SimSD: Simple Speculative Decoding in Diffusion Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Diffusion large language models (dLLMs) have recently emerged as a promising alternative to autoregressive (AR) LLMs, offering faster inference through parallel or blockwise decoding. However, their masked language modeling formulation remains incompatible with standard token-level speculative decoding, one of the most effective acceleration techniques for AR models. In AR decoding, the causal mask preserves temporally valid token-level contexts, enabling a target model to verify multiple drafted tokens in a single forward pass. In contrast, dLLMs rely on mask tokens and bidirectional attention, causing the effective context to change across denoising steps and preventing direct token-level speculative verification. To bridge this gap, we propose a simple but effective speculative decoding algorithm for diffusion language models, named SimSD, which mainly adopts a plug-and-play masking strategy that equips dLLMs with temporally valid token-level contexts for speculative decoding. Our method explicitly introduces reference tokens from draft-model predictions and designs an attention mask that regulates their interaction with current-step tokens, allowing dLLMs to compute valid logits for drafted tokens in a single forward pass. This restores the key verification ability provided by causal masking in AR models while preserving the parallel decoding advantages of dLLMs. The proposed method is training-free and can be flexibly integrated with other acceleration techniques such as KV cache and blockwise decoding. Experiments on SDAR-family dLLMs across four benchmarks show that our method achieves up to $7.46\times$ higher decoding throughput while maintaining and even improving average generation quality. Our code will be publicly released.

1 Introduction

Large language models (LLMs) [1, 2, 3] have achieved remarkable success across a wide range of natural language processing tasks [4, 5]. These advances are largely driven by autoregressive (AR) Transformer architectures [6]. Despite the performance, AR decoding is inherently sequential, which requires strictly performing token-by-token generation [7, 8]. As an alternative, diffusion large language models (dLLMs) [9, 10] generate text by iteratively denoising an incomplete sentence with bidirectional attention. With a properly designed denoising schedule, dLLMs enable predicting multiple tokens at one decoding step, leading to promising parallelism and lower inference latency.

Recent commercial dLLMs [11, 12, 13] have substantially narrowed the performance gap with state-of-the-art AR models, while demonstrating much faster inference speed with parallel or blockwise decoding [14, 15, 16, 17]. However, further improving dLLM inference speed remains challenging because its masked language modeling design [18, 19] is not naturally compatible with many acceleration techniques developed for AR models, especially for the widely-used speculative decoding method [8, 20, 21]. In AR models, the token-level causal mask ensures that every generated token

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

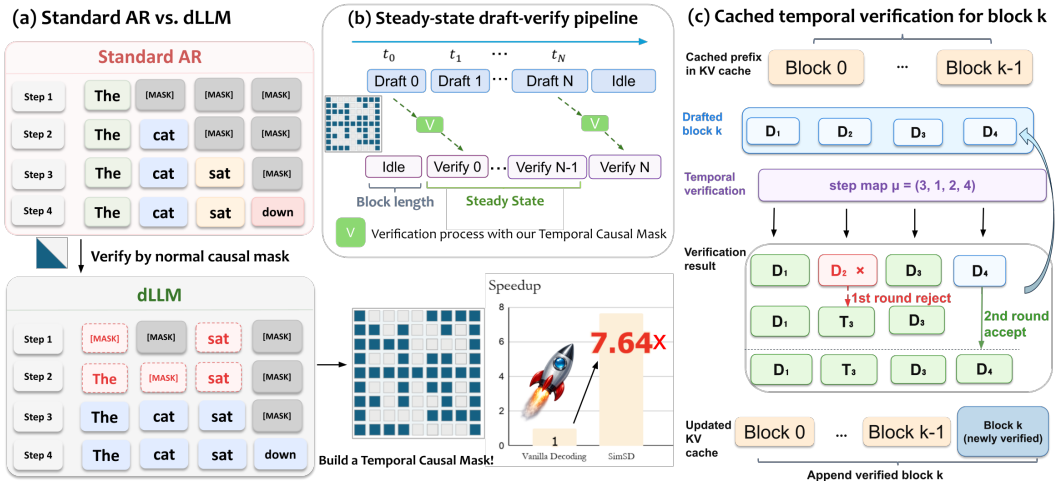


Figure 1: SimSD restores token-level speculative decoding for diffusion language models. Vanilla dLLMs cannot directly support speculative decoding because bidirectional attention breaks temporal token-level contexts. SimSD adopts a temporal causal mask for supporting standard speculative verification in a single forward pass, leading to faster decoding without hurting performance.

remains in the context for subsequent positions. Thus, for speculative decoding in AR models, all the drafted tokens from a smaller model only require a single forward pass over their concatenated sequence to produce valid logits for all drafted tokens under their correct prefixes. This property is central to speculative decoding, where multiple possible tokens can be verified in parallel. By contrast, dLLMs rely on mask tokens to indicate unresolved positions and use bidirectional attention to condition on the visible context. Consequently, the effective input context changes across denoising or generation steps, and dLLMs cannot maintain temporally valid token-level contexts. Therefore, standard token-level speculative verification cannot be applied directly to dLLMs.

