WHY DIFFUSION LANGUAGE MODELS STRUGGLE WITH TRULY PARALLEL (NON-AUTOREGRESSIVE) DECODING?

Anonymous authors
Paper under double-blind review

ABSTRACT

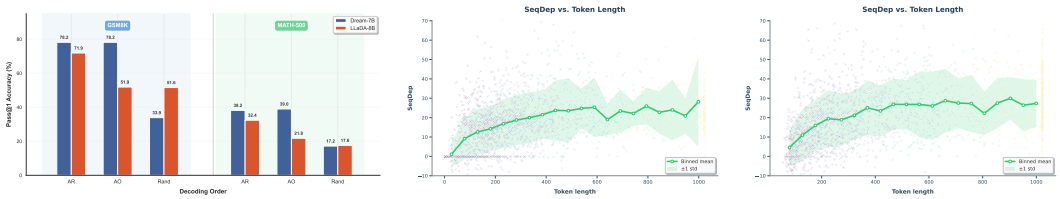
Diffusion Language Models (DLMs) are often advertised as enabling parallel token generation, yet practical "fast" DLMs frequently converge to left-to-right, autoregressive (AR)-like decoding dynamics. In contrast, genuinely non-AR generation is promising because it removes AR's sequential bottleneck, better exploiting parallel hardware to reduce synchronization/communication overhead and improve latency scaling with output length. We argue that a primary driver of AR-like decoding is a mismatch between DLM objectives and the highly sequential structure of widely used training data, including standard pretraining corpora and long chain-of-thought (CoT) supervision. Motivated by this diagnosis, we propose **NAP** (Non-Autoregressive Parallel DLMs), a proof-of-concept, data-centric approach that better aligns supervision with non-AR parallel decoding. NAP curates examples as multiple independent reasoning trajectories and couples them with a parallel-forced decoding strategy that encourages multi-token parallel updates. Across math reasoning benchmarks, NAP yields stronger performance under parallel decoding than DLMs trained on standard long CoT data, with gains growing as parallelism increases. Our results suggest that revisiting data and supervision is a principled direction for mitigating AR-like behavior and moving toward genuinely non-autoregressive parallel generation in DLMs.

1 INTRODUCTION

Large language models (LLMs) have become a cornerstone of modern AI, yet their rapidly growing computational and environmental footprints raise pressing sustainability concerns (Patterson et al., 2021; Luccioni et al., 2023). This motivates renewed interest in alternative generation paradigms that can reduce inference latency and cost without sacrificing capability. *Diffusion Language Models* (DLMs) have recently emerged as a compelling candidate: by iteratively denoising a sequence, DLMs can in principle enable *parallel token generation*, offering a path toward faster, more efficient generation Austin et al. (2021b); Lou et al. (2023); Shi et al. (2024b); Sahoo et al. (2024a); Nie et al. (2025b); Gong et al. (2024); Ye et al. (2025). When paired with established inference accelerators, such as *KV caching* (Ma et al., 2025; Wu et al., 2025; Liu et al., 2025) and *speculative decoding* (Christopher et al., 2025; Gao et al., 2025), DLM-based systems are often claimed as substantially faster alternatives to standard autoregressive (AR) decoding.

Yet, despite their promise, practical "fast" DLMs exhibit a striking and under-discussed behavior: many methods that aim for highly parallel decoding *converge toward AR-like generation*, where the effective reasoning trajectory proceeds largely *from left to right* (Nie et al., 2025b; Israel et al., 2025; Wu et al., 2025; Gong et al., 2025). In other words, even when the model architecture permits bidirectional context and parallel refinement, the realized decoding dynamics can resemble a sequential construction of the output. This phenomenon makes real-world DLM usage more nuanced than the headline promise of "truly parallel decoding": speedups are often coupled to subtle quality trade-offs, and the conditions under which DLMs depart meaningfully from AR behavior remain unclear Kang et al. (2025).

The payoff for achieving genuinely (non-AR) parallel decoding is substantial: AR-style decoding is fundamentally sequential, every token depends on the previous one, so generation latency scales roughly with output length. Although we can switch to fast parallel decoding in subsequent blocks after earlier blocks have largely converged, the need to wait for upstream stabilization introduces



(a) Accuracy vs. Decoding Order. (b) OpenR1-Math SeqDep (c) Fineweb SeqDep

Figure 1: **(a)** Performance on GSM8K (left) and MATH-500 (right). Forcing low-ARness behavior (Random) generally causes reasoning performance to collapse (e.g., Dream-7B). Notably, for LLaDA, we employ a *block-wise* decoding strategy to ensure generation validity; this constraint preserves local structure, resulting in Random and AO decoding exhibiting comparable performance, unlike the sharp drop in fully unstructured decoding. **(b, c)** Sequential Dependence (SeqDep) analysis on OpenR1-Math and FineWeb datasets. The consistently high and rising SeqDep scores indicate that standard training corpora possess strong intrinsic sequentiality, driving models to internalize AR-like dependencies.

a sequential critical path, leading to extra latency and communication cost Wang et al. (2025); Fu et al. (2025). In contrast, truly non-AR parallel decoding is naturally compatible with the distributed hardware, i.e., when dependencies across spans are weak, decoding is naturally compatible with parallel hardware and can be distributed across devices, with only occasional synchronization to maintain global consistency.

In this work, we argue that one primary caveat of this AR-bias is a ***mismatch between the learning objective and the training data***. Existing DLM pipelines blindly reuse training data originally designed for AR models, where reasoning trajectories are implicitly encoded as left-to-right progressions (e.g., next-token prediction–style ordering Ye et al. (2025); Allal et al. (2025); Li et al. (2024), or sequential Chain-of-Thought (CoT) rationales (Zhao et al., 2025; Lambert et al., 2024)). As a result, even if the diffusion process is nominally position-agnostic, the model can learn denoising strategies that preferentially reconstruct outputs in an AR-shaped manner. This "AR-shaped data" effect not only limits the extent to which DLMs can exploit genuine parallelism, but also complicates evaluation: a method may appear effective while largely reproducing AR model's dynamics under a different wrapper.

To test this conjecture, we conduct a systematic analysis of the decoding behavior of commonly used DLMs. The main findings are summarized below.

I. Widely used training corpora are strongly sequential. We quantify the sequential dependency of datasets by measuring how strongly the token at one position is determined by its preceding context. We show a consistent trend: commonly used pre-training corpora (i.e., FineWeb Penedo et al. (2024)) and long CoT reasoning datasets (i.e., Open-R1-Math Team (2025)) display strong sequence dependence.

II. DLMs decoding remains largely autoregressive. Across widely used DLM families such as LLaDA Nie et al. (2025c) and Dream Ye et al. (2025), ARness remains high: the model still tends to "lock in" decisions in a quasi-left-to-right pattern, despite the nominally parallel update rules. Conversely, forcing genuinely low ARness behavior, for instance, by randomizing the update order aggressively, can reduce ARness but typically causes reasoning performance to collapse. Taken together, these results indicate a frustrating tradeoff: in standard setups, either ARness stays high, or lowering ARness breaks reasoning.

III. Training on long CoT data escalates ARness. While DLMs trained from scratch (e.g., LLaDA) tend to exhibit lower ARness than those adapted from pretrained AR models (e.g., Dream), continued post-training on standard long CoT datasets further increases ARness over time. Intuitively, long CoT supervision provides an explicit step-by-step trajectory with a privileged order, so matching the training target rewards the model for producing and stabilizing earlier tokens before later ones, pushing the learned decoding dynamics toward an increasingly AR-like proceed.

IV. Recent parallel fast-DLM methods gain speed by *amplifying*, not removing, AR-like generation. Despite being motivated by parallel decoding, many recent fast-DLM approaches achieve

practical speedups by reinforcing an underlying autoregressive computation pattern. In particular, they rely on increasingly confident early predictions or staged block-wise updates that stabilize prefixes before allowing limited parallelism downstream. As a result, parallelism is effectively gated by an AR-like convergence order, and the achieved acceleration stems from exaggerating this sequential structure rather than eliminating it.

The above findings suggest that even though the DLMs permit arbitrary decoding strategy, as DLMs are trained on highly sequentially structured data, the model tends to internalize an AR-like computational strategy. In other words, the training distribution teaches the model that reasoning is a *chain* with a privileged order, and changing the decoding procedure alone is often insufficient to undo this learned reliance. Addressing the issue therefore requires revisiting the data and supervision that shape the model's generation strategy in the first place.

To this end, we propose **NAP** (*Non-Autoregressive Parallel DLMs*), a **proof-of-concept** approach that tackles the problem from a data and decoding codesign perspective. First, we curate supervision in which each example consists of multiple *independent reasoning trajectories* generated in parallel, this format deemphasizes any privileged token order and is naturally compatible with denoising-style learning in DLMs. Second, we introduce a *parallel-forced* decoding strategy that explicitly encourages multi-token parallel updates at different reasoning traces, further steering generation away from AR-like critical paths. Together, these two components provide a simple and effective way to better align DLM behavior with truly parallel decoding. Across a range of math reasoning benchmarks, our results show that NAP, fine-tuned with 103K samples, consistently yields stronger performance under parallel decoding than the baseline trained on standard long CoT datasets. Moreover, the improvement becomes more pronounced as we increase the degree of parallelism, indicating that NAP is better aligned with non-AR decoding dynamics rather than relying on an implicit sequential critical path.

Note that our goal is **not** to claim that NAP fully resolves the challenges of non-AR parallel decoding. Rather, we aim to use this small-scale post-training only result to show that revisiting data and supervision design is a principled direction for mitigating AR-like behavior in DLMs and moving toward genuinely non-autoregressive parallel generation. We hope our results motivate further work on data-centric approaches to unlock the full efficiency potential of DLMs.

2 Preliminaries

2.1 Diffusion Language Models

We consider diffusion language models (DLMs), and in particular masked diffusion models (MDMs), which generate discrete token sequences by iteratively denoising a partially masked state. Let x denote the input prompt and let $y_0 = (y_0^1, \ldots, y_0^L) \in \mathcal{V}^L$ denote a clean output sequence of length L over vocabulary $\mathcal{V}$. MDMs define a forward masking process indexed by a continuous time variable $t \in [0, 1]$, where t represents the masking ratio. Given y_0, the forward process independently masks each token with probability t:

$$q(y_t^k \mid y_0^k) = \begin{cases} \texttt{[MASK]}, & \text{with prob. } t, \\ y_0^k, & \text{with prob. } 1 - t, \end{cases} \tag{1}$$

and factorizes across positions as $q(y_t \mid y_0) = \prod_{k=1}^{L} q(y_t^k \mid y_0^k)$. At $t = 1$, the sequence is fully masked; at $t = 0$, it remains unchanged.

2.2 Measuring Autoregressive Bias

To quantify how autoregressive-like a DLM decoding trajectory is, we adopt the **ARness** metrics proposed by Gong et al. (2025), which distinguish between global left-to-right bias and local sequential continuity. Let the decoding process be represented by a sequence of unmasked positions $\mathbf{p} = (p_1, p_2, \ldots, p_L)$, where $p_t \in \{1, \ldots, L\}$ denotes the position index of the token committed at step t. Let M_{t-1} be the set of masked positions just before step t.

Global ARness. This metric measures the tendency to prioritize unmasking the leftmost remaining tokens, capturing a front-to-back filling strategy. For a tolerance window $k \geq 1$, we define an

indicator $\mathbb{I}_{\text{global}}(t, k)$ that is 1 if the chosen position p_t is among the k earliest positions in M_{t-1}:

$$\mathbb{I}_{\text{global}}(t, k) = \begin{cases} 1, & \text{if } p_t \in \text{smallest-}k(M_{t-1}), \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The Global ARness score is the average over the sequence:

$$\text{Global-ARness@}k = \frac{1}{L} \sum_{t=1}^{L} \mathbb{I}_{\text{global}}(t, k) \in [0, 1]. \tag{3}$$

A score of 1.0 (at $k = 1$) indicates a strict autoregressive (left-to-right) generation order.

Unless otherwise stated, we use **Global-ARness@1** as the primary measure of ARness in our analysis, as it directly quantifies the adherence to a causal generation order.

2.3 Measuring Sequential Dependence (SeqDep)

To quantify the intrinsic sequentiality of data, we use a prefix-gain score computed by an external autoregressive scorer p_{AR}. For a sequence split into segments $(s_1, \ldots, s_K)$:

$$\text{SeqDep}(x, s_{1:K}) = \frac{1}{K-1} \sum_{i=2}^{K} \left(\log p_{\text{AR}}(s_i \mid x, s_{<i}) - \log p_{\text{AR}}(s_i \mid x) \right) \tag{4}$$

Higher SeqDep indicates that later tokens are strongly determined by preceding context, implying a chain-like structure.

3 Decoding Behaviors of DLMs

In this section, we conduct a systematic analysis of the decoding behavior of commonly used DLMs. We fix the pretrained masked diffusion model, the token budget, the number of refinement steps, and the mask-ratio schedule, and vary only the decoding rule. This isolates the effect of the induced generation order from all other factors. Our main findings are summarized below.

3.1 Strong Sequential Dependence in Training Corpora

A primary driver of sequential behavior is the data itself. We hypothesize that if the training distribution is highly sequential, the model learns an implicit left-to-right dependency that persists even under parallel decoding objectives.

We quantify this using the SeqDep metric (Sec. 2.3) on two representative datasets: FineWeb (pre-training corpora) and OpenR1-Math (long-CoT reasoning). As shown in Figure 1b and 1c, both datasets display strong sequence dependence. Notably, reasoning steps in OpenR1-Math exhibit increasing dependence as the chain progresses (p_{AR} predicts later steps with much higher confidence given the prefix). This suggests that standard training data teaches the model that reasoning is a fundamentally ordered chain, creating a mismatch with position-agnostic diffusion objectives.

3.2 DLMs' Decoding Remains Largely Autoregressive

Given sequential training data, we examine how DLMs behave during inference. We evaluate two popular models, LLaDA-8B Nie et al. (2025c) and Dream-7B Ye et al. (2025), under three decoding strategies: AR order (left-to-right), Arbitrary Order (AO) (confidence-based), and Random.

High ARness in DLM Decoding. Table 1 reports the ARness scores. While AR order is 1.0 by definition, AO decoding converges to extremely high ARness (~ 0.92 for Dream), indicating that the model's most "confident" tokens are almost always the next tokens in the sequence. As a result, DLMs exhibit behavior closely resembling autoregressive generation.

The Accuracy–ARness Tradeoff. Is it possible to force genuinely parallel behavior? We test this using a Random decoding strategy, which successfully yields near-zero ARness. However, as shown in Figure 1, this comes at a severe cost: reasoning accuracy on GSM8K Cobbe et al. (2021) and MATH 500 Lightman et al. (2023) collapses when the model is prevented from following a sequential path. These results suggest that strong reasoning performance is often obtained at the cost of genuine parallelism, as improved accuracy tends to coincide with higher *AR-ness* under standard setups.

Table 1: **Quantifying Autoregressive Bias (Global ARness@1).** Comparison of sequential bias across different decoding strategies. While AR order is strictly sequential (1.00), confidence-based arbitrary order (AO) remains highly autoregressive compared to the random baseline.

Model	AR Order	AO (Conf)	Rand
LLaDA-8B Nie et al. (2025c)	1.00	0.73	0.01
Dream-7B Ye et al. (2025)	1.00	0.92	0.01
Fast-dLLM (LLaDA) Wu et al. (2025)	1.00	0.87	-
Fast-dLLM (Dream) Wu et al. (2025)	1.00	0.94	-

3.3 Long-CoT Supervision Escalates AR-ness

Table 2: **Long-CoT Supervision Increases ARness.** Comparison of Global ARness@1 scores (using AO decoding) before and after fine-tuning.

Model	Base (Pretrained)	Long-CoT (SFT)	Change
LLaDA-8B	0.73	0.81	↑ 0.08
Dream-7B	0.92	0.93	↑ 0.01

We further investigate how supervised fine-tuning (SFT) on long Chain-of-Thought (CoT) data affects decoding dynamics. We compare the ARness of base models against checkpoints post-trained on standard CoT datasets (Open-R1 Math (Team, 2025)).

Results indicate a clear trend: **post-training further increases ARness**. For instance, LLaDA's base ARness under AO decoding rises from 0.73 to 0.81 after CoT tuning. Intuitively, CoT supervision provides explicit step-by-step trajectories with a privileged order. Minimