the shadow-token approximation (Section 3.2), we measure how often the shadow and oracle (leave-one-out) predictions agree on the top-1 token. As shown in Figure 12, the consistency rate is above 96% for all unmasking thresholds λ_1, and increases as fewer tokens are unmasked per step.

[6] `https://github.com/Feng-Hong/WINO-DLLM/blob/main/LLaDA/decoding.py`

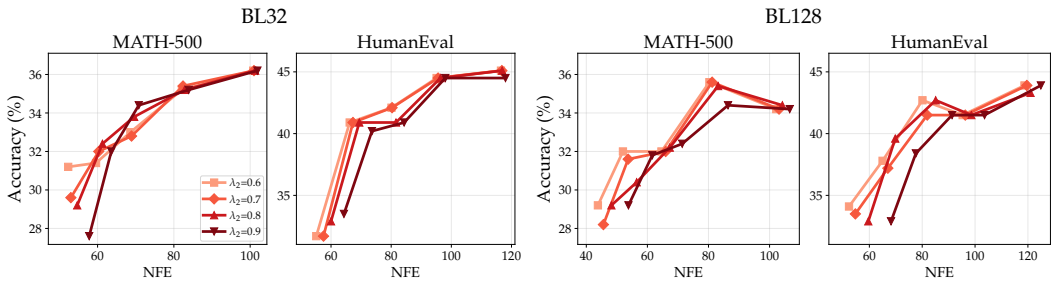


Figure 11: WINO Pareto frontiers for different remasking thresholds λ_2 (LLaDA-8B-Instruct, $\tau = 0$, $BL \in \{32, 128\}$), sweeping the unmasking threshold $\lambda_1 \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. Results are largely insensitive to the choice of λ_2.

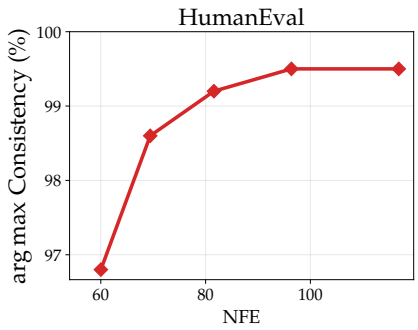


Figure 12: Consistency rate ($\{\arg\max_v q_\theta^{k_s}(v \mid \tilde{\boldsymbol{x}}_t) = \arg\max_v q_\theta^k(v \mid \boldsymbol{x}_{t,-k})\}$) of the highest-confidence tokens between the shadow and oracle predicted tokens *without* remasking ($\lambda_2 = 0$, LLaDA-8B-Instruct, HumanEval, $BL = 32$). Unmasking fewer tokens per step, i.e., increasing the threshold, further improves shadow predictions, potentially due to fewer approximations of KV matrices of other tokens at each step.

B.3 Flip-flop analysis

Expanded remasking To better understand the high flip-flop rates observed in Figure 3, we perform additional ablations that vary the set of positions remasked at each sampling step. Beyond the position k flagged by confidence thresholding (Equation 4), we additionally remask either its *spatial* or *temporal* neighbors. Formally, each flagged position $k \in \mathcal{R}_t$ is remasked together with its spatial or temporal neighborhood:

$$\mathcal{N}^S(k) := \{\, j \in \mathcal{B} \setminus \mathcal{M}_t \mid |j - k| \le S \,\},$$
$$\mathcal{N}^T(k) := \{\, j \in \mathcal{B} \setminus \mathcal{M}_t \mid 0 \le u_j - u_k \le T - 1 \,\},$$

where u_j is the step at which position j was last unmasked. Spatial neighbors are thus the S positions on either side of k, and temporal neighbors span the T steps from k's own unmasking onward ($T = 1$ the same step, $T = 2$ that step and the next). We denote the resulting variants WINO (S1), WINO (S2), WINO (T1), and WINO (T2), where the number gives the neighborhood size S or T. In addition, to ensure convergence, we switch to the aforementioned *MR1* loop-guard, in which each position can be flagged for remasking only once, but can still be remasked as a neighbor of other positions.

As shown in Figure 13 (left), neither expansion method improves over default WINO: both variants match or *degrade* accuracy across the NFE range, with spatial expansion degrading more as the neighborhood grows ($S = 2$ vs. $S = 1$). In addition, Figure 13 (right) shows that expanded remasking does not reduce the flip-flop rate, and often increases it. This argues against the cascading-dependency explanation: even when the surrounding context that would supposedly pin the prediction at x_t^k is partially removed, the dLLM still re-proposes the same token. The bottleneck therefore appears to lie in the model's predictive distribution at position k itself, rather than in the conditioning structure of the unmasked context.

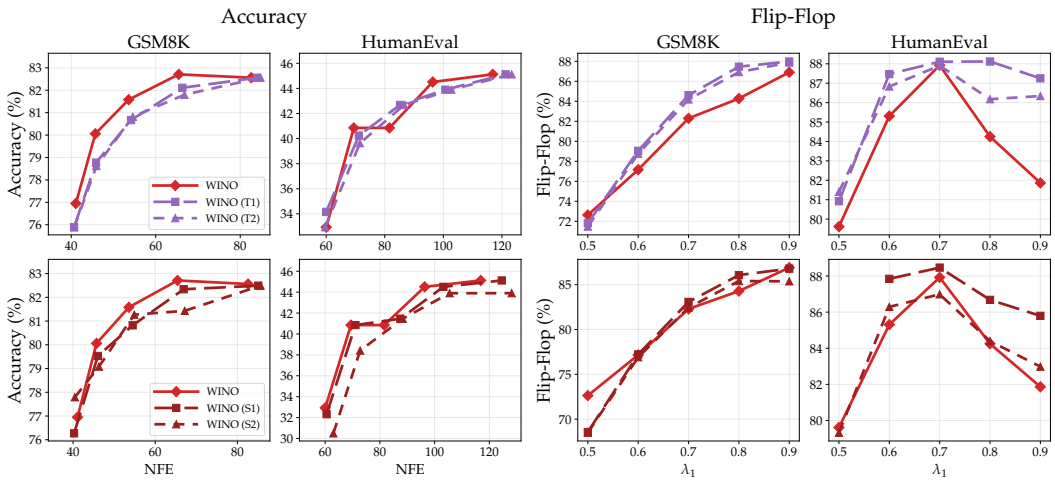


Figure 13: Accuracy (left, vs. NFE) and flip-flop rate (right, vs. λ_1) of expanded remasking ablations that additionally remask temporal (top) or spatial (bottom) neighbors (LLaDA-8B-Instruct, GSM8K and HumanEval, $BL = 32$).

Stochastic sampling at remasked positions Another possible explanation for the high flip-flop rates is greedy decoding ($\tau = 0$): when a remasked position is selected for unmasking again, the greedy sampler deterministically reproduces the model's most likely token, which is often the token that was just remasked. To test this, we sample remasked positions with $\tau = 1$ whenever they are selected for unmasking again, while all other positions are still decoded greedily, encouraging the model to explore alternative tokens rather than reproduce its earlier prediction. As shown in Figure 14, stochastic sampling at remasked positions has no significant effect on accuracy and only slightly lowers the flip-flop rates.

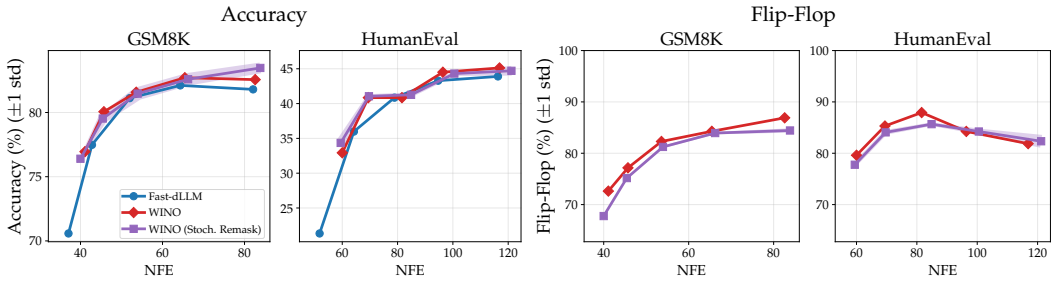


Figure 14: Accuracy (left) and flip-flop rate (right) of WINO with greedy ($\tau = 0$) vs. stochastic ($\tau = 1$) sampling at remasked positions (LLaDA-8B-Instruct, GSM8K and HumanEval, $\lambda_2 = 0.8$, $BL = 32$), sweeping the unmasking threshold $\lambda_1 \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. Fast-dLLM is shown for reference. Stochastic remasking results are averaged over 3 seeds (± 1 std).

Correct vs. incorrect generations To test whether flip-flops concentrate on generations where the model makes mistakes, we split the flip-flop rate by whether the final answer is correct (LLaDA-8B-Instruct, WINO with $\lambda_1 = 0.6$, $\lambda_2 = 0.8$, $BL = 32$). As shown in Table 2, the rates are nearly identical for correct and incorrect generations across all four datasets, suggesting that flip-flops cannot reliably indicate when the model is likely to produce an incorrect answer.

Table 2: Average flip-flop rate (%) for correct and incorrect generations (LLaDA-8B-Instruct, WINO, $\lambda_1 = 0.6$, $\lambda_2 = 0.8$, $BL = 32$). Number of generations in parentheses.

	GSM8K	MATH-500	HumanEval	MBPP
Correct	74.6 (1056)	76.3 (162)	85.6 (67)	81.9 (162)
Incorrect	77.9 (263)	77.6 (338)	82.4 (97)	81.9 (338)

B.4 Token entropy under stochastic decoding

To better understand the loss in diversity of WINO under stochastic decoding (Figure 4), we measure the entropy of the model's predictive distribution at each position when it is unmasked (LLaDA-8B-Instruct, $BL = 32$, $\tau = 0.8$, $\lambda_1 = 0.6$, $\lambda_2 = 0.8$, 64 samples per problem). Since WINO can unmask a position more than once, we report the entropy at both its first and final unmasking. As shown in Table 3, WINO remasks only $\sim$4% of positions, but these are highly uncertain when first unmasked. After remasking, they are re-committed with much lower entropy, below that of the average position, which lowers WINO's overall token entropy compared to Fast-dLLM.

Table 3: Mean token entropy at the first and final unmasking of each position (LLaDA-8B-Instruct, $BL = 32$, $\tau = 0.8$, $\lambda_1 = 0.6$, $\lambda_2 = 0.8$, 64 samples per problem). Since Fast-dLLM unmasks each position once, its final entropy equals its first (=). WINO remasks 3.6% (MATH-500) and 4.4% (HumanEval) of positions.

		MATH-500		HumanEval	
Method	Positions	First	Final	First	Final
Fast-dLLM	All	0.64	=	0.79	=
WINO	All	0.63	0.61	0.77	0.74
WINO	Remasked	1.15	0.43	1.38	0.55

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: For all the main claims in the abstract/introduction, we conduct extensive experiments in Section 3.

   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: We have a paragraph on limitations in Section 5, where we note that our failure-mode analyses and the stochastic decoding experiments focus on WINO.

   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: [N/A]

Justification: Our paper does not include theoretical results - instead we aim to provide empirical insights and analysis into limits and promises of confidence-based remasking in masked dLLMs.

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: We aimed to provide exact settings of our experiments (including hyperparameters) throughout the paper. We also release our code at `https://github.com/stipefrkovic/remasking-dllm`.

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: [Yes]

   Justification: We release our code at `https://github.com/stipefrkovic/remasking-dllm`. It builds on publicly accessible repositories, e.g., `https://github.com/Feng-Hong/WINO-DLLM`.

   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: We aimed to provide exact settings of our experiments (including hyperparameters) throughout the paper. We also follow the standard evaluation setting in the ongoing masked dLLM literature (e.g., by using the 4 datasets: GSM8K, MATH-500, HumanEval, MBPP).

   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: [Yes]

   Justification: Whenever applicable, we report error bars over multiple seeds (e.g., Figures 5 and 14). For `pass@`k (Figure 4), performance is estimated analytically. For greedy decoding (Figure 1), which is deterministic, we report binomial standard errors of the test-set accuracies (Figure 8).

   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: Given that we evaluate training-free remasking approaches, our experiments require standard compute resources only (i.e., all experiments were conducted on 4xA100 GPUs).

   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: We adhere to the NeurIPS Code of Ethics strictly.

   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: We include a paragraph on broader impacts in Section 5.

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: Our work introduces no potentially high-risk artefacts (i.e., no pretrained model, no new dataset).

    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 make sure to cite and reference all prior work (including code-bases) that helped us complete our project.

    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: [Yes]

    Justification: Our only new asset is our code, released and documented at `https://github.com/stipefrkovic/remasking-dllm`.

    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: Our work 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: Our work 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: In the preparation of our work, we used LLMs in a "standard way", i.e. to help with editing/polish of the writing and to help with code implementation. The use of LLMs did not impact the core methodology.

    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.