To address this limitation, we propose a simple but effective method namely **SimSD** that adopts a plug-and-play masking strategy that equips dLLMs with temporally valid token-level contexts for speculative decoding. The key idea is to construct a token-level causal structure in which each token can directly attend to other information from the additional reference context, *i.e.,*inference results from a smaller model, analogous to how AR models read in-context tokens through the causal mask. To this end, we explicitly add several tokens as the reference context and design an attention mask that regulates interactions between these reference context and the tokens predicted at the current step. In this way, the visible context is no longer determined only by the current masked input, but built to include both current-step tokens and reference context. This restored temporal context allows a dLLM to compute logits for drafted tokens and thereby supports standard speculative verification in a single forward pass. It matches the verification mechanism used in AR speculative decoding, and gives dLLMs the key ability while retaining the parallel decoding benefits.

Importantly, the proposed mask requires no additional training and can be directly applied to existing dLLM decoding pipelines. Since we only modify the design of attention mask and copy few hidden states, our method can be flexibly integrated with other acceleration techniques, such as KVCache and blockwise decoding. To verify it, we conduct extensive experiments on SDAR [15] family dLLMs and four benchmarks, GSM8K [22], MBPP [23], TriviaQA [24], and MMLU [25]. Experimental results have demonstrated that our method can achieve up to $7.46\times$ higher decode throughput, and the generation quality does not degrade and even slightly boosts (+1.7%), compared with the Vanilla SDAR baseline with tensor parallelism.

2 Related work

Diffusion Language Models. Masked language modeling was first established as a pretraining objective by BERT style bidirectional encoders[18], which recover a small fraction of held out tokens from their bidirectional context. Subsequent work recast this objective as a continuous time generative

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

process, with DiffusionBERT [26] providing an explicit bridge between BERT style masked prediction and absorbing state diffusion. Building on this masked language modeling tradition, discrete diffusion language models [19, 27] corrupt a clean sample by replacing a randomly chosen subset of its tokens with the special symbol `[MASK]`, producing a noised sequence. The forward corruption factorizes across all response token positions: each position is independently replaced by `[MASK]` with a certain probability. The reverse process supports an accelerated parameterization in which several masked positions are recovered jointly within a single transition from high noise level to a lower level, rather than one position at a time.

Blockwise Decoding for dLLMs. A complementary line of work adapts pretrained autoregressive checkpoints into diffusion language models, preserving the AR backbone's linguistic competence while enabling parallel decoding [28]. A common strategy is to introduce block causal attention, which imposes causal dependencies across coarse-grained blocks while allowing bidirectional denoising within each block. This design has been adopted by recent dLLMs [15, 28, 29, 30], suggesting that AR pretraining can transfer effectively to masked diffusion objectives when the attention pattern is properly constrained. Block diffusion models [15, 31] formalize this idea by partitioning the sequence into contiguous fixed-size blocks and replacing full bidirectional attention with block causal attention. During inference, all positions within a block can be denoised jointly, preserving the parallel sampling advantage of diffusion, while dependencies across blocks remain causal. This structure also enables standard KV caching across blocks, avoiding the cache invalidation issues caused by combining bidirectional attention with KV reuse in full attention dLLMs [31, 32, 33, 34, 35].

Speculative Decoding. Speculative decoding (SD) accelerates LLM inference by using a small draft model to propose multiple candidate tokens and a larger target model to verify them in parallel [20, 21]. This paradigm is particularly effective at low batch sizes on GPUs, where decoding is often memory bandwidth bound rather than compute bound. Subsequent work improves this pipeline through better proposal models, tree-based verification, multi-token prediction heads, feature level drafting, and drafter-free parallel decoding [7, 36, 37, 38]. Existing work on diffusion-based speculative decoding falls into two main directions. The first uses diffusion models as drafters for autoregressive verifiers. SpecDiff and SpecDiff-2 train or align diffusion drafters to improve proposal quality for AR targets [14, 39]; DiffuSpec instead reuses a pretrained diffusion language model as a training-free drafter while retaining an AR verifier [40], and DFlash explores block diffusion drafting under the same AR target setting [17]. The second direction accelerates a single dLLM through self-speculation or trajectory-level verification, where the same diffusion model or block diffusion process proposes and checks candidate states [16, 41, 42, 43]. For example, S2D2 is described as a training-free self-speculative decoding framework for block-diffusion language models, where the same pretrained model serves as both drafter and verifier under different modes; BlockSpec similarly targets blockwise speculative decoding for dLLMs by exploring future decoding trajectories.

3 Methodology

As dLLMs can not maintain temporally valid token-level contexts as AR models to support multiple possible tokens verification in a single forward pass, we aim to devise a new attention masking strategy to solve it in dLLMs. Based on it, we propose a simple but effective speculative decoding method for accelerating dLLMs during inference, which can significant reduce the inference latency and support the integration with other techniques such as blockwise decoding and KV cache.

3.1 Token-level Temporal Causal Attention

Original dLLMs use bidirectional attention to control the visible context. Here, we replace this design with a token-level temporal causal attention strategy, which ensures that each token can attend to its current step context as well as additional reference information from earlier temporal steps. Concretely, we first construct an input layout that keeps copies of related tokens as reference context, then build an attention mask according to token-level temporal order, and finally align position encodings before predicting the masked tokens.

Input Layout Prepare. Given a prompt c, we construct the attention context by appending two response sections of length B: a data section $D = (D_1, \ldots, D_B)$ and a mask section

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

$\widetilde{D} = (\widetilde{D}_1, \ldots, \widetilde{D}_B)$. The data section stores reference tokens, while the mask section contains the prediction slots, where $\widetilde{D}_k = \texttt{[MASK]}$. Each pair $(D_k, \widetilde{D}_k)$ corresponds to the same response position but plays different roles: D_k provides contextual information from a reference or previously decoded trajectory, whereas $\widetilde{D}_k$ is the position where logits are computed for token prediction. This layout allows the model to explicitly keep reference tokens in the context while still performing masked token prediction.

Temporal Attention Mask. Each response position is assigned a temporal step label $\tau \in \{1, \ldots, B\}$, indicating the inference step associated with that token. We represent this assignment using a step map $\mu = (\mu_1, \ldots, \mu_B)$, a permutation of $\{1, \ldots, B\}$, where both D_k and $\widetilde{D}_k$ share the same temporal label $\tau = \mu_k$. Prompt tokens are assigned $\tau = 0$. For example, the left-to-right order corresponds to $\mu = (1, 2, \ldots, B)$, while other permutations allow different decoding orders. This step map extends block-level causal attention [31] to a finer token-level temporal structure within each block. We define the attention mask A by $A_{ij} = 1$ if position i is allowed to attend to position j, and $A_{ij} = 0$ otherwise. Prompt tokens are visible to all response positions. For response positions, the mask is defined as

$$A_{ij} = \begin{cases} 1 & \text{if } i, j \in D \text{ and } \tau(j) \le \tau(i), \\ 1 & \text{if } i \in D,\ j \in \widetilde{D} \text{ and } \tau(j) > \tau(i), \\ 1 & \text{if } i \in \widetilde{D},\ j \in D \text{ and } \tau(j) < \tau(i), \\ 1 & \text{if } i, j \in \widetilde{D} \text{ and } \tau(j) \ge \tau(i), \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

This mask enforces a temporally valid visible context. A masked prediction slot $\widetilde{D}_i$ can attend to decoded or reference data tokens from earlier steps, D_j with $\tau(j) < \tau(i)$, while future or current unresolved positions remain represented by mask placeholders. Conversely, a data token D_i is computed from the information that would have been available when it became known: earlier data tokens and later mask placeholders. Padding positions are masked out by zeroing their corresponding rows and columns.

Position Encoding Alignment. Because the data and mask sections contain paired positions, their positional encodings must be aligned. Under the above layout, D_k and $\widetilde{D}_k$ refer to the same response position but appear at different absolute sequence indices. With standard rotary position embedding (RoPE) [44], this would rotate their query and key vectors using different position indices, even though they correspond to the same token position. To avoid this mismatch, we apply a RoPE copy strategy: each mask position $\widetilde{D}_k$ reuses the RoPE index of its paired data position D_k, while prompt and data tokens keep their original absolute indices. This alignment ensures that logits computed at masked positions are positionally consistent with the corresponding response tokens.

Masked Token Prediction. After applying the temporal attention mask and position encoding alignment, the model predicts tokens only at the mask section. For a prediction slot $\widetilde{D}_i$, the visible context consists of the prompt, earlier data tokens D_j with $\tau(j) < \tau(i)$, and mask placeholders $\widetilde{D}_j$ with $\tau(j) \ge \tau(i)$. Therefore, the logits at $\widetilde{D}_i$ approximate the token distribution under a temporally consistent context, analogous to the prefix-conditioned distribution used in autoregressive speculative verification. This enables dLLMs to verify drafted tokens at the token level while preserving their parallel decoding structure.

3.2 Application for Speculative Decoding

We next describe how token-level temporal causal attention enables standard speculative decoding for dLLMs. Speculative decoding typically uses a small draft model to propose candidate tokens and a larger target model to verify them in parallel with rejection sampling [20, 21]. In our setting, both the drafter and verifier are dLLMs: the smaller dLLM first proposes candidate tokens together with their temporal order, and the target dLLM then scores all candidates in a single forward pass using the proposed token-level temporal causal mask.

Draft Model for Candidates Collection. Consider a dLLM decoding sequence of length B, and let R denote the number of remaining undecoded positions. When $R \ge \gamma$, a smaller draft dLLM

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

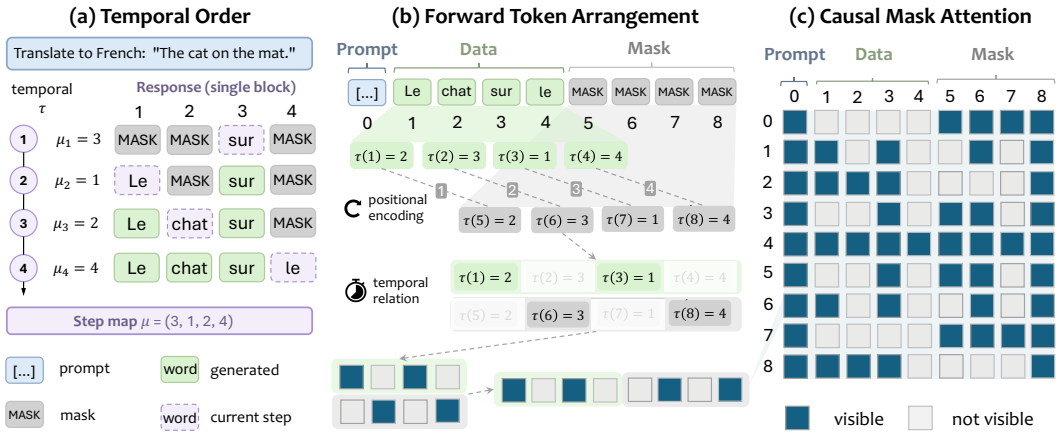


Figure 2: Token-level temporal causal attention. (a) A step map assigns each token a temporal decoding order. (b) The input is arranged as prompt, data, and mask tokens, with shared temporal labels and aligned positional encoding. (c) The attention mask enforces token-level temporal causality.

with parameters ϕ proposes γ candidate tokens $\widehat{D} = (\widehat{D}_{o_1}, \ldots, \widehat{D}_{o_\gamma})$, where $o_1, \ldots, o_\gamma$ denotes the draft-induced temporal order. Specifically, position o_r is generated at draft step r. The drafter therefore determines both the candidate tokens and the order in which they should be verified. The target verifier reuses this order by assigning temporal label $\tau = r$ to the slot corresponding to $\widehat{D}_{o_r}$.

Target Model Verification. The verification input follows the paired data-mask layout introduced in Sec. 3.1, but only for the γ drafted positions in the current speculative step. The r-th data slot stores the drafted token $\widehat{D}_{o_r}$, and the paired mask slot serves as its verification position. We then construct the temporal attention mask using $\tau = r$ for each drafted token and apply the same RoPE-copy alignment between each drafted token and its paired verification slot. Under this mask, the verification slot for $\widehat{D}_{o_r}$ can attend to the prompt and to earlier drafted tokens $\widehat{D}_{o_1}, \ldots, \widehat{D}_{o_{r-1}}$, but not to the current token or later drafted tokens. Thus, the target distribution at this slot, denoted by $p_\theta^{(r)}$, scores $\widehat{D}_{o_r}$ under the valid draft-induced causal context. Importantly, all target distributions $\{p_\theta^{(r)}\}_{r=1}^{\gamma}$ are obtained from one target forward pass, allowing the target dLLM to verify multiple drafted tokens in parallel while preserving token-level causal validity. We apply the standard speculative decoding acceptance rule along the draft-induced temporal order. Let $p_\phi^{(r)}$ be the draft distribution used to sample $\widehat{D}_{o_r}$. For $r = 1, \ldots, \gamma$, the candidate token is accepted with probability

$$\alpha_r = \min\left(1, \frac{p_\theta^{(r)}(\widehat{D}_{o_r})}{p_\phi^{(r)}(\widehat{D}_{o_r})}\right). \tag{2}$$

The algorithm accepts the longest prefix before the first rejection. If $\widehat{D}_{o_r}$ is rejected, the remaining drafted suffix is discarded, and the target dLLM samples the token at position o_r. The next speculative step then continues from the updated block state. If all γ candidates are accepted, they are committed to the block, and speculative decoding proceeds while at least γ undecoded positions remain. When fewer than γ positions remain, drafting stops and the target dLLM completes the block directly.

Training-free Deployment. The proposed procedure is training-free. It only modifies the verification-time input layout, attention mask, and RoPE indexing, and can therefore be applied to pretrained draft and target dLLMs without parameter updates. The rejection-sampling correction preserves the target distribution induced by the token-level temporal causal mask, while reducing the number of expensive target forward passes. This makes cross model dLLM-to-dLLM speculative decoding practical without jointly designing or retraining the drafter and verifier.

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

3.3 Efficiency Discussion

Consider one speculative verification step with draft length γ. Let T_θ be the walltime of one target dLLM verification pass, and let T_ϕ be the average walltime of one draft step. One speculative verification step with draft length γ costs, where ρ define the draft-to-target cost ratio:

$$T_{\mathrm{SD}}(\gamma) = \gamma T_\phi + T_\theta = (1 + \gamma\rho)T_\theta;\ \rho = \frac{T_\phi}{T_\theta}. \tag{3}$$

Following the analysis of Leviathan et al. [20], we aim to quantify the benefit of verifying multiple draft tokens with one target forward pass. Our verification layout introduces an additional overhead because it duplicates the drafted sequence to form the data-mask paired input. Let ℓ denote the draft sequence length and let n be the original verification sequence length before duplication. The paired layout increases the effective sequence length from n to $n + \ell$. Therefore, the attention cost increases from $O(n^2)$ to $O((n + \ell)^2)$, with an extra cost.

$$O((n + \ell)^2 - n^2) = O(2n\ell + \ell^2). \tag{4}$$

Adaptation under Blockwise Decoding and KV Cache. The duplicated verification layout has a bounded overhead under blockwise decoding with KV cache. Let n be the prefix length and ℓ be the duplicated draft length. With KV cache, the prefix keys and values are reused, so the verifier only computes the newly introduced duplicated positions. Thus the additional computation is bounded by

$$O((n + \ell)^2 - n^2) = O(n\ell + \ell^2) \le O(nB + B^2). \tag{5}$$

where the inequality follows from $\ell \le B$ under blockwise decoding. Since B is a small constant in our experiments, e.g., $B = 4$ or $B = 8$, the overhead is effectively linear in the cached prefix length and is much smaller than duplicating and recomputing the full response sequence. In practical settings, this overhead is acceptable.

4 Experiments

4.1 Experimental setup

Models. Following previous studies[20][21], we use small scale draft model and large scale target model from the same model series for the speculative decoding experiments. We use dLLMs trained with different block lengths from the widely used SDAR[15] series.

Datasets & metrics. To assess the generation efficiency and effectiveness of our method, we evaluate it on four standard benchmarks covering different task categories: **GSM8K** [22] for mathematical reasoning, **MBPP** [23] for code generation, and **TriviaQA** [24] and **MMLU** [25] for general NLP and knowledge-intensive evaluation. For efficiency, we report **TPS** and the **speedup ratio** against vanilla decoding. For effectiveness, we report accuracy for GSM8K, TriviaQA, and MMLU, and pass@1 for MBPP.

Baselines. (i) **Vanilla** follows the standard dLLM decoding procedure with block causal attention and KV cache. Since speculative decoding uses two GPUs, we run Vanilla with tensor parallelism = 2 for a fair comparison. (ii) **Vanilla with CUDA Graph** augments Vanilla decoding with CUDA Graphs to reduce kernel launch overhead. (iii) **S2D2** [42] performs training free self speculation, where the dLLM first drafts tokens and then verifies them from left to right using the same model with block size set to 1. SSD [41] is designed for fully bidirectional dLLMs such as LLaDA and Dream, while our experiments use SDAR [15], a block causal dLLM models series with KV cache. BlockSpec [16] is orthogonal to our verifier-side contribution, as it focuses on blockwise trajectory speculation rather than token-level target verification. Therefore, these methods are not direct baselines under our SDAR-based setting, and we do not include them in the main comparison.

Configurations. We follow the default prompts and chat templates from the OpenCompass [45] evaluation framework. Unless otherwise specified, we fix the maximum generation length to $L = 512$ and set the number of denoising steps to the response sequence length N. For block length $B = 4$, we use SDAR-1.7B-Chat as the draft model and SDAR-8B-Chat as the target model; for block length $B = 8$, we use SDAR-1.7B-Chat-b8 and SDAR-8B-Chat-b8. We set the speculative draft length to

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

$\gamma = B$, and fall back to vanilla target decoding when the number of remaining positions in the current block satisfies $R < \gamma$. All generation-quality evaluations are conducted in the zero-shot setting with temperature $= 1.0$ and top_$p = 1.0$. All experiments use batch size 1, FP32 precision, and KV cache. We use two NVIDIA RTX PRO 6000 Blackwell GPUs, with the draft and target models deployed on separate GPUs to avoid resource contention; for the Vanilla baseline, we use tensor parallelism of size 2. CUDA Graph is enabled for all methods except the basic Vanilla baseline. Throughput is measured as decoded tokens per second, excluding prompt prefilling, with 3 warmup runs. For speed evaluation, we use 200 examples from each dataset, while generation quality is evaluated on the full benchmark. Speedup is computed relative to Vanilla TP-2 under the same block length.

4.2 Main results

Table 1: Efficiency Performance. We report decode tokens per second. S2D2 denotes self-speculative decoding for diffusion LLMs, and SD denotes our speculative decoding method. Speedup is measured relative to Vanilla with tensor parallelism of size 2 under the same block size. BL denotes the block length. CG denotes CUDA Graph. Van. denotes vanilla decoding

BL	Methods	Decoded Tokens per Second					Speedup
		GSM8K	TriviaQA	MBPP	MMLU	Mean	
4	Vanilla	10.0	11.4	11.4	5.59	9.6	1.00×
	Vanilla + CG	50.1	50.5	50.7	56.04	51.8	5.40×
	S2D2	54.8	28.3	47.7	32.3	40.8	4.25×
	SimSD	**72.6**	**73.3**	**72.2**	**68.4**	**71.6**	**7.46×**
8	Vanilla	15.3	20.4	18.7	6.21	15.2	1.00×
	Vanilla + CG	30.9	31.9	30.4	36.8	32.5	2.14×
	S2D2	57.6	30.0	50.2	28.0	41.5	2.74×
	SimSD	**74.1**	**88.3**	**80.4**	**84.4**	**81.8**	**5.40×**

Efficiency Performance. Table 1 demonstrates the main efficiency comparison between our method and various baselines. Overall, our method consistently outperforms all the other baselines on decoding throughput across all benchmarks and block sizes. Our method reaches 71.6 and 81.8 tokens/s on average for $B = 4$ and $B = 8$, corresponding to $7.46\times$ and $5.40\times$ speedups over Vanilla decoding. Compared with S2D2, our method also provides substantially higher throughput, showing that our construction of temporal causal mask provides more precise context information.

Table 2: Accuracy performance. We report accuracy for GSM8K, TriviaQA, and MMLU, and pass@1 for MBPP. Van. denotes vanilla decoding, SD denotes our method SimSD, BL denotes the block length and α denotes the token-level acceptance rate.

BL	GSM8K			MBPP			TriviaQA			MMLU			Avg.	
	Van.	SD	α	Van.	SD	α	Van.	SD	α	Van.	SD	α	Van.	SD
4	**0.922**	0.896	0.915	**0.669**	0.662	0.806	0.486	**0.537**	0.576	0.706	**0.758**	0.973	0.696	**0.713**
8	**0.906**	0.894	0.867	**0.642**	0.634	0.731	0.471	**0.503**	0.519	0.704	**0.760**	0.949	0.681	**0.698**

Accuracy performance. As the results shown in table 2, it is convinced that our speculative decoding method preserves generation quality comparable to, and in some cases slightly better than, vanilla decoding. Across block lengths 4 and 8, the average performance remains on par with vanilla decoding and even shows small improvements. For block length 4, the average score increases from 0.696 under vanilla decoding to 0.713 with speculative decoding. Similarly, for block length 8, the average score improves from 0.681 to 0.698. Although our attention mask construction introduces a certain approximation to the exact step by step denoising process, these results indicate that the approximation does not noticeably harm task accuracy in the speculative decoding setting. Instead, the proposed verification mechanism improves decoding efficiency while maintaining, and sometimes slightly improving, generation quality.

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

4.3 Additional Results

Precision Analysis of SimSD. Since all positions are evaluated in one forward pass, this construction is not an exact lossless replay of sequential denoising. Instead, it provides an approximation of the same temporal conditioning structure. The approximation arises only from higher order interactions among hidden states that are computed simultaneously, while the direct attention visibility of each position matches the intended denoising step.

We use two experiments to quantify the practical effect of this approximation. (i) **Self draft distribution analysis.** In this setting, we compare the output distributions induced by conventional generation and by SD generation conditioned on the correct tokens, which directly measures the distributional discrepancy introduced by the parallel construction. (ii) **Cross model distribution analysis.** In the second experiment, we evaluate whether this approximation affects the normal speculative decoding pipeline, where draft tokens are proposed by a smaller model and verified by a larger target model.

We further analyze the distributional approximation error introduced by the causal mask construction. For both figures, D and V in $\mathrm{KL}(D\|V)$ denote the draft and verification distributions, respectively, and Loss denotes the CE loss. As shown in Figure 4 and Figure 3, the discrepancy between sequential denoising and parallel verification gradually increases from step 0 to step 3. In the self draft setting, $\mathrm{KL}(D\|V)$ increases from 0.0889 at step 0 to 0.2537 at step 3, while the loss increases from 0.0141 to 0.2567. A similar trend appears in the cross model setting, where $\mathrm{KL}(D\|V)$ increases from 0.1528 at step 0 to 0.5010 at step 3, and the loss increases from 0.0605 to 0.4389.

The self draft setting consistently exhibits smaller approximation error than the cross model setting. For example, at step 3, $\mathrm{KL}(D\|V)$ is 0.2537 for self draft but 0.5010 for cross model, and the corresponding loss is 0.2567 versus 0.4389. This is expected because self draft uses the same model on both sides, whereas cross model verification additionally includes the distributional mismatch between the draft and target models. Nevertheless, the overall approximation error remains small in both settings, as reflected by the high top 1 agreement. The self draft setting maintains top 1 agreement of at least 0.9750, and the cross model setting remains at least 0.9400 even at the last step. These results indicate that the causal mask provides highly accurate reference context for each verification position, quantitatively explaining why the main accuracy results do not show degradation under our speculative decoding method.

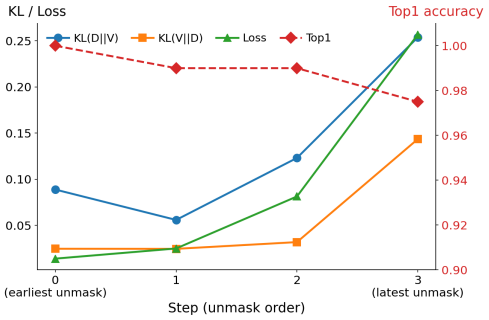


Figure 3: Self draft distribution analysis. We use SDAR-8B-Chat as both draft and target with $B = 4$ and temperature 0.

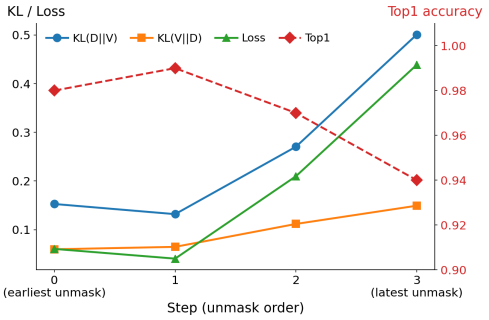


Figure 4: Cross model distribution analysis. We use SDAR-1.7B-Chat as draft and SDAR-8B-Chat as target with $B = 4$ and temperature 0.

Effect of γ. We analyze how the choice of γ affects the quality and efficiency of SimSD. As shown in Table 3, both the realized acceptance rate and final accuracy generally increase as γ becomes larger. This trend suggests that a longer draft scope provides richer target side verification context, allowing SimSD to accept higher quality draft tokens and produce more reliable generations. In particular, increasing γ from 3 to 8 improves real α from 0.834 to 0.964 and accuracy from 50.0% to 82.0%, showing that larger γ values can substantially improve the quality of the draft and verify process.

In contrast, TPS does not exhibit a simple monotonic pattern with respect to γ. This is expected because changing γ affects more than the verification computation itself. A larger γ changes the number of tokens verified per step, the interaction between γ and the block length, and the overhead

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

of KV cache management during multi block verification. These factors jointly determine the end to end runtime, so the throughput cannot be explained solely by the acceptance rate or the verification scope. For example, $\gamma = 4$ achieves the highest TPS in our experiment, while $\gamma = 8$ achieves the best accuracy and acceptance rate. This indicates a practical tradeoff: larger γ improves generation quality and acceptance behavior, whereas intermediate γ values may provide a better speed quality balance. Therefore, we choose γ equal to the block length for main experiment, which provides the best overall balance between generation quality and TPS in our setting.

Ablation Study on RoPE Alignment. To examine the impact of RoPE alignment, we fix the attention structure and vary only the position ids assigned to tokens in the mask block. In the default setting, referred to as Aligned RoPE, each mask token copies the position id of its corresponding token in the reference context, so the two tokens share the same rotary position embedding. In the Without Aligned RoPE setting, mask tokens are instead assigned position ids according to their physical locations in the concatenated sequence, continuing after the reference context rather than reusing the data-token positions.

Table 3: Effect of γ on GSM8K. We evaluate different values of γ using SDAR-1.7B-b8 as draft model and SDAR-8B-b8 as target model with block length $B = 8$ on GSM8K. TPS denotes generated tokens per second, real-α denotes the realized token-level acceptance rate, and avg accept denotes the average number of accepted tokens per verification step.

γ	Accuracy	TPS	real-α	Avg. accept	ms/block
3	50.0%	55.14	0.834	4.54/8	90
4	54.0%	**74.40**	0.887	4.93/8	**71**
5	68.0%	51.17	0.909	4.98/8	104
6	70.0%	62.94	0.933	5.31/8	89
7	76.0%	51.92	0.948	5.61/8	113
8	**82.0%**	58.74	**0.964**	**6.18/8**	109

Table 4 shows that sharing RoPE positions between each mask token and its paired data token is crucial for valid target verification. Without aligned RoPE, GSM8K accuracy drops from 84.0% to 0.0%, while the token-level acceptance rate also decreases substantially from 0.902 to 0.718. The large KL divergence of 7.12 further indicates that physically continued position ids break the intended temporal verification geometry.

Table 4: RoPE Alignment Ablation. We evaluate SDAR-1.7B-Chat as the draft model and SDAR-8B-Chat as the target model on GSM8K. Acc. denotes accuracy, and KL denotes KL divergence averaged over the target output distributions.

Setting	**GSM8K Acc.**	**Token-level α**	**Accepted / Block**	**KL(Default $\|$ $\cdot$)**
With Aligned RoPE	**84.0%**	**0.902**	**3.61/4**	–
Without Aligned RoPE	0.0%	0.718	2.87/4	7.12

5 Conclusion

In this paper, we propose **SimSD**, a simple but effective plug-and-play strategy that restores token-level temporal structure for supporting dLLMs to use speculative decoding for inference acceleration. By introducing reference tokens from draft-model predictions and controlling their interactions with current-step tokens through a tailored attention mask, our method enables dLLMs to compute valid logits for drafted tokens under temporally consistent contexts. This brings the core verification capability of AR speculative decoding, where multiple drafted tokens are verified in parallel by the target model, to dLLM inference while preserving their parallel generation advantages. Since our method only modifies the attention mask and copies a small number of hidden states, it can be seamlessly integrated into existing dLLM decoding pipelines and combined with acceleration techniques such as KV caching and blockwise decoding. Extensive experiments on SDAR-family dLLMs across GSM8K, MBPP, TriviaQA, and MMLU demonstrate the effectiveness of our method. Without additional training, it achieves up to $7.46\times$ higher decoding throughput than the vanilla SDAR baseline with tensor parallelism, while maintaining generation quality and improving average task performance by $+1.7\%$.

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

References

[1] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

[2] Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

[3] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

[4] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models, 2023. URL https://arxiv.org/abs/2303.18223.

[5] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2024. URL https://arxiv.org/abs/2402.06196.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[7] Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. Break the sequential dependency of LLM inference using lookahead decoding. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 14060–14079. PMLR, 2024. URL https://proceedings.mlr.press/v235/fu24a.html.

[8] Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7655–7671, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.456. URL https://aclanthology.org/2024.findings-acl.456/.

[9] 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., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/958c530554f78bcd8e97125b70e6973d-Paper.pdf.

[10] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/1be5bc25d50895ee656b8c2d9eb89d6a-Abstract-Conference.html.

[11] Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, et al. Mercury: Ultra-fast language models based on diffusion, 2025. URL https://arxiv.org/abs/2506.17298.

[12] Google DeepMind. Gemini diffusion - google deepmind, 2025. URL https://deepmind.google/models/gemini-diffusion/. Accessed: 2025-09-19.

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

[13] Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, Chengxi Li, Chongxuan Li, Jianguo Li, Zehuan Li, Huabin Liu, Lin Liu, Guoshan Lu, Xiaocheng Lu, Yuxin Ma, Jianfeng Tan, Lanning Wei, Ji-Rong Wen, Yipeng Xing, Xiaolu Zhang, Junbo Zhao, Da Zheng, Jun Zhou, Junlin Zhou, Zhanchao Zhou, Liwang Zhu, and Yihong Zhuang. LLaDA2.0: Scaling up diffusion language models to 100b, 2025. URL https://arxiv.org/abs/2512.15745.

[14] Jacob K. Christopher, Brian R. Bartoldson, Tal Ben-Nun, Michael Cardei, Bhavya Kailkhura, and Ferdinando Fioretto. Speculative diffusion decoding: Accelerating language generation through diffusion. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 12042–12059. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.601. URL https://aclanthology.org/2025.naacl-long.601/.

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

[16] Tianxiang Pan, Baitao Gong, Mo Guang, Hongwei Yong, Tianpeng Jiang, Yaqian Li, Zheng Cao, and Kaiwen Long. BlockSpec: Blockwise speculative decoding for diffusion llms, 2025. URL https://openreview.net/forum?id=hmAviop5rm. OpenReview preprint.

[17] Jian Chen, Yesheng Liang, and Zhijian Liu. DFlash: Block diffusion for flash speculative decoding, 2026. URL https://arxiv.org/abs/2602.06036.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics, 2019. URL https://aclanthology.org/N19-1423.

[19] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KnqiC0znVF.

[20] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

[21] Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling, 2023. URL https://arxiv.org/abs/2302.01318.

[22] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

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

[24] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

[25] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

[26] Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. DiffusionBERT: Improving generative masked language models with diffusion models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4521–4534, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.248. URL https://aclanthology.org/2023.acl-long.248/.

[27] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal large diffusion language models, 2025. URL https://arxiv.org/abs/2505.15809.

[28] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7B: Diffusion large language models, 2025. URL https://arxiv.org/abs/2508.15487.

[29] Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=j1tSLYKwg8. arXiv:2410.17891.

[30] Zhihui Xie, Jiacheng Ye, Lin Zheng, Jiahui Gao, Jingwei Dong, Zirui Wu, Xueliang Zhao, Shansan Gong, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream-Coder 7B: An open diffusion language model for code, 2025. URL https://arxiv.org/abs/2509.01142.

[31] 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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tyEyYT267x.

[32] Zhanqiu Hu, Jian Meng, Yash Akhauri, Mohamed S. Abdelfattah, Jae sun Seo, Zhiru Zhang, and Udit Gupta. FlashDLM: Accelerating diffusion language model inference via efficient kv caching and guided diffusion, 2025. URL https://arxiv.org/abs/2505.21467.

[33] Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang, and Linfeng Zhang. dLLM-Cache: Accelerating diffusion large language models with adaptive caching, 2025. URL https://arxiv.org/abs/2506.06295.

[34] Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dKV-Cache: The cache for diffusion language models, 2025. URL https://arxiv.org/abs/2505.15781.

[35] Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling KV cache and parallel decoding. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3Z3Is6hnOT.

[36] Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunan Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. SpecInfer: Accelerating generative large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, 2024. doi: 10.1145/3620666.3651335. URL https://dl.acm.org/doi/10.1145/3620666.3651335.

[37] Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 5209–5235. PMLR, 2024. URL https://proceedings.mlr.press/v235/cai24b.html.

[38] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 28935–28948. PMLR, 2024. URL https://proceedings.mlr.press/v235/li24bt.html.

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

[39] Jameson Sandler, Jacob K. Christopher, Thomas Hartvigsen, and Ferdinando Fioretto. SpecDiff-2: Scaling diffusion drafter alignment for faster speculative decoding, 2025. URL https://arxiv.org/abs/2511.00606.

[40] Guanghao Li, Zhihui Fu, Min Fang, Qibin Zhao, Ming Tang, Chun Yuan, and Jun Wang. DiffuSpec: Unlocking diffusion language models for speculative decoding, 2025. URL https://arxiv.org/abs/2510.02358.

[41] Yifeng Gao, Ziang Ji, Yuxuan Wang, Biqing Qi, Hanlin Xu, and Linfeng Zhang. Self speculative decoding for diffusion large language models, 2025. URL https://arxiv.org/abs/2510.04147.

[42] Ligong Han, Hao Wang, Han Gao, Kai Xu, and Akash Srivastava. S2D2: Fast decoding for diffusion llms via training-free self-speculation, 2026. URL https://arxiv.org/abs/2603.25702.

[43] Sudhanshu Agrawal, Risheek Garrepalli, Raghavv Goel, Mingu Lee, Christopher Lott, and Fatih Porikli. Spiffy: Multiplying diffusion llm acceleration via lossless speculative decoding, 2025. URL https://arxiv.org/abs/2509.18085.

[44] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding, 2021. URL https://arxiv.org/abs/2104.09864.

[45] OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023.

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

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: See our abstract and introduction.

   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: Our method currently applies the proposed token-level causal mask only during inference. Since the models are not trained with this masking pattern, their behavior may not be fully optimized for our verification process. Training or fine-tuning with the proposed mask is an interesting direction for future work.

   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?

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Answer: [Yes]

Justification: The assumptions and proof details for our theoretical result are provided in Section 3.3. We also include the necessary conditions under which the result holds, together with the corresponding derivation.

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: The information needed to reproduce the main experimental results is described in Section 4, including the experimental setup and evaluation details relevant to our main claims.

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.

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

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 will provide the code and data files needed to reproduce the main experimental results, together with the experimental setup and evaluation details relevant to our main claims.

   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: All experimental details are provided in Section 4.1, including the data splits, key hyperparameters, evaluation settings, and other information needed to understand the results.

   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: Section 4 provides a detailed analysis of the experimental results, including appropriate information for interpreting their statistical significance.

   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.

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

- 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: Section 4 provides the computer resource details needed to reproduce the experiments, including the compute environment, memory usage, and execution time where applicable.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in this paper conforms to the NeurIPS Code of Ethics in all relevant respects.

   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: [No]

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Justification: This paper focuses on improving the inference efficiency of diffusion language models, and we do not identify direct societal impacts that are specific to the proposed method.

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: This question is not applicable to our work, as the paper does not release high-risk data or models that require additional safeguards for responsible release.

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 properly credit the creators and original owners of the assets used in this paper, and we explicitly mention and respect the relevant licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

- 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: The new assets introduced in this paper are documented, and the relevant documentation will be provided alongside the released assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: This question is not applicable, as our work does not involve crowdsourcing experiments or 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]

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.

Justification: This question is not applicable, as our work does not involve study participants or human-subject research.

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: [Yes]

Justification: Since our method is designed for language model inference, LLMs are a core component of the research. We describe the model usage, decoding procedure, proposed verification mechanism, and evaluation setup in Sections 3 and 4.

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.

Confidential reviewer copy. This manuscript is submitted to the 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Unauthorized sharing, redistribution, or disclosure is strictly prohibited.