A Tale of Two Temperatures: Simple, Efficient, and Diverse Sampling from Diffusion Language Models

Theo X. Olausson [1 *] **Metod Jazbec** [2 *] **Xi Wang** [3] **Armando Solar-Lezama** [1] **Christian A. Naesseth** [2]
Stephan Mandt [4] **Eric Nalisnick** [3]

Abstract

Much work has been done on designing fast and accurate sampling for diffusion language models (dLLMs). However, these efforts have largely focused on the tradeoff between speed and quality of *individual* samples; how to additionally ensure *diversity* across samples remains less well understood. In this work, we show that diversity can be increased by using softened, tempered versions of familiar confidence-based remasking heuristics, retaining their computational benefits and offering simple implementations. We motivate this approach by introducing an idealized formal model of *fork tokens* and studying the impact of remasking on the expected entropy at the forks. Empirically, the proposed tempered heuristics close the exploration gap (pass@k) between existing confidence-based and autoregressive sampling, hence outperforming both when controlling for cost (pass@NFE). We further study how the increase in diversity translates to downstream post-training and test-time compute scaling. Overall, our findings demonstrate that simple, efficient, and diverse sampling from dLLMs is possible.

1. Introduction

Diffusion language models (dLLMs) have recently seen significant uptake, offering native bidirectional context and the ability to generate multiple tokens in parallel (Nie et al. 2025b; Ye et al. 2025; Yang et al. 2025; Bethune et al. 2026). Unlike in the autoregressive (AR) setting, dLLMs require a *remasking* strategy that determines the order in which tokens are sampled. As this choice significantly affects sample quality, much prior work has focused on optimizing the speed–quality tradeoff for individual samples (Kim et al. 2025a; Ben-Hamu et al. 2025; Jazbec et al. 2025).

While this line of work has helped make dLLMs more competitive with AR LLMs, efficiently obtaining a *single* high-quality sample is not always sufficient: post-training, test-time compute scaling, and A/B testing all require *diversity* across samples in order to be effective. Surprisingly, Ni et al. (2026) found that confidence-based heuristics yielding high single-sample accuracy lead to worse diversity than left-to-right autoregressive generation. By delaying uncertain tokens, such heuristics avoid genuine branching points in the reasoning process. Autoregressive generation does not suffer from this problem, but comes at a significant cost as it forgoes parallel generation of multiple tokens altogether.

In this work, we argue that the diversity problem can be addressed with a simple fix, without abandoning parallel generation or adopting complex sampling schemes. Our key observation is that it is the greedy nature of existing heuristics, not parallel generation in itself, that limits diversity. By replacing them with softened, tempered approximations thereof—for example, sampling positions proportional to their tempered confidence instead of selecting them greedily based on the highest confidence—we demonstrate that the diversity can be recovered while preserving speed and simplicity. In summary, our contributions are as follows:

- We introduce *Tempered Low-Confidence* (TLC) and *Tempered Confidence Thresholding* (TCT) remasking, stochastic relaxations of two widely used remasking heuristics (Nie et al., 2025b; Wu et al., 2025) controlled by a *position temperature* T_{pos} (Section 3.2). This provides a knob complementary to the standard token-level temperature T_{token}: T_{token} governs the entropy of what token is predicted at each position, while T_{pos} governs how stochastic the order in which positions are committed to is.
- We give an operational definition of the *fork tokens*, high-entropy positions that prior work has empirically found to be important for effective post-training, and

[*]Equal contribution [1]Massachusetts Institute of Technology [2]UvA-Bosch Delta Lab, University of Amsterdam [3]Johns Hopkins University [4]University of California, Irvine. Correspondence to: Metod Jazbec <m.jazbec@uva.nl>, Theo X. Olausson <theoxo@csail.mit.edu>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

show that, under an idealized model, increasing T_{pos} provably increases the expected entropy at these forks for TLC (Section 3.1).

- We empirically validate that TLC recovers pass@k scaling comparable to AR rollouts across multiple benchmarks and models, confirming that tempering the remasking strategy is sufficient to restore diversity (Section 4.1, Figure 2). TCT goes further by retaining the adaptive speed of confidence thresholding; when the cost of each rollout is accounted for, TCT often scales better than either untempered heuristics or AR generation (Section 4.1, Figure 3).
- We study how pass@k improvements translate downstream: when used for GRPO rollout generation in dLLM post-training (Zhao et al., 2025) at matched computational budgets, TCT yields stronger policies than AR generation (Section 4.3). In the context of test-time compute, we find that while the increased diversity of TCT can overwhelm simple strategies such as majority voting, this gap is largely closed when using outcome reward models for answer selection (Section 4.2).

2. Background

We use the notation [L] to represent the set $\{1, \dots, L\}$. We denote a sequence of tokens by $\boldsymbol{x} = (x_1, \dots, x_L) \in \mathcal{V}^L$, where $\mathcal{V} := [V]$ is the vocabulary and L is the sequence length.

2.1. From Masked Diffusion Models to Diffusion Language Models

Masked Diffusion Models (MDMs) are a form of absorbing discrete diffusion (Sohl-Dickstein et al., 2015; Austin et al., 2021a) in which the target (noise) distribution places all its mass on a single token, `[MASK]`, and the forward process gradually corrupts each token towards the absorbing mask state by transitioning tokens independently through a Markov chain. Formally, in the continuous-time formulation of Campbell et al. (2022), this can be characterized compactly as follows:

$$\begin{aligned} \boldsymbol{x}_0 \sim p_{\text{data}}, \qquad \boldsymbol{x}_1 \sim \delta_{\texttt{[MASK]}}^{\otimes L} \\ \boldsymbol{x}_t \sim p_t(\cdot \mid \boldsymbol{x}_0) \triangleq \textstyle\prod_{l=1}^{L} \alpha(t)\mathbf{1}\left(x_t^l = \texttt{[MASK]}\right) \\ + (1-\alpha(t))\mathbf{1}\left(x_t^l = x_0^l\right) \end{aligned} \tag{1}$$

where $0 \le t \le 1$ denotes the time, $\alpha : t \to [0,1]$ is a monotonically non-decreasing function in t with endpoints $\alpha(0) = 0, \alpha(1) = 1$, and $\delta_{\texttt{[MASK]}}^{\otimes L}$ is a product of L Dirac measures placing all of their mass on `[MASK]`. The generative (i.e., reverse) process can be modeled either by learning to predict the time-dependent reverse transition ratios (Campbell et al., 2022; Meng et al., 2022; Lou et al., 2024), or by training a model q_θ to directly approximate the conditional distribution $p_{0|t}$. The latter strategy has typically been the choice of language-modeling practitioners, as later work (Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2025) showed that its evidence lower bound (ELBO) simplifies to:

$$-\mathcal{L}(\theta) \triangleq \mathbb{E}\left[w(t) \sum_{k=1}^{L} \mathbf{1}\left(x_t^k = \texttt{[MASK]}\right) \log q_\theta^k(x_0^k \mid \boldsymbol{x}_t) \right] \tag{2}$$

with the expectation taken over $t \sim U[0,1]$, $\boldsymbol{x}_0 \sim p_{\text{data}}$, and $\boldsymbol{x}_t \sim p_t(\cdot \mid \boldsymbol{x}_0)$, where $w(\cdot)$ is a weighting term determined by the noise schedule $\alpha(\cdot)$, and $q_\theta^k(\cdot|\boldsymbol{x}_t)$ is the model's predicted *marginal* distribution over possible tokens at the k-th position. This formulation eliminates explicit conditioning on t, connecting MDM training to masked language modeling (Devlin et al., 2019; Hoogeboom et al., 2021). Combined with empirically validated scaling laws for MDMs (Nie et al., 2025a; Bethune et al., 2026), this has enabled the open-source community to train multi-billion parameter MDMs on textual data, which we refer to as *diffusion (large) language models* (dLLMs).

2.2. Remasking strategies and sampling from dLLMs

Early work on MDMs obtained samples by discretizing the reverse process and simulating the Markov chain (Campbell et al., 2022; Austin et al., 2021a; Lou et al., 2024). As observed by Zheng et al. (2025), this is potentially wasteful as each individual token transitions exactly once in Equation (1). They instead proposed a *first-hit* sampling scheme, which when the time-agnostic objective of Equation (2) is used simplifies to what Nie et al. (2025b) call *random remasking*. Concretely, denoting the current generation as $\boldsymbol{x}_t \in \mathcal{V}^L$ with still-masked positions $\mathcal{M}_t := \{k \in [L] \mid x_t^k = \texttt{[MASK]}\}$, the next generation $\boldsymbol{x}_{t'}$ is obtained via

$$x_{t'}^k := \begin{cases} x \sim q_\theta^k(\cdot \mid \boldsymbol{x}_t\,; T_{\text{token}}), & \text{if } k \in \mathcal{U}_t^K, \\ x_t^k\,, & \text{otherwise.} \end{cases}$$

where the *unmasking set* $\mathcal{U}_t^K$ of $K \ge 1$ positions is sampled uniformly at random (without replacement) from $\mathcal{M}_t$. As in the autoregressive setting, T_{token} tempers the final softmax of q_θ^k: intuitively, higher temperatures yield more diverse (but possibly lower quality) generations. This process continues until reaching a time where $\mathcal{M}_t = \emptyset$.

Later work revealed that in practice, test-time performance can often be improved by using *heuristic* remasking strategies that select $\mathcal{U}_t$ based on distributional information about the model's predictions. Two particularly popular choices use the model's per-position confidence $c_t^k \triangleq \max_v q_\theta^k(v \mid$

$\boldsymbol{x}_t$):[1] *low-confidence* (LC; Chang et al. 2022; Nie et al. 2025b) remasking unmasks the K most confident positions, while *confidence thresholding* (CT; Wu et al. 2025) unmasks all positions exceeding a fixed threshold $\lambda \in [0, 1]$:

$$\text{LC:} \quad \mathcal{U}_t^K := \left\{ \underset{I \subseteq \mathcal{M}_t,\ |I|=K}{\arg\max} \sum_{k \in I} c_t^k \right\} \tag{3}$$

$$\text{CT:} \quad \mathcal{U}_t^\lambda := \{k \in \mathcal{M}_t \mid c_t^k > \lambda\} \tag{4}$$

The key distinction is that low-confidence remasking unmasks a fixed number of tokens per step (K), while confidence thresholding is *adaptive*: $|\mathcal{U}_\mathcal{M}^\lambda|$ varies with the model's confidence, allowing sampling to proceed quickly when the model is certain and slowly when it is not.

2.3. Sample diversity and the role of remasking

The remasking strategies discussed above are typically compared in terms of accuracy at a given computational budget under greedy, $T_{\text{token}} = 0$, decoding (Nie et al., 2025b; Wu et al., 2025; Ben-Hamu et al., 2025). However, many practical applications require diverse *sets* of samples; in this case the more important metric becomes pass@k, the probability that at least one of k independent samples solves a given problem (Kulal et al., 2019; Chen et al., 2021). Ni et al. (2026) investigated pass@k scaling for dLLMs and found that low-confidence remasking, despite its strong greedy performance, exhibits notably flatter pass@k curves than autoregressive (left-to-right) decoding. They attribute this to a phenomenon they dub *entropy degradation*, where low-confidence systematically defers high-entropy branching positions until surrounding context has already been established, collapsing the entropy that would otherwise enable qualitatively diverse samples. Autoregressive decoding, Ni et al. (2026) argue, sidesteps this by resolving tokens in a fixed positional order agnostic to confidence, forcing the model to commit to branching points as they arise.

3. Methods

We now introduce our approach for generating diverse and efficient samples with dLLMs. We first study Ni et al. (2026)'s empirical findings through the lens of a formal model of *fork tokens* and their relationship to semantic sample diversity. (Section 3.1). Motivated by our observations, we then propose to temper confidence-based heuristics via a new position temperature T_{pos}, and discuss two concrete instantiations of this strategy (Section 3.2). Figure 1 shows an overview of our modeling framework and methodology.

[1]For non-greedy sampling we define the confidence as $c_t^k \triangleq q_\theta^k(x \mid \boldsymbol{x}_t),\ x \sim q_\theta^k(\cdot \mid \boldsymbol{x}_t\,; T_{\text{token}})$.

3.1. Fork tokens and the anchor-fork degeneracy

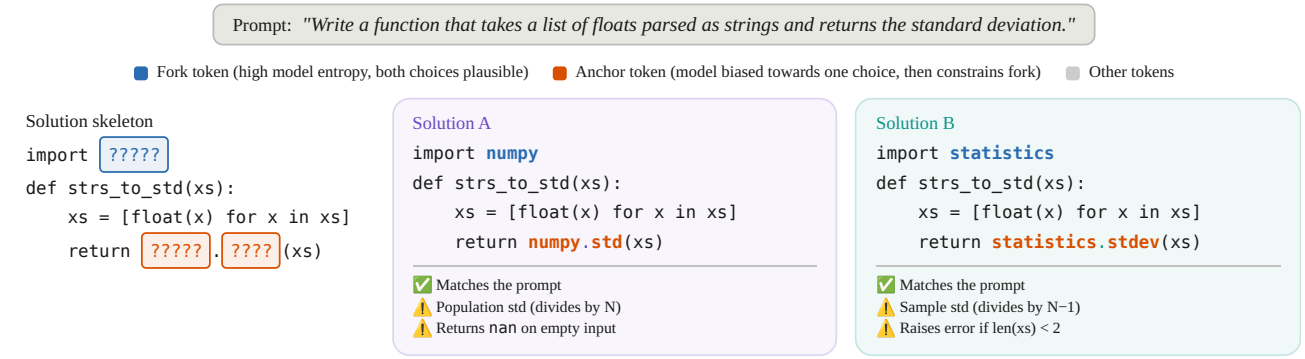
Ni et al. (2026)'s empirical study of entropy degradation is grounded in the observational definition of *forking tokens* given by prior work (see Section A). These works identify forks as positions in the sequence that have high entropy, and observe that maintaining diversity at these positions translates to effective post-training (for AR models).

To allow us to reason more precisely about this phenomenon, we develop a formal model in Appendix D. Key to our analysis is an *operational* definition of fork tokens as positions in the sequence whose value sharply bounds the set of possible semantic outcomes under the data distribution (Definition D.1), allowing us to formally connect their uncertainty to the semantic sample diversity (Equation (5)). To connect this machinery to Ni et al. (2026)'s observations of the *sampling distribution* induced by pairing q_θ with low-confidence or autoregressive remasking, we build on their intuition that the root cause of entropy degradation is that contextual tokens surrounding the fork tend to dominate it in confidence, and that revealing them collapses the fork's entropy before it is ever sampled. We capture this intuition in what we call the *anchor-fork degeneracy* model (Assumption D.2), where the sampling distribution's uncertainty about the fork token is determined by the set of high-confidence *anchor tokens* that have been revealed so far. This phenomenon is visualized in the upper half of Figure 1.

This simple model recovers all three of Ni et al. (2026)'s empirical observations: (i) LC yields minimal fork entropy; (ii) adjusting T_{token} does not help with preserving entropy at forks, since it does not change the unmasking *order* (Proposition D.8); and (iii) left-to-right generation yields strictly higher fork entropy than LC whenever at least one anchor follows the fork in positional order (Proposition D.9). Additionally, however, it also yields a new prediction: autoregressive generation is not special, and *any* strategy that increases the probability of resolving a fork before its anchors will retain higher fork entropy than LC. This suggests that semantic diversity can be improved without sacrificing the efficiency of parallel generation.

3.2. Tempered remasking heuristics

The formal analysis of Appendix D suggests a simple fix for confidence heuristics: rather than deterministically selecting positions to unmask, *sample* them in proportion to a softened distribution determined by their confidences. This distribution can then be controlled through a *position temperature* T_{pos} that provides a complementary knob to T_{token}: T_{token} governs how greedily we determined *what* is sampled at each position, while T_{pos} governs how greedily we determined the *order* in which to do so. We next describe concrete instantiations corresponding to each heuristic from Section 2.2.



Remasking strategy	Token sampling order	Semantically diverse samples	Parallel generation
Deterministic Confidence-based	import def ··· numpy std ··· numpy [...] ···	✗	✓
Autoregressive (Left-to-Right)	import statistics ··· statistics . stdev (xs)	✓	✗
Tempered Confidence-based (ours)	import def ··· statistics [...] ··· statistics stdev ···	✓	✓

Figure 1. An example illustrating our model of entropy degradation. The **fork token** (Definition D.1) has high marginal uncertainty under the dLLM, indicating that both the `numpy` and `statistic` solutions are plausible. However, the dLLM is simultaneously overly confident in its prediction at the **anchor tokens** in the function body, biasing low-confidence remasking towards one solution as it reveals those tokens first. Tempered remasking allows the fork to still be revealed first, increasing semantic diversity without sacrificing parallelism. Note that while both paths lead to generations that match the prompt (✅), they have subtly different semantics (⚠️); sampling both is important to achieve good coverage.

Tempered low-confidence (TLC) remasking (Algorithm 2). Rather than deterministically selecting $\mathcal{U}_t^K$ as in Equation (3), TLC samples K positions (without replacement) from a tempered distribution induced by the token confidences:

$$\mathcal{U}_t^{K,T_{\text{pos}}} \overset{\text{w/o repl.}}{\sim} \text{Cat}(\tilde{c}_t),\ \tilde{c}_t^k \propto (c_t^k)^{1/T_{\text{pos}}} \cdot \mathbf{1}\,(k \in \mathcal{M}_t)$$

where $T_{\text{pos}} > 0$ is a free parameter. As $T_{\text{pos}} \downarrow 0$, TLC recovers deterministic low-confidence remasking; as $T_{\text{pos}} \uparrow \infty$, it approaches uniform (random) remasking.

In Appendix D we prove that TLC resolves the anchor-fork degeneracy: under the idealized model, sufficiently large increases of T_{pos} provably increase the expected fork-token entropy (Proposition D.4); combined with an assumption that q_θ captures the data distribution well enough to preserve the fork structure (Assumption D.6), this yields a direct increase in semantic diversity, and thus pass@k, whenever the fork-entropy gain is large enough (Corollary D.7). [2]

While TLC offers a mechanism to recover sample diversity, it does not yield the *adaptive* speed-ups that make confidence-based heuristics particularly attractive. This motivates applying the same tempering idea to confidence thresholding.

Tempered confidence thresholding (TCT) remasking (Algorithm 4). TCT replaces the hard threshold in Equation (4) with stochastic inclusion:

$$\mathcal{U}_t^{\lambda,T_{\text{pos}}} = \{k \in \mathcal{M}_t \mid b_t^k = 1\},\ b_t^k \sim \text{Ber}\big(\sigma\big((c_t^k - \lambda)/T_{\text{pos}}\big)\big)$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. This recovers the hard threshold at λ as $T_{\text{pos}} \downarrow 0$, while yielding increasingly stochastic (and λ-agnostic) behavior as T_{pos} increases. Intuitively, at moderate temperatures, λ controls the speed of generation while T_{pos} controls the degree of diversity: under hard thresholding, a fork token whose confidence falls below λ is deterministically deferred; under TCT, the sigmoid gives it nonzero probability of being unmasked at every step, breaking the systematic deferral. TCT thus retains the adaptive efficiency of confidence thresholding while introducing the same diversity benefits as TLC.

4. Experiments

We begin our experiments by verifying that our proposed tempered samplers can close the gap to autoregressive sampling in terms of pass@k and also outperform it when the cost of generation is taken into account (Section 4.1). Next, we investigate whether the improvements in diversity translate to improved downstream performance for test-time compute (Section 4.2). Finally, we examine the implications of tempering when generating rollouts during RL post-training of dLLMs (Section 4.3).

4.1. Tempered heuristics recover pass@k scaling

We start by investigating the core hypothesis that tempering the heuristics can recover pass@k scaling compared to autoregressive sampling proposed in (Ni et al., 2026).

[2] In practice, excessively increasing T_{pos} risks revealing positions with less context than is needed to yield coherent generations (same as random unmasking), so for smaller values of k the optimal T_{pos} needs to be chosen to balance the diversity gain against this loss of quality (see Figure 6).

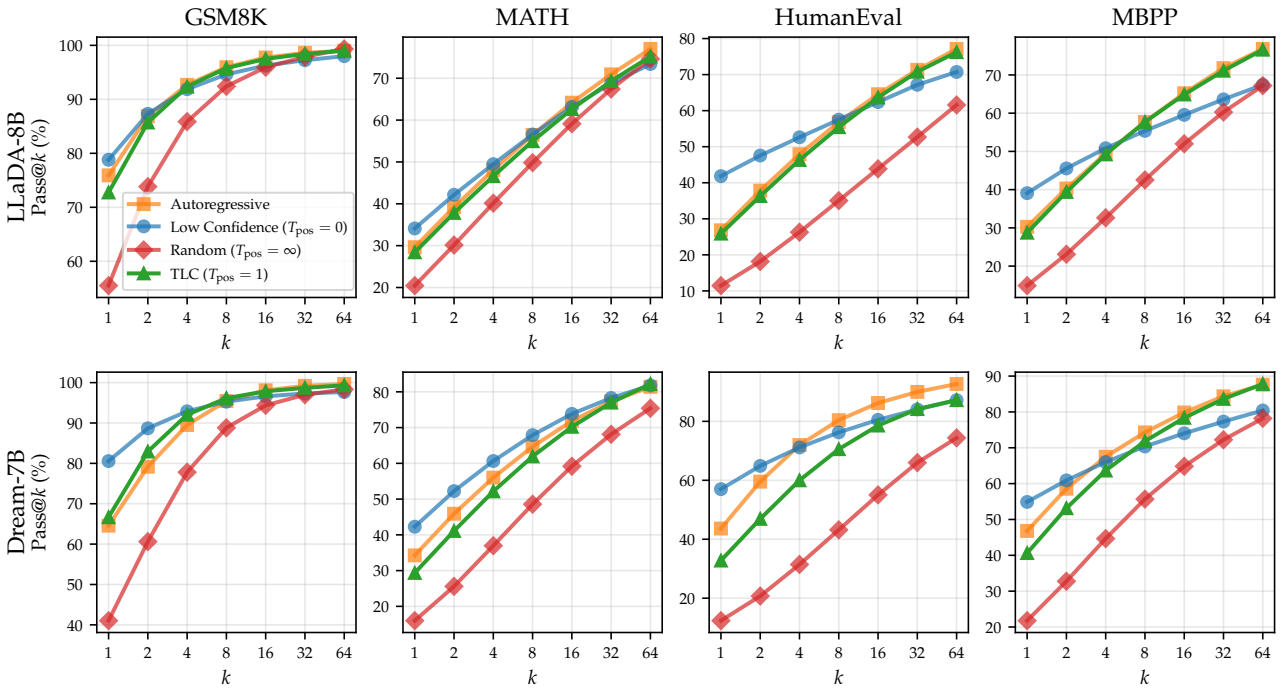


Figure 2. Pass@k for TLC ($T_{\text{pos}} = 1$) on LLaDA-8B-Instruct and Dream-7B-Instruct across four benchmarks. TLC closely tracks the AR baseline, recovering almost all of the diversity lost by low-confidence remasking.

Experimental setting. We evaluate our proposed remasking strategies on two open-source dLLMs: LLaDA-8B-Instruct (Nie et al., 2025b) and Dream-v0-Instruct-7B (Ye et al., 2025). We evaluate on 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). For GSM8k we evaluate on a subset of $N_{\text{test}} = 300$ test samples to reduce computational cost. Following prior work (Zhao et al., 2025), we use $T_{\text{token}} = 0.8$ as well as semi-AR/block-wise decoding with blocks of 32 tokens (Arriola et al., 2025; Nie et al., 2025b), at a sequence length of $L = 256$. Due to cost, and to avoid overfitting the position temperature to any one particular setting, we fix $T_{\text{pos}} = 1$ for TLC and $T_{\text{pos}} = 0.1$ for TCT.

We report pass@k, the standard measure of whether at least one of k samples solves a given problem (Kulal et al., 2019; Chen et al., 2021). When evaluating the adaptive remasking functions, we also report *pass@NFE*—the pass rate as a function of the cumulative NFEs required. This metric was also used in recent work by Shen et al. (2026), and avoids pass@k's bias towards methods that use more compute per sample (Olausson et al., 2024).

TLC recovers diversity at high sample counts. Figure 2 shows pass@k for TLC across all four benchmarks, as well as for autoregressive, low-confidence and random baselines. We first note that our results echo the findings of Ni et al. (2026), as the low-confidence baseline underperforms autoregressive generation for both models, on all four datasets. In contrast, despite the lack of task- or model-specific tuning of T_{pos}, TLC closely tracks the AR baseline in all settings other than for Dream on HumanEval. These findings confirm that tempering the remasking heuristic can restore rollout diversity. Moreover, while random baseline catches up to TLC and AR on math datasets, it exhibits a large gap on coding datasets (HumanEval, MBPP), indicating that too much randomness in sampling unmasking positions (i.e., $T_{\text{pos}} = \infty$) can hurt the performance.

In Figure 7, we additionally study how robust these findings are across different token temperatures, finding TLC to be the least dependent on the exact choice of T_{token}. Concretely, performance under AR sampling degrades severely when using $T_{\text{token}} > 1$, while low-confidence experiences a stronger drop in performance for $T_{\text{token}} < 1$ compared to the proposed TLC. Importantly, TLC exhibits larger performance improvements when increasing k compared to low-confidence sampling across all considered T_{token}, confirming the benefits of additional randomness (via $T_{\text{pos}} > 0$) on sampling diversity.

TCT: lower pass@k, but higher pass@NFE. We depict TCT sampling results for LLaDA in Figure 3 (see Figure 8 for Dream results). Looking only at pass@k (top row), TCT ($T_{\text{pos}} = 0.1$, $\lambda = 0.6$) slightly underperforms the AR baseline on all benchmarks, although it still beats out the deterministic confidence-thresholding ($\lambda = 0.6$; Wu et al. (2025)). However, when cost is taken into account (bottom row), TCT starts to shine, substantially outperform-

ing both baselines. As a concrete example, on MATH dataset, TCT requires around 4000 NFEs to achieve performance above 70%: a 2x speed-up compared to AR which requires around 8000 NFEs.[3] Notably, deterministic confidence-thresholding alone is not sufficient to outperform autoregressive rollouts at higher NFE values, as it underperforms AR on HumanEval and MBPP; only TCT consistently comes out on top, highlighting the importance of tempering confidence-based heuristics to preserve diversity.

4.2. TCT for Test-time compute

We next turn our attention to test-time compute to check whether the observed improved diversity of our proposed TCT sampler translates to better downstream performance.

Experimental setting. We focus on evaluating LLaDA-8B-Instruct on GSM8k (same subset of $N_{\text{test}} = 300$ samples) and MATH-500. We study these datasets as they require the use of heuristic answer selection strategies, unlike in coding where pass@k already directly captures scaling with respect to a fixed test suite. We consider two widely used such strategies: (i) *self-consistency*, where the most common answer is selected (Wang et al., 2023), and (ii) *ORM*, where an (output) reward model is used to score each generation, and the one with the highest score is returned. For the ORM, we use AceMath-7B (Liu et al., 2024).

Results We report Best@NFE results in Figure 4. Focusing first on the self-consistency results, we observe that while TCT outperforms the AR baseline at lower NFE counts, it plateaus sooner and thus achieves a lower top performance. As depicted in Figure 10, we attribute this to TCT producing less peaked, higher-entropy predictive distributions when aggregating over k samples (see Figure 11 for concrete examples of such predictive distributions). However, when an outcome reward model (ORM) is used to select the sample, the performance trends from pass@NFE in the previous section are largely recovered, and both TCT and confidence-thresholding outperform AR baseline in terms of Best@NFE.

4.3. TCT for GRPO rollouts

Recall that our interest in pass@k (and pass@NFE) stems from it being a useful indicator of the performance gains that can obtained through policy optimization post-training (Yue et al., 2025). In this section, we thus investigate whether the pass@NFE improvements of TCT translate to better post-training results, when controlling for training cost.

Experimental setting. We focus on post-training LLaDA-8B-Instruct (Nie et al., 2025b) on mathematical datasets (GSM8k, MATH-500). We adopt the d1 reinforcement learning framework (Zhao et al., 2025), as it is widely used for post-training dLLMs and provides efficient (albeit biased) policy likelihood estimation requiring only a single NFE. For generating GRPO rollouts, we use either TCT (with $T_{\text{pos}} = 0.1$, $\lambda = 0.6$), deterministic confidence-thresholding (with $\lambda = 0.6$), or autoregressive rollouts. Since the considered samplers differ in generation speed, we ensure a fair comparison by training for a fixed duration rather than a fixed number of training steps or epochs. Specifically, all models are trained for 72 hours on a single node with 4×A100 GPUs. After training, we evaluate the resulting RL-trained models under greedy decoding ($T_{\text{token}} = 0$) with confidence-thresholding, varying the threshold $\lambda \in \{0.6, 0.8, 1.0\}$ to obtain accurary-NFEs pareto frontiers for each trained policy.

Results. In Figure 5, we show training rewards (left), the standard deviation of rewards within a group during training (middle), and evaluation results (right). In the training plots, we indicate the number of completed steps (within a fixed compute budget) using vertical dashed lines. We observe that TCT completes the most GRPO training steps. For example, on GSM8K, TCT completes around 44K steps compared to only around 21K steps for AR. This is a direct consequence of adaptive nature of samplers TCT and Fast-dLLM which generate multiple tokens in-parallel, hence leading to much faster generation of GRPO rollouts compared to one-token-at-a-time, left-to-right autoregressive sampling. Next, we observe that TCT exhibits lower training rewards compared to both AR and confidence-thresholding (Fast-dLLM). Note, however, that training rewards are not directly comparable, as different sampling strategies are used for rollout generation. Hence, the lower training reward of TCT can be directly attributed to its slightly lower pass@1 rates (Figure 3). More important than the average reward for RL is the diversity of rewards within a group of rollouts used to compute GRPO advantages. In the extreme case where all samples receive the same reward, all advantages are equal to zero, and there is no learning signal for RL to exploit. Encouragingly, we find that TCT helps preserve high rollout diversity, as it exhibits the highest standard deviation of rewards throughout training.

Lastly, at test time, we observe that the policies obtained under TCT or Fast-dLLM sampling yield better evaluation performance compared to AR-trained policies. This complements recent findings in the literature (Ni et al., 2026) by demonstrating that when training cost is taken into account, there is no clear advantage to using autoregressive rollouts for GRPO, at least when using the efficient likelihood estimator from Zhao et al. (2025). We leave for future work the investigation of whether our findings also hold when using unbiased, yet more computationally expensive, likelihood estimators (Turok et al., 2026; Ni et al., 2026).

[3] When measuring the NFEs for AR sampling, we count the number of tokens generated before ⟨eos⟩. For TCT/CT, we count NFEs until all L positions are unmasked.

5. Conclusion

We have introduced tempered remasking heuristics for dLLMs, showing that simple, stochastic relaxations controlled by a single temperature T_{pos} can recover the rollout diversity needed for effective post-training without sacrificing the advantages of parallel generation.

Use of AI assistance

We acknowledge the use of AI tools in the preparation of this manuscript. Specifically, we utilized language models (LMs) to assist with grammar and style editing, preparation of figures, general coding assistance, drafting and checking proofs, and other tasks. We have reviewed and edited the content to ensure accuracy and clarity, and we take full responsibility for the final version of the manuscript.

References

Arriola, M., Gokaslan, A., Chiu, J. T., Yang, Z., Qi, Z., Han, J., Sahoo, S. S., and Kuleshov, V. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.09573.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993. Curran Associates, Inc., 2021a. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/958c530554f78bcd8e97125b70e6973d-Paper.pdf.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., and Sutton, C. Program synthesis with large language models, 2021b. URL https://arxiv.org/abs/2108.07732.

Ben-Hamu, H., Gat, I., Severo, D., Nolte, N., and Karrer, B. Accelerated Sampling from Masked Diffusion Models via Entropy Bounded Unmasking, May 2025.

Bethune, L., Turrisi, V., Mlodozeniec, B. K., Lopez, P. R., Boominathan, L., Bhendawade, N., Shidani, A., Pelemans, J., Olausson, T. X., Hjelm, D., Dixon, P., Monteiro, J., Ablin, P., Banna, V., Blaas, A., Henderson, N., Noriy, K., Busbridge, D., Susskind, J., Cuturi, M., Belousova, I., Zappella, L., Webb, R., and Ramapuram, J. The design space of tri-modal masked diffusion models, 2026. URL https://arxiv.org/abs/2602.21472.

Bigelow, E., Holtzman, A., Tanaka, H., and Ullman, T. Forking paths in neural text generation. *arXiv preprint arXiv:2412.07961*, 2024.

Campbell, A., Benton, J., De Bortoli, V., Rainforth, T., Deligiannidis, G., and Doucet, A. A continuous time framework for discrete denoising models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 28266–28279. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b5b528767aa35f5b1a60fe0aaeca0563-Paper-Conference.pdf.

Chang, H., Zhang, H., Jiang, L., Liu, C., and Freeman, W. T. Maskgit: Masked generative image transformer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Chen, S., Jiao, J., Ratliff, L. J., and Zhu, B. dultra: Ultrafast diffusion language models via reinforcement learning, 2025. URL https://arxiv.org/abs/2512.21446.

Cheng, D., Huang, S., Zhu, X., Dai, B., Zhao, X., Zhang, Z., and Wei, F. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30377–30385, 2026.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Cui, G., Zhang, Y., Chen, J., Yuan, L., Wang, Z., Zuo, Y., Li, H., Fan, Y., Chen, H., Chen, W., Liu, Z., Peng, H.,

Bai, L., Ouyang, W., Cheng, Y., Zhou, B., and Ding, N. The entropy mechanism of reinforcement learning for reasoning language models, 2025. URL https://arxiv.org/abs/2505.22617.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12454–12465. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/67d96d458abdef21792e6d8e590244e7-Paper.pdf.

Huang, G., Xu, T., Wang, M., Yi, Q., Gong, X., Li, S., Xiong, R., Li, K., Jiang, Y., and Zhou, B. Low-probability tokens sustain exploration in reinforcement learning with verifiable reward. *arXiv preprint arXiv:2510.03222*, 2025a.

Huang, Z., Chen, Z., Wang, Z., Li, T., and Qi, G.-J. Reinforcing the diffusion chain of lateral thought with diffusion language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Jazbec, M., Olausson, T. X., Béthune, L., Ablin, P., Kirchhof, M., Monteiro, J., Turrisi, V., Ramapuram, J., and Cuturi, M. Learning unmasking policies for diffusion language models, 2025. URL https://arxiv.org/abs/2512.09106.

Kim, J., Shah, K., Kontonis, V., Kakade, S. M., and Chen, S. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=DjJmre5IkP.

Kim, S. H., Hong, S., Jung, H., Park, Y., and Yun, S.-Y. KLASS: KL-guided fast inference in masked diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=gOG9Zoyn4R.

Kulal, S., Pasupat, P., Chandra, K., Lee, M., Padon, O., Aiken, A., and Liang, P. S. Spoc: Search-based pseudocode to code. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/7298332f04ac004a0ca44cc69ecf6f6b-Paper.pdf.

Lamont, S., Walder, C., Montague, P., Dezfouli, A., and Norrish, M. Free lunch for pass@ k? low cost diverse sampling for diffusion language models. *arXiv preprint arXiv:2603.04893*, 2026.

Liu, Z., Chen, Y., Shoeybi, M., Catanzaro, B., and Ping, W. Acemath: Advancing frontier math reasoning with post-training and reward modeling. *arXiv preprint*, 2024.

Lou, A., Meng, C., and Ermon, S. Discrete diffusion modeling by estimating the ratios of the data distribution. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Meng, C., Choi, K., Song, J., and Ermon, S. Concrete score matching: Generalized score matching for discrete data. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 34532–34545. Curran Associates, Inc., 2022.

Ni, Z., Wang, S., Yue, Y., Yu, T., Zhao, W., Hua, Y., Chen, T., Song, J., Yu, C., Zheng, B., and Huang, G. The flexibility trap: Why arbitrary order limits reasoning potential in diffusion language models. *arXiv preprint arXiv:2601.15165*, 2026.

Nie, S., Zhu, F., Du, C., Pang, T., Liu, Q., Zeng, G., Lin, M., and Li, C. Scaling up masked diffusion models on text. In Yue, Y., Garg, A., Peng, N., Sha, F., and Yu, R. (eds.), *International Conference on Learning Representations*, volume 2025, pp. 82974–82997, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/

ce1c1ff5d94079dea348a2317a889281-Paper-Conference.pdf.

Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., and Li, C. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025b.

Olausson, T. X., Inala, J. P., Wang, C., Gao, J., and Solar-Lezama, A. Is self-repair a silver bullet for code generation? In *International Conference on Learning Representations (ICLR)*, 2024.

Ou, J., Nie, S., Xue, K., Zhu, F., Sun, J., Li, Z., and Li, C. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In Yue, Y., Garg, A., Peng, N., Sha, F., and Yu, R. (eds.), *International Conference on Learning Representations*, volume 2025, pp. 64972–65009, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a365e37c18fb91af547a2f0012a89e98-Paper-Conference.pdf.

Petrenko, A., Lipkin, B., Chen, K., Wijmans, E., Cusumano-Towner, M. F., Giryes, R., and Kraehenbuehl, P. Entropy-preserving reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=E8MR8jgEeZ.

Rojas, K., Lin, J., Rasul, K., Schneider, A., Nevmyvaka, Y., Tao, M., and Deng, W. Improving reasoning for diffusion language models via group diffusion policy optimization. *arXiv preprint arXiv:2510.08554*, 2025.

Sahoo, S. S., Arriola, M., Gokaslan, A., Marroquin, E. M., Rush, A. M., Schiff, Y., Chiu, J. T., and Kuleshov, V. 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.

Shen, Y., Feng, T., Han, J., Wang, W., Chen, T., Shen, C., Leskovec, J., and Ermon, S. Improving diffusion language model decoding through joint search in generation order and token space, 2026. URL https://arxiv.org/abs/2601.20339.

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.

Snell, C. V., Lee, J., Xu, K., and Kumar, A. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Turok, G., Sa, C. D., and Kuleshov, V. DUEL: Exact likelihood for masked diffusion via deterministic unmasking, 2026. URL https://arxiv.org/abs/2603.01367.

Wang, C., Rashidinejad, P., Su, D., Jiang, S., Wang, S., Zhao, S., Zhou, C., Shen, S. Z., Chen, F., Jaakkola, T., et al. Spg: Sandwiched policy gradient for masked diffusion language models. *arXiv preprint arXiv:2510.09541*, 2025a.

Wang, G., Turok, G., Schiff, Y., Arriola, M., and Kuleshov, V. d2: Improved techniques for training reasoning diffusion language models, 2026. URL https://arxiv.org/abs/2509.21474.

Wang, S., Yu, L., Gao, C., Zheng, C., Liu, S., Lu, R., Dang, K., Chen, X., Yang, J., Zhang, Z., et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025b.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://arxiv.org/abs/2203.11171.

Wang, Y., Yang, L., Li, B., Tian, Y., Shen, K., and Wang, M. Revolutionizing reinforcement learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*, 2025c.

Wu, C., Zhang, H., Xue, S., Liu, Z., Diao, S., Zhu, L., Luo, P., Han, S., and Xie, E. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding, 2025. URL https://arxiv.org/abs/2505.22618.

Wu, J., Wan, Z., Yu, X., Yang, Y., Huang, Y., Tsang, I., and You, Y. Time-annealed perturbation sampling: Diverse generation for diffusion language models. *arXiv preprint arXiv:2601.22629*, 2026.

Yang, L., Tian, Y., Li, B., Zhang, X., Shen, K., Tong, Y., and Wang, M. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025.

Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., and Kong, L. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., Liu, X., Lin, H., Lin, Z., Ma, B., Sheng, G., Tong, Y., Zhang, C., Zhang, M., Zhang, W., Zhu, H., Zhu, J., Chen, J., Chen, J., Wang, C., Yu, H., Song, Y., Wei, X., Zhou, H., Liu, J., Ma, W.-Y., Zhang, Y.-Q., Yan, L., Qiao, M., Wu, Y., and Wang, M. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Yue, Y., Song, S., and Huang, G. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Zhao, H., Liang, D., Tang, W., Yao, D., and Kallus, N. DiFFPO: Training diffusion llms to reason fast and furious via reinforcement learning, 2026. URL https://arxiv.org/abs/2510.02212.

Zhao, S., Gupta, D., Zheng, Q., and Grover, A. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*, 2025.

Zheng, K., Chen, Y., Mao, H., Liu, M.-Y., Zhu, J., and Zhang, Q. Masked Diffusion Models are Secretly Time-Agnostic Masked Models and Exploit Inaccurate Categorical Sampling, April 2025.

A. Related Work

Diverse generation in dLLMs While sample diversity is well-studied in the autoregressive setting, it has received comparatively little attention in the context of dLLMs. Ni et al. (2026) and Shen et al. (2026) both study how the generation order impacts the reachable solution space, finding that confidence-based heuristics collapse diversity; however, neither proposes a sampling strategy that recovers diversity while retaining the efficiency of parallel generation. Concurrently, Wu et al. (2026) propose TAPS, which injects time-dependent noise into the conditioning signal to encourage semantic branching in early denoising steps, and Lamont et al. (2026) repel samples in feature space during generation to penalize within-batch redundancy. Both are training-free and effective, but operate on different axes than our approach where we modify the remasking order itself. The three approaches are thus complementary.

Post-training dLLMs. Most work on adapting policy optimization to dLLMs has focused on tackling the intractability of dLLM's likelihood (Zhao et al. 2025; Rojas et al. 2025; Wang et al. 2025c;a; 2026; Turok et al. 2026; *inter alia*), while the impact of remasking strategy on policy exploration has received less attention thus far. Most closely related to our work, Ni et al. (2026) observed a tension between efficiency and diversity in deterministic confidence-based samplers, and instead proposed using AR sampling to encourage exploration in dLLMs. Crystallizing their observation into a formal model directly led to the alternative solution of tempered samplers we propose in this work, which improve sample diversity without sacrificing generation speed.

Remasking strategies for dLLMs. Building on the simplicity of the first-hit sampler proposed by Zheng et al. (2025), a variety of heuristic remasking strategies have been proposed. Fixed-budget methods choose a pre-determined number of positions after ranking candidates by confidence (Chang et al., 2022; Nie et al., 2025b), prediction margin (Kim et al., 2025a), or KL stability across steps (Kim et al., 2025b), while adaptive methods unmask variable-size sets via confidence thresholds (Wu et al., 2025) or entropy bounds (Ben-Hamu et al., 2025). Orthogonally, some works have sought to *learn* remasking strategies (Huang et al., 2025b; Jazbec et al., 2025; Chen et al., 2025; Zhao et al., 2026). Common to all of the above is their focus on the accuracy–speed tradeoff for individual samples under greedy decoding, in contrast to our focus on diversity.

The role of sample diversity in language modeling. Sample diversity has been found to be a critical component in multiple stages of the LLM scaling pipeline: Yue et al. (2025) showed that RL accuracy gains are bounded by base-model pass@k, and test-time scaling requires meaningfully distinct candidates for strategies such as self-consistency (Wang et al., 2023) and Best-of-N (Snell et al., 2025). In autoregressive models, maintaining diversity during post-training has required indirect interventions such as entropy regularization (Yu et al., 2025; Cui et al., 2025; Petrenko et al., 2026), entropy-based advantage shaping (Cheng et al., 2026), or selective regularization of low-probability exploratory tokens (Huang et al., 2025a); we show that in the case of dLLMs, the flexible generation order itself provides an alternative lever.

Fork tokens. Bigelow et al. (2024) studied forking tokens in neural text generation, showing that resampling a small set of high-impact tokens can induce qualitatively different continuations. More recently, Wang et al. (2025b) popularized the term *forking tokens* to describe high-entropy tokens that steer reasoning, and demonstrated their crucial role for effective RLVR of autoregressive LLMs. Ni et al. (2026) investigated forks in dLLMs and empirically showed that their entropy degrades under popular confidence-based samplers. In our work, we formalize these fork tokens operationally to better understand how the choice of remasking strategy impacts sample diversity for dLLMs.

B. Limitations and Future work

Our work does not provide a mechanism for how to tune T_{pos} to optimally balance sample quality and diversity, and except for Figure 6 we do not empirically analyze our method's sensitivity to the parameter. Moreover, we observe that although TCT strongly outperforms Fast-dLLM sampling (Wu et al., 2025) in our pass@NFE experiments (Figure 3), we find that the two often perform comparably on downstream tasks such as test-time compute (Figure 4) and RL post-training (Figure 5). Given that prior work has shown a connection between pass@k and success in these settings, it would be valuable for future work to investigate why this is not reflected in d1-style (Zhao et al., 2025) GRPO for dLLMs. Finally, our formal results rest on idealized assumptions (Appendix D), notably that fork-token entropy degrades linearly in revealed anchors (Assumption D.2) and that q_θ perfectly preserves the data distribution's fork structure (Assumption D.6); furthermore, we only theoretically study TLC at $K = 1$, leaving a theoretical gap to parallel generation, which has been studied elsewhere (Wu et al., 2025; Ben-Hamu et al., 2025).

C. Algorithms

Algorithm 1 Low-Confidence (LC) Sampling

Input: $\boldsymbol{x}_t$, q_θ, K, T_{token}
Output: $\boldsymbol{x}_{t-1}$
$\mathcal{M}_t \leftarrow \{k \in [L] \mid x_t^k = \texttt{[MASK]}\}$
$x_0^k \sim q_\theta^k(\cdot \mid \boldsymbol{x}_t;\, T_{\text{token}})$
$c_t^k \leftarrow q_\theta^k(x_0^k \mid \boldsymbol{x}_t)$
$\mathcal{U}_t \leftarrow \text{argtop-K}_{k \in \mathcal{M}_t}\ c_t^k$
$x_{t-1}^k := \begin{cases} x_0^t\,, & k \in \mathcal{U}_t \\ x_t^k\,, & \text{else} \end{cases}$

Algorithm 2 Tempered Low-Confidence (TLC) Sampling

Input: $\boldsymbol{x}_t$, q_θ, K, T_{token}, T_{pos}
Output: $\boldsymbol{x}_{t-1}$
$\mathcal{M}_t \leftarrow \{k \in [L] \mid x_t^k = \texttt{[MASK]}\}$
$x_0^k \sim q_\theta^k(\cdot \mid \boldsymbol{x}_t;\, T_{\text{token}})$
$c_t^k \leftarrow q_\theta^k(x_0^k \mid \boldsymbol{x}_t)$
$\tilde{c}_t^k \propto (c_t^k)^{1/T_{\text{pos}}} \cdot \mathbf{1}\,(k \in \mathcal{M}_t)$
$\mathcal{U}_t \sim \text{Cat}(\tilde{\boldsymbol{c}}_t)$ w/o repl., $|\mathcal{U}_t| = K$
$x_{t-1}^k := \begin{cases} x_0^t\,, & k \in \mathcal{U}_t \\ x_t^k\,, & \text{else} \end{cases}$

Algorithm 3 Confidence Thresholding (CT) Sampling ; Fast-dLLM (Wu et al., 2025)

Input: $\boldsymbol{x}_t$, q_θ, λ, T_{token}
Output: $\boldsymbol{x}_{t-1}$
$\mathcal{M}_t \leftarrow \{k \in [L] \mid x_t^k = \texttt{[MASK]}\}$
$x_0^k \sim q_\theta^k(\cdot \mid \boldsymbol{x}_t;\, T_{\text{token}})$
$c_t^k \leftarrow q_\theta^k(x_0^t \mid \boldsymbol{x}_t)$
$\mathcal{U}_t \leftarrow \{k \in \mathcal{M}_t : c_t^k > \lambda\}$
$x_{t-1}^k := \begin{cases} x_0^t\,, & k \in \mathcal{U}_t \\ x_t^k\,, & \text{else} \end{cases}$

Algorithm 4 Tempered Confidence Thresholding (TCT) Sampling

Input: $\boldsymbol{x}_t$, q_θ, λ, T_{token}, T_{pos}
Output: $\boldsymbol{x}_{t-1}$
$\mathcal{M}_t \leftarrow \{k \in [L] \mid x_t^k = \texttt{[MASK]}\}$
$x_0^k \sim q_\theta^k(\cdot \mid \boldsymbol{x}_t;\, T_{\text{token}})$
$c_t^k \leftarrow q_\theta^k(x_0^t \mid \boldsymbol{x}_t)$
$b_t^k \sim \text{Ber}\big(\sigma\big((c_t^k - \lambda)/T_{\text{pos}}\big)\big)$
$\mathcal{U}_t \leftarrow \{k \in \mathcal{M}_t : b_t^k = 1\}$
$x_{t-1}^k := \begin{cases} x_0^t\,, & k \in \mathcal{U}_t \\ x_t^k\,, & \text{else} \end{cases}$

D. Formal Statements and Proofs

In this appendix we develop the formal machinery described in Section 3.1.

Roadmap. We begin by giving an operational definition of fork tokens (Definition D.1) and state the main assumption underlying our analysis (Assumption D.2). We then establish a pairwise ordering lemma between anchors and forks (Lemma D.3) and use it to prove our main result: increasing T_{pos} increases fork-token entropy (Proposition D.4). Next, Corollary D.7 ties this to the semantic sample entropy through an additional assumption on the model quality (Assumption D.6). Finally, we show that under a mild additional assumption token temperature does not help preserve fork token entropy (Proposition D.8) while autoregressive generation provably does, compared to low-confidence remasking (Proposition D.9).

Notation.

- $\boldsymbol{x}_t$: a (possibly partially masked) string of length L at time t.
- $\mathcal{M}_t = \{k \in [L] \mid x_t^k = \texttt{[MASK]}\}$: the set of masked positions in $\boldsymbol{x}_t$.
- q_θ: the dLLM used.
- $q_\theta^k(\cdot|\boldsymbol{x}_t)$: its predicted marginal distribution over tokens at position k, when prompted with $\boldsymbol{x}_t$.
- c_t^k or $c_{t,k}$: the confidence of the model q_θ at position k in state $\boldsymbol{x}_t$ (see Section 2.2). When using the superscript for other purposes (such as taking powers), we use the double subscript notation.
- T_{pos}: position temperature, controlling the stochasticity of the remasking order for TLC.
- $P^{\pi,\theta}$ or $P_{T_{\text{pos}}}$: the joint distribution over generation trajectories induced by pairing q_θ with remasking strategy π. The shorthand $P_{T_{\text{pos}}}$ is used when π is TLC with position temperature T_{pos}.
- $T(j)$ or $T_\pi(j)$: the step at which position j is unmasked. Since this is a random quantity in $P^{\pi,\theta}$, the subscript π is included when the remasking strategy used needs to be made explicit.
- $[\![\cdot]\!] : \mathcal{V}^L \to \mathcal{S}$: a function mapping strings to semantic outcomes (e.g., final answers for mathematics problems, or program outputs with respect to a fixed test suite).

With this notation in hand, we can now give an operational definition of fork tokens.

Definition D.1 ((ϵ, δ)-fork token)**.** Let $\boldsymbol{x}_t$ be a partially masked string with masked positions $\mathcal{M}_t$. A position $\ell \in \mathcal{M}_t$ is an (ϵ, δ)-*fork token* with respect to p_{data} and $\boldsymbol{x}_t$ if:

1. $H_{p_{\text{data}}}(x_0^\ell \mid \boldsymbol{x}_t, [\![\boldsymbol{x}_0]\!]) \le \epsilon$;
2. $H_{p_{\text{data}}}([\![\boldsymbol{x}_0]\!] \mid \boldsymbol{x}_t, x_0^\ell) \le \delta$.

That is, the fork token's value and the semantic outcome are tightly coupled: knowing one constrains the other to a high degree of certainty. Repeatedly applying the chain rule, it follows immediately that

$$H_{p_{\text{data}}}(x_0^\ell \mid \boldsymbol{x}_t) - \epsilon \le H_{p_{\text{data}}}([\![\boldsymbol{x}_0]\!] \mid \boldsymbol{x}_t) \le H_{p_{\text{data}}}(x_0^\ell \mid \boldsymbol{x}_t) + \delta \, . \tag{5}$$

This aligns with the intuition in prior work (Wang et al., 2025b; Ni et al., 2026) that fork-token entropy largely dictates the semantic diversity of samples.

To relate this definition to Ni et al. (2026)'s empirical observations, we need a model of *how* the distribution $P^{\pi,\theta}$ induced by pairing q_θ with a remasking function π interacts with such fork tokens. Based on their findings, we posit a simple model in which a fork token's predictive entropy is governed by a small set of high-confidence *anchor* positions whose confidences dominate that of the fork.

Assumption D.2 (Anchor-fork degeneracy)**.** Let ℓ be an (ϵ, δ)-fork token with respect to p_{data}, $[\![\cdot]\!]$, and $\boldsymbol{x}_t$. We say that $P^{\pi,\theta}$ suffers from an *anchor-fork degeneracy* at ℓ in $\boldsymbol{x}_t$ if there exist anchor positions $\mathcal{A} \subset \mathcal{M}_t \setminus \{\ell\}$ with constants $\eta_a > 0$ such that for every state $\boldsymbol{x}_{t'}$ reachable through $P^{\pi,\theta}(\cdot \mid \boldsymbol{x}_t)$ with $\ell \in \mathcal{M}_{t'}$:

1. **There is a persistent anchor-fork confidence gap:** $c_{t'}^{a} > c_{t'}^{\ell}$ for every remaining anchor $a \in \mathcal{A} \cap \mathcal{M}_{t'}$.
2. **Revealing anchors linearly degrades the fork entropy:**

$$H_{q_\theta^\ell}(x_0^\ell \mid \boldsymbol{x}_{t'}) = H_{q_\theta^\ell}(x_0^\ell \mid \boldsymbol{x}_t) - \sum_{a \in \mathcal{A} \setminus \mathcal{M}_{t'}} \eta_a \tag{6}$$

where $H_{q_\theta^\ell}(x_0^\ell \mid \boldsymbol{x}_t) > \sum_{a \in \mathcal{A}} \eta_a > 0$.

As mentioned in Section 3.1, this model is directly inspired by (and consistent with) the empirical findings of Ni et al. (2026): (i) LC deterministically reveals all anchors before the fork, yielding minimal entropy; (ii) T_{token} will not affect the unmasking *order* unless it flips the confidences, and hence often leaves the expectation over π unchanged (Appendix D.3); and (iii) left-to-right generation yields strictly higher fork entropy than LC whenever at least one anchor follows the fork in positional order (Appendix D.4).

We now give formal evidence for each of these claims, beginning with a pairwise ordering bound that underlies the main proposition.

D.1. A pairwise anchor-fork ordering bound

Lemma D.3 (Pairwise anchor-fork ordering bound)**.** *Let ℓ be a fork token subject to an anchor-fork degeneracy (Assumption D.2), and let $a \in \mathcal{A}$ be one of its anchors. Under TLC with $K = 1$ at position temperature $T_{pos} > 0$, let $T(\ell)$ and $T(a)$ denote the steps at which ℓ and a are unmasked, respectively. Define*

$$\delta_a \triangleq \inf_{\boldsymbol{x}_{t'}} \ln \frac{c_{t',a}}{c_{t',\ell}} \qquad \text{and} \qquad \Delta_a \triangleq \sup_{\boldsymbol{x}_{t'}} \ln \frac{c_{t',a}}{c_{t',\ell}}$$

where both extrema are taken over all states $\boldsymbol{x}_{t'}$ reachable from $\boldsymbol{x}_t$ with $\{\ell, a\} \subset \mathcal{M}_{t'}$. Then

$$\sigma\left(\frac{\delta_a}{T_{pos}}\right) \le P_{T_{pos}}(T(a) < T(\ell)) \le \sigma\left(\frac{\Delta_a}{T_{pos}}\right)$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function.

Proof. Note first that $0 < \delta_a \le \Delta_a < \infty$: the lower bound follows from the confidence gap (Assumption D.2, condition 1), and the upper bound from the fact that $c_{t',a} \le 1$ and $c_{t',\ell} > 0$ in every reachable state. The statement is thus well-posed.

Now, under TLC with $K = 1$, each step selects a single masked position j with probability proportional to $c_{t,j}^{1/T_{\text{pos}}}$. Let $t^* = \min(T(a), T(\ell))$ be the first step at which either a or ℓ is unmasked. At step t^*, conditioning on the event that the selected position lies in $\{a, \ell\}$ and marginalizing out all other masked positions, we obtain

$$P_{T_{\text{pos}}}(T(a) < T(\ell)) = \frac{c_{t^*,a}^{1/T_{\text{pos}}}}{c_{t^*,a}^{1/T_{\text{pos}}} + c_{t^*,\ell}^{1/T_{\text{pos}}}} = \frac{(c_{t^*,a}/c_{t^*,\ell})^{1/T_{\text{pos}}}}{1 + (c_{t^*,a}/c_{t^*,\ell})^{1/T_{\text{pos}}}} = \sigma\left(\frac{\ln(c_{t^*,a}/c_{t^*,\ell})}{T_{\text{pos}}}\right).$$

Since $\delta_a \le \ln(c_{t^*,a}/c_{t^*,\ell}) \le \Delta_a$ and σ is strictly increasing, the result follows. □

D.2. Increasing T_{pos} increases fork-token entropy

Proposition D.4 (Increasing T_{pos} increases fork-token entropy)**.** *Let $T_{pos}, T'_{pos} > 0$ be two position temperatures for TLC at $K = 1$. Let ℓ be a fork token in state $\boldsymbol{x}_t$ subject to an anchor-fork degeneracy (Assumption D.2) under both $P_{T_{pos}}$ and $P_{T'_{pos}}$. Let δ_a, Δ_a be as in Lemma D.3, and define*

$$\delta \triangleq \min_{a \in \mathcal{A}} \delta_a, \qquad \Delta \triangleq \max_{a \in \mathcal{A}} \Delta_a.$$

Then if $T'_{pos} > T_{pos} \cdot \Delta/\delta$,

$$H_{P_{T_{pos}}}(x_0^\ell \mid \boldsymbol{x}_t) < H_{P_{T'_{pos}}}(x_0^\ell \mid \boldsymbol{x}_t).$$

Proof. By the linear entropy reduction condition (Assumption D.2, condition 2) as well as standard results about $\mathbb{E}$ (the tower rule and the linearity of expectation), we have for $\tau \in \{T_{\text{pos}}, T'_{pos}\}$ that

$$\begin{aligned}
H_{P^{\tau,\theta}}(x_0^\ell \mid \boldsymbol{x}_t) &= \mathbb{E}_{\boldsymbol{x}_{T_\tau(\ell)-1} \sim P^{\tau,\theta}} \left[H_{q_\theta^\ell}(x_0^\ell \mid \boldsymbol{x}_{T_\tau(\ell)-1}) \right] && \text{(tower rule)} \\
&= \mathbb{E}_{\boldsymbol{x}_{T_\tau(\ell)-1} \sim P^{\tau,\theta}} \left[H_{q_\theta^\ell}(x_0^\ell \mid \boldsymbol{x}_t) - \sum_{a \in \mathcal{A} \setminus \mathcal{M}_{T_\tau(\ell)-1}} \eta_a \right] && \text{(Assumption D.2)} \\
&= H_{q_\theta^\ell}(x_0^\ell \mid \boldsymbol{x}_t) - \mathbb{E}_{\boldsymbol{x}_{T_\tau(\ell)-1} \sim P^{\tau,\theta}} \left[\sum_{a \in \mathcal{A} \setminus \mathcal{M}_{T_\tau(\ell)-1}} \eta_a \right] \\
&= H_{q_\theta^\ell}(x_0^\ell \mid \boldsymbol{x}_t) - \sum_{a \in \mathcal{A}} P_\tau(T(a) < T(\ell)) \cdot \eta_a
\end{aligned}$$

where $T_\tau(\ell)$ denotes the unmasking time of the fork. Since each $\eta > 0$, it thus suffices to show that $P_{T'_{\text{pos}}}(T(a) < T(\ell)) < P_{T_{\text{pos}}}(T(a) < T(\ell))$ for every $a \in \mathcal{A}$. Per Lemma D.3 and the definitions of δ, Δ:

$$P_{T'_{\text{pos}}}(T(a) < T(\ell)) \le \sigma\left(\frac{\Delta}{T'_{\text{pos}}}\right) < \sigma\left(\frac{\delta}{T_{\text{pos}}}\right) \le P_{T_{\text{pos}}}(T(a) < T(\ell)),$$

where the strict inequality follows from the gap condition $T'_{\text{pos}} > T_{\text{pos}} \cdot \Delta/\delta$, which gives $\Delta/T'_{\text{pos}} < \delta/T_{\text{pos}}$, combined with σ being strictly increasing. The result follows. □

Remark D.5. Lemma D.3 only applies when $T_{\text{pos}}, T'_{\text{pos}} > 0$, so this proof does not allow us to compare TLC (at some $T_{\text{pos}} > 0$) vs LC. However, note that this case is trivial: Under LC, all anchors are revealed before the fork with probability 1 by the confidence gap (Assumption D.2, condition 1), so

$$H_{P^{0,\theta}}(x_0^\ell \mid \boldsymbol{x}_t) = H_{q_\theta^\ell}(x_0^\ell \mid \boldsymbol{x}_t) - \sum_{a \in \mathcal{A}} \eta_a,$$

the minimal fork entropy achievable under the anchor-fork degeneracy model. Since any $T_{\text{pos}} > 0$ yields $P_{T_{\text{pos}}}(T(a) < T(\ell)) < 1$ for every anchor a (that is, $\sigma(x) < 1$ for all finite x), it follows that $H_{P^{T_{\text{pos}},\theta}}(x_0^\ell \mid \boldsymbol{x}_t) > H_{P^{\text{LC},\theta}}(x_0^\ell \mid \boldsymbol{x}_t)$ for any $T_{\text{pos}} > 0$. That is, any amount of tempering strictly improves fork-token entropy over deterministic LC.

In order to tie this proposition back to Definition D.1 and conclude that the semantic entropy of the final generation will increase (Corollary D.7), we need one more assumption: that q_θ captures the data distribution well enough that no matter what π is (within the set of strategies under consideration), it will not stop ℓ from also being a fork token with respect to $P^{\pi,\theta}$.

Assumption D.6 (Model quality)**.** Let ℓ be an (ϵ, δ)-fork token with respect to p_{data} and $\boldsymbol{x}_t$. We say that q_θ *preserves the fork structure* at ℓ if ℓ is also an (ϵ, δ)-fork token with respect to $P^{\pi,\theta}$ for all remasking strategies π.

That is, if the data distribution has a fork at position ℓ, then q_θ's sampling distribution preserves that property regardless of the remasking strategy, so that the sandwich bound (5) also holds under $P^{\pi,\theta}$.

We are now ready to finally make a statement about how $H_{P^{\pi,\theta}}([\![\boldsymbol{x}_0]\!] \mid \boldsymbol{x}_t)$ varies with T_{pos}: that is, how changing the position temperature affects the semantic entropy of our samples.

Corollary D.7 (Semantic entropy increases with T_{pos})**.** *Under the conditions of Proposition D.4, suppose further that q_θ preserves the fork structure at ℓ (Assumption D.6). If*

$$H_{P_{T'_{pos}}}(x_0^\ell \mid \boldsymbol{x}_t) - H_{PT_{pos}}(x_0^\ell \mid \boldsymbol{x}_t) > \epsilon + \delta,$$

then

$$H_{P_{T'_{pos}}}([\![\boldsymbol{x}_0]\!] \mid \boldsymbol{x}_t) > H_{P_{T_{pos}}}([\![\boldsymbol{x}_0]\!] \mid \boldsymbol{x}_t).$$

Proof. Applying the sandwich bound (5) under $P^{T'_{\text{pos}},\theta}$ and $P^{T_{\text{pos}},\theta}$ respectively (which is valid by Assumption D.6):

$$\begin{aligned} H_{P_{T'_{\text{pos}}}}([\![\boldsymbol{x}_0]\!] \mid \boldsymbol{x}_t) &\geq H_{P_{T'_{\text{pos}}}}(x_0^\ell \mid \boldsymbol{x}_t) - \epsilon \\ &> H_{P_{T_{\text{pos}}}}(x_0^\ell \mid \boldsymbol{x}_t) + \delta \\ &\geq H_{P_{T_{\text{pos}}}}([\![\boldsymbol{x}_0]\!] \mid \boldsymbol{x}_t) \end{aligned}$$

where the strict inequality uses the assumed gap. □

D.3. Token temperature does not affect fork-token entropy under deterministic remasking

As shown in Appendix C, the confidences c_t^k do not depend on T_{token} as typically implemented[4] in confidence-based heuristics. One may speculate that allowing T_{token} to affect the confidences and then increasing it would yield similar diversity gains to adjusting T_{pos}. We here show that, under a strengthened version of the confidence-gap from Assumption D.2, this is provably not the case: the unmasking order under deterministic LC is invariant to T_{token} regardless, so adjusting T_{token} does not help with preserving entropy at forks.

Proposition D.8 (Token temperature invariance under deterministic LC)**.** *Let ℓ be a fork token subject to an anchor-fork degeneracy (Assumption D.2), and suppose sampling is performed using deterministic low-confidence remasking at $K = 1$, with confidences taken post-tempering with T_{token}. Suppose further that for all reachable states $\boldsymbol{x}_{t'}$ with $\ell \in \mathcal{M}_{t'}$, every remaining anchor $a \in \mathcal{A} \cap \mathcal{M}_{t'}$ satisfies a strengthened gap condition:*

$$z^a_{(1)} - z^a_{(i)} \geq z^\ell_{(1)} - z^\ell_{(i)} \qquad \textit{for all } i \geq 2, \textit{with at least one inequality strict}$$

where $z^j_{(1)} \geq z^j_{(2)} \geq \cdots$ denotes the logit vector at position j sorted in descending order. Then all anchors are revealed before ℓ regardless of T_{token}.

Proof. Under this version of deterministic LC, the unmasking order is determined by $\arg\max_{j \in \mathcal{M}_t} c_t^j(T_{\text{token}})$, where $c^j(T) = \max_v[\text{softmax}(\boldsymbol{z}^j/T)]_v$. It suffices to show that $c^a(T) > c^\ell(T)$ for all $T > 0$.

Let $V = |\mathcal{V}|$ and let $\boldsymbol{z}$ be any logit vector with sorted entries $z_{(1)} \geq z_{(2)} \geq \cdots \geq z_{(V)}$. Then

$$c(T) = \frac{e^{z_{(1)}/T}}{\sum_{i=1}^V e^{z_{(i)}/T}} = \frac{1}{1 + \sum_{i=2}^V e^{(z_{(i)} - z_{(1)})/T}}.$$

By the assumption that $z^a_{(1)} - z^a_{(i)} \geq z^\ell_{(1)} - z^\ell_{(i)}$ for all $i \geq 2$, we have $z^a_{(i)} - z^a_{(1)} \leq z^\ell_{(i)} - z^\ell_{(1)}$ and hence $e^{(z^a_{(i)} - z^a_{(1)})/T} \leq e^{(z^\ell_{(i)} - z^\ell_{(1)})/T}$ for each i and all $T > 0$. Summing over $i \geq 2$ and using the assumption that at least one inequality is strict (which we note follows anyway from the confidence-gap condition of Assumption D.2), we have that:

$$c^a(T) = \frac{1}{1 + \sum_{i=2}^V e^{(z^a_{(i)} - z^a_{(1)})/T}} > \frac{1}{1 + \sum_{i=2}^V e^{(z^\ell_{(i)} - z^\ell_{(1)})/T}} = c^\ell(T).$$

The unmasking order is thus preserved for all $T > 0$, so all anchors are revealed before ℓ regardless of T_{token}. □

D.4. Autoregressive generation increases fork-token entropy

Finally, we observe that autoregressive (left-to-right) generation yields higher fork-token entropy than deterministic low-confidence remasking, since it does not systematically defer the fork.

Proposition D.9 (Autoregressive generation improves fork-token entropy)**.** *Let ℓ be a fork token subject to an anchor-fork degeneracy (Assumption D.2) with respect to both LC and AR, and suppose at least one anchor appears after ℓ in left-to-right order. Then*

$$H_{P^{\text{LC},\theta}}(x_0^\ell \mid \boldsymbol{x}_t) < H_{P^{\text{AR},\theta}}(x_0^\ell \mid \boldsymbol{x}_t).$$

[4] See, e.g., the source code for LLaDA's low-confidence remasking or Fast-dLLM's reference implementation for confidence-thresholding.

Proof. Under both strategies the remasking order is deterministic, so $R_\pi = \{a \in \mathcal{A} : T(a) < T(\ell)\}$ is a fixed set for each. Under LC, the confidence gap ensures $R_{\mathrm{LC}} = \mathcal{A}$. Under AR, $R_{\mathrm{AR}} = \{a \in \mathcal{A} : a < \ell\} \subsetneq \mathcal{A}$ by assumption. By the tower rule (as in the proof of Proposition D.4),

$$H_{P^{\pi,\theta}}(x_0^\ell \mid \boldsymbol{x}_t) = \mathbb{E}_{\boldsymbol{x}_{T_\pi(\ell)-1}}\left[H_{q_\theta^\ell}(x_0^\ell \mid \boldsymbol{x}_{T_\pi(\ell)-1})\right] = H_{q_\theta^\ell}(x_0^\ell \mid \boldsymbol{x}_t) - \sum_{a \in R_\pi} \eta_a \,,$$

where the second equality applies condition 2 of Assumption D.2, noting that $\mathcal{A} \setminus \mathcal{M}_{T_\pi(\ell)-1} = R_\pi$. Since $R_{\mathrm{AR}} \subsetneq R_{\mathrm{LC}} = \mathcal{A}$ and each $\eta_a > 0$, we thus conclude that

$$H_{P^{\mathrm{AR},\theta}}(x_0^\ell \mid \boldsymbol{x}_t) > H_{P^{\mathrm{LC},\theta}}(x_0^\ell \mid \boldsymbol{x}_t).$$

□

E. Additional Figures

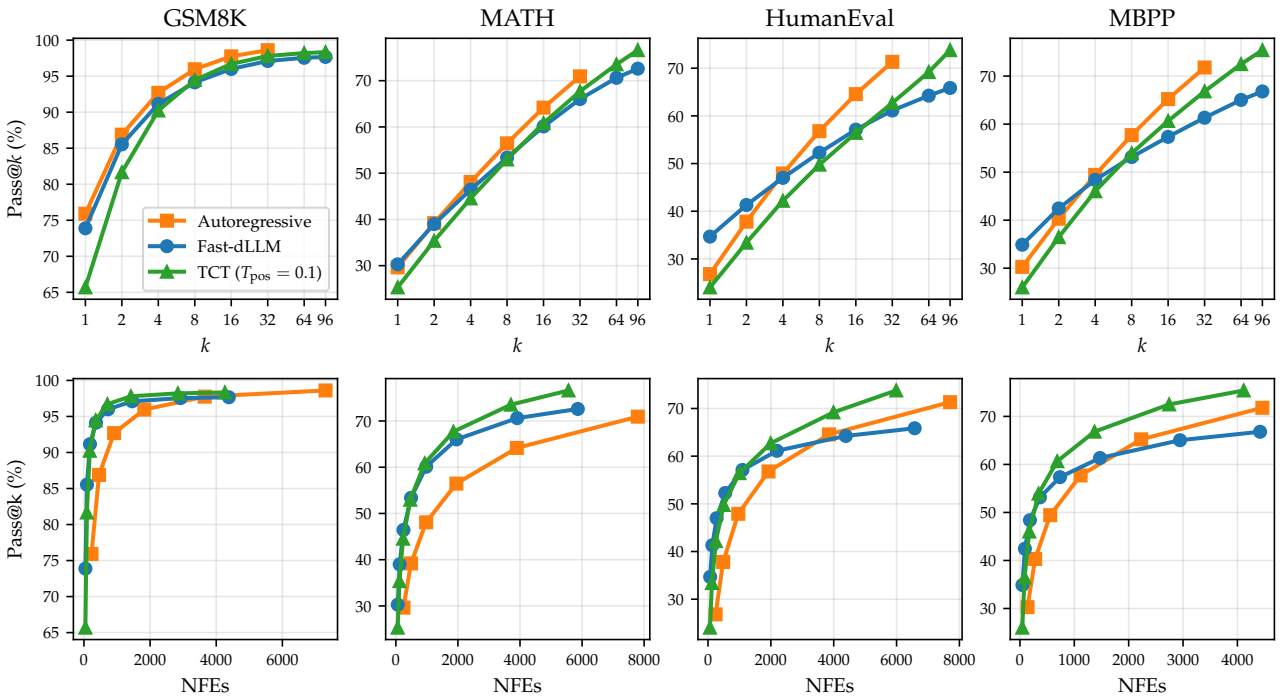


Figure 3. Pass@k (top) and pass@NFE (bottom) for TCT on LLaDA-8B-Instruct. Both rows plot the same data; only the metric changes (pass@k vs pass@NFE). TCT slightly underperforms AR in pass@k, but substantially outperforms it when the cost of each sample is taken into account. TCT also outperforms deterministic confidence-thresholding (Fast-dLLM; Wu et al. 2025) thanks to its additional source of randomness via T_{pos}.

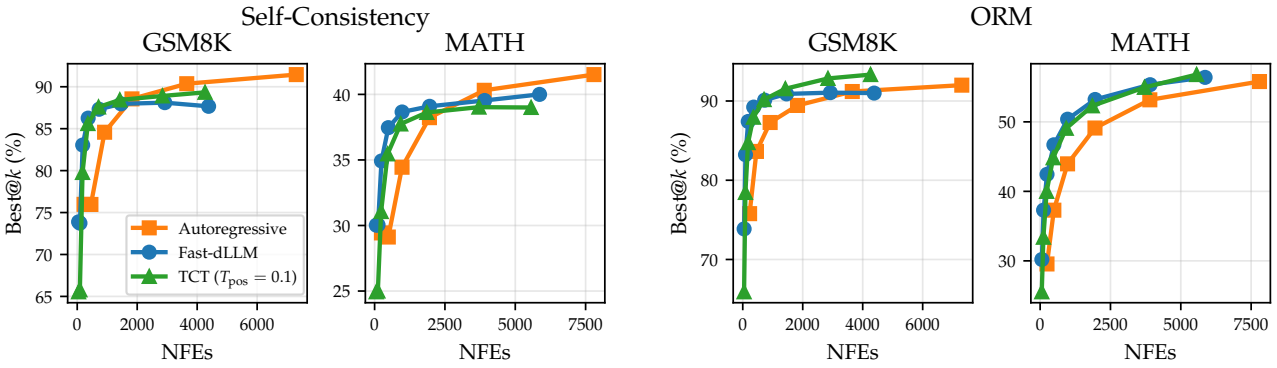


Figure 4. Test-time scaling for LLaDA-8B-Instruct in terms of Best@NFE when using either self-consistency or an ORM for final answer selection. TCT performs worse compared to AR at higher NFEs when using self-consistency; however, when using an ORM, TCT improves the pareto frontier compared to autoregressive rollouts. See Figure 9 for Best@k results.

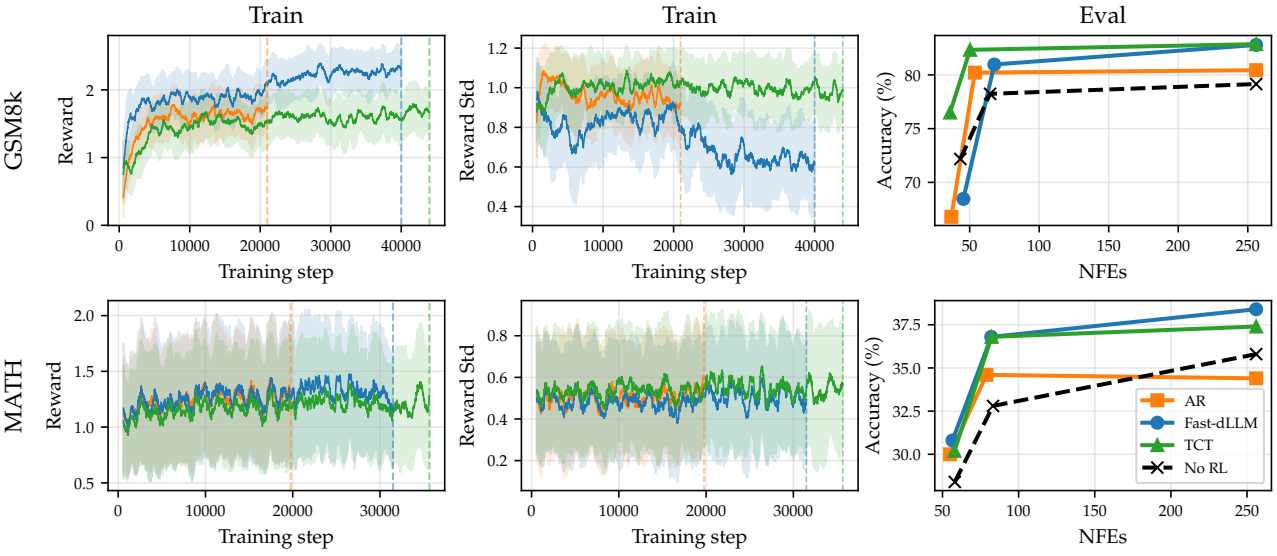


Figure 5. GRPO post-training results for LLaDA-8B-Instruct on GSM8k (top) and MATH (bottom). *Left*: Group mean training reward. *Middle*: Group standard deviation of the reward. *Right*: evaluation accuracy under greedy decoding with confidence thresholding at varying λ. All models are trained for a fixed wall-clock budget of 72 hours; vertical dashed lines indicate the number of completed training steps for each sampler. TCT completes around twice as many training steps compared to AR due to its adaptive parallelism, maintains the highest reward diversity throughout training due to $T_{pos} > 0$, and yields evaluation performance comparable to or better than both AR and Fast-dLLM baselines.

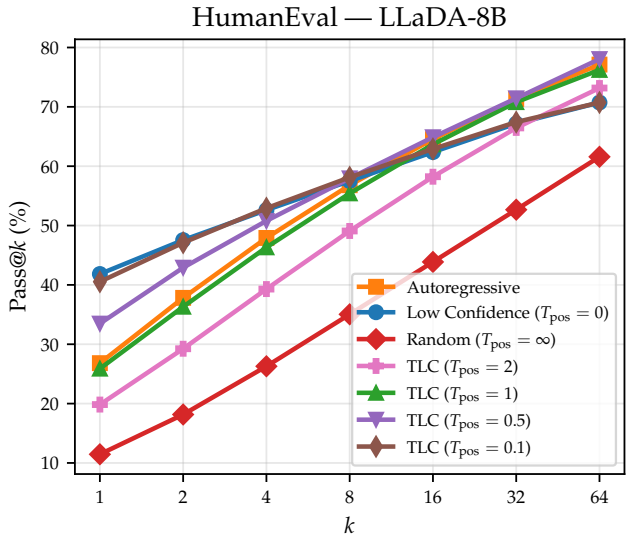


Figure 6. TLC T_{pos} ablation on HumanEval with LLaDA-8B-Instruct with $T_{\text{token}} = 0.8$. Too little tempering ($T_{\text{pos}} = 0.1$) remains close to low-confidence remasking, while too much ($T_{\text{pos}} = 2$) approaches random remasking and degrades quality. Intermediate values ($T_{\text{pos}} \in \{0.5, 1\}$) strike the best balance, matching or exceeding the autoregressive baseline at high k.

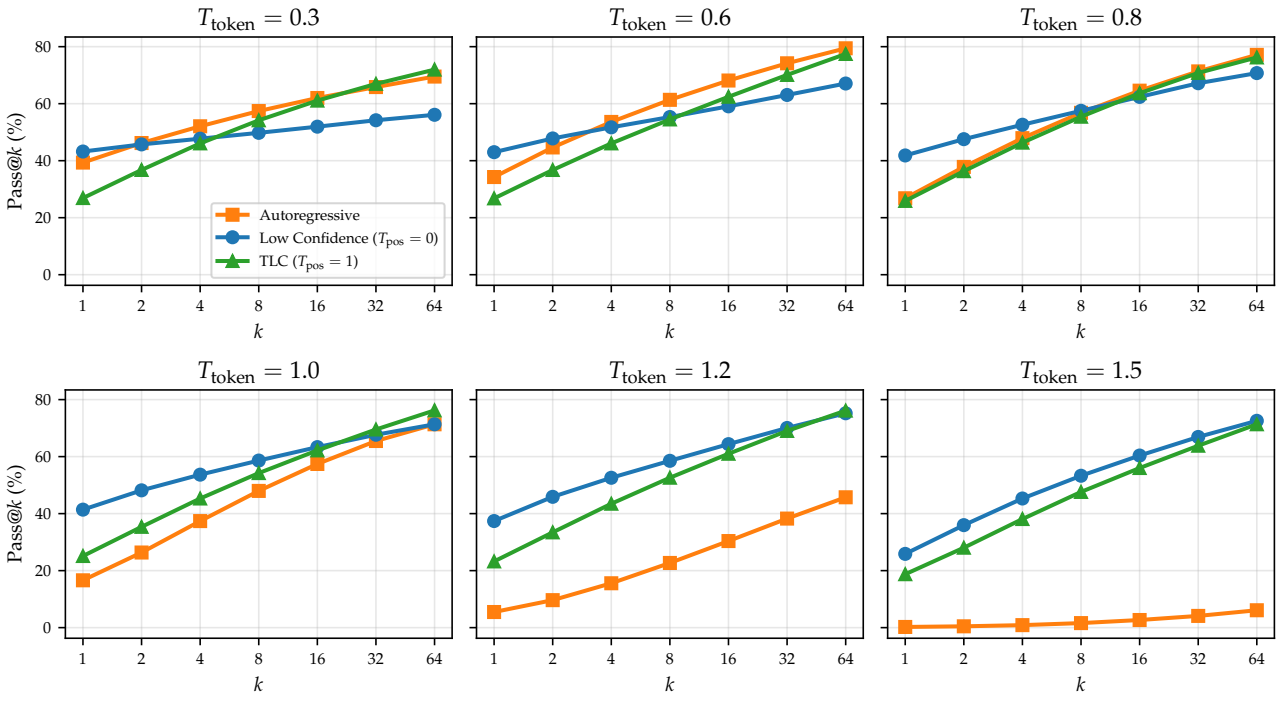


Figure 7. Varying T_{token} for LLaDa-8B-Instruct on HumanEval.Autoregressive generation favors low temperatures, obtaining its best pass@64 at $T_{\text{token}} = 0.6$. Meanwhile, low-confidence remasking scales very poorly at low token temperatures due to its greedy position selection. TLC consistently shows strong scaling and most robustness to the exact choice of T_{token}, matching or exceeding the best of autoregressive and low-confidence remasking in pass@64 at all temperatures. Moreover, it exhibits stronger scaling compared to low-confidence across all considered T_{token}, suggesting it achieves diversity through an alternative mechanism.

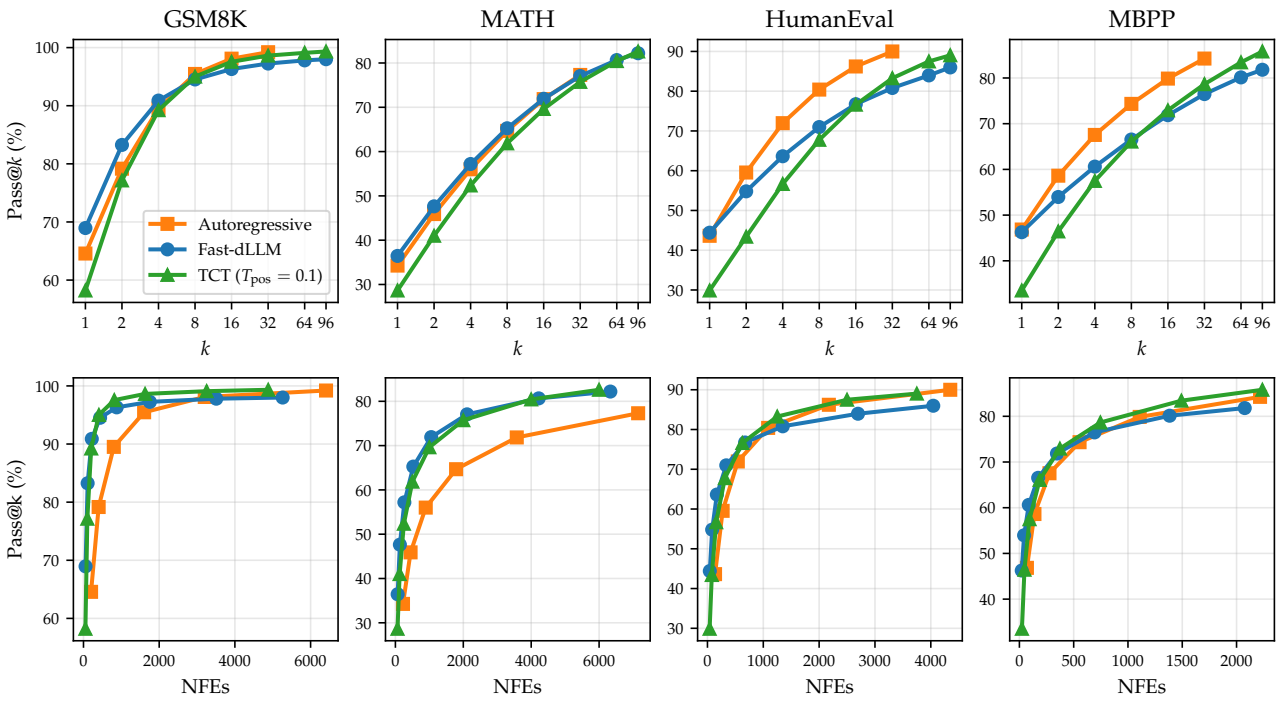


Figure 8. Pass@k (top) and pass@NFE (bottom) for TCT on Dream-7B-Instruct. Note that both rows plot the same data, we just vary the x-axis measure (k vs NFEs). While TCT slightly underperforms AR in pass@k, it outperforms AR when the cost of each rollout is taken into account.

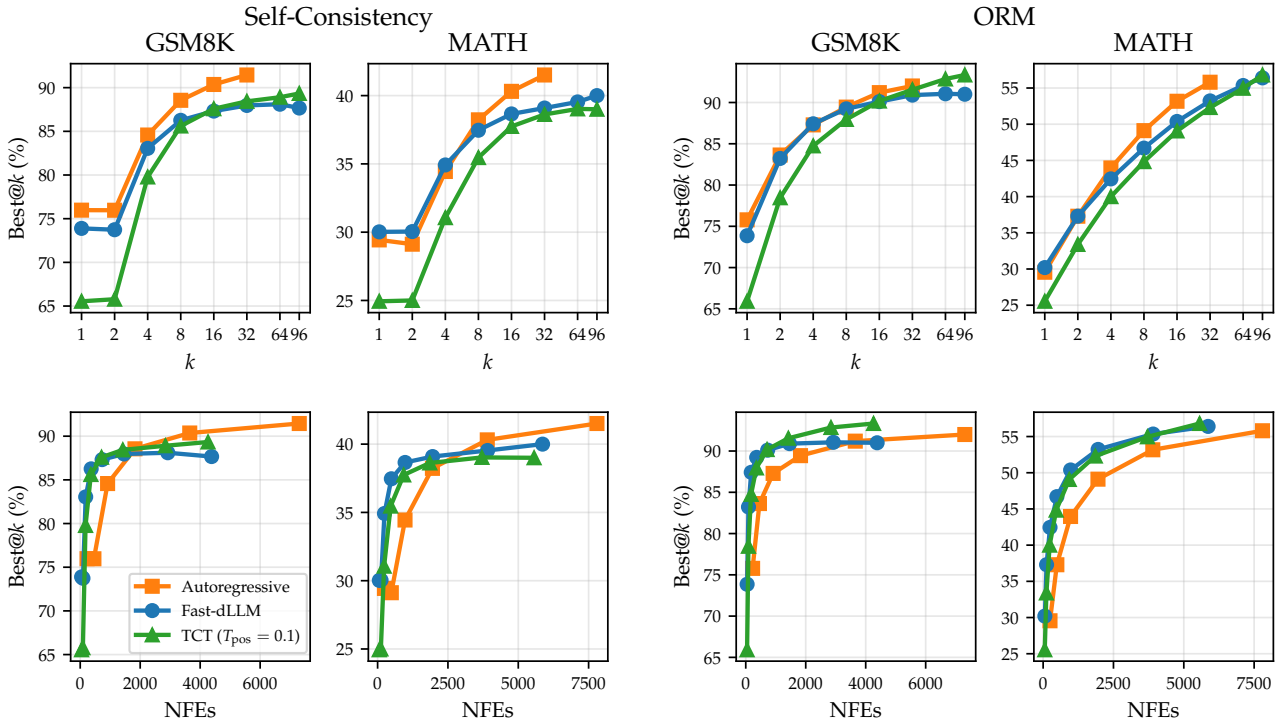


Figure 9. Test-time scaling for LLaDA-8B-Instruct in terms of Best@k (top) and Best@NFE (bottom) when using either self-consistency or an ORM for final answer selection. TCT performs worse compared to AR at higher NFEs when using self-consistency; however, when using an ORM, TCT improves the pareto frontier compared to autoregressive rollouts.

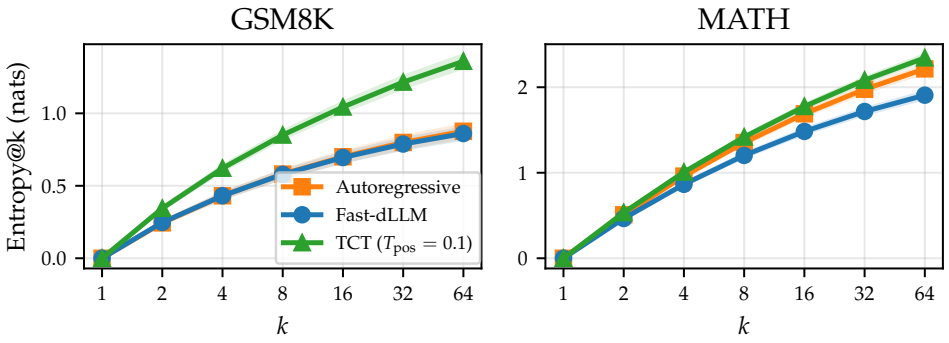


Figure 10. Mean empirical entropy of the per-question answer distributions (see Figure 11) as a function of the group size k. TCT consistently yields higher-entropy answer distributions than autoregressive and confidence thresholding, particularly on GSM8k.

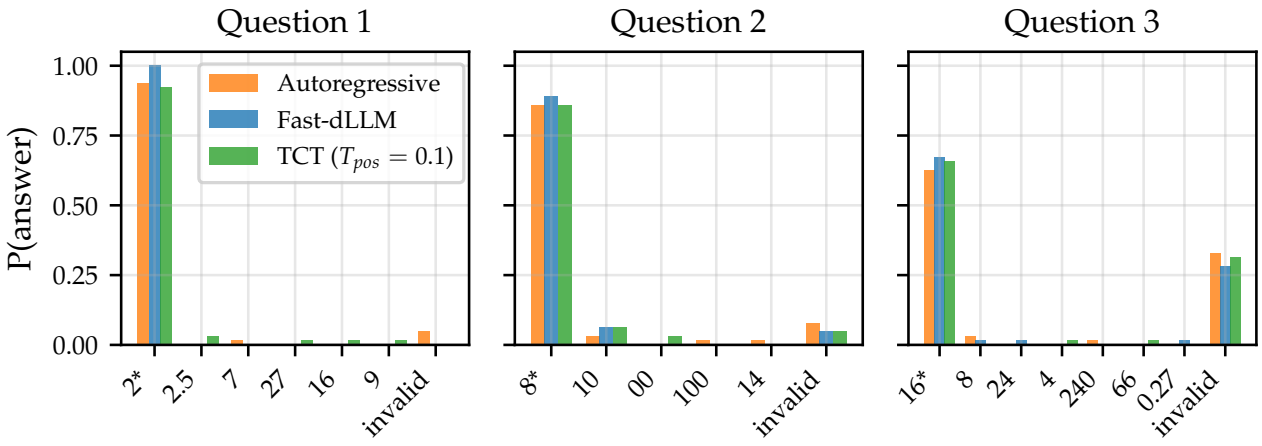


Figure 11. Per-question answer distributions over $k = 64$ samples for three GSM8K questions, comparing autoregressive, Fast-dLLM, and TCT sampling. Correct answers are marked with an asterisk (*). We observe that TCT sometimes spreads mass across a wider set of candidates (left). This illustrates the higher-entropy behavior observed in Figure 10: TCT produces more diverse answers, which benefits pass@k and ORM-based selection but can hurt majority voting when the correct answer no longer dominates the distribution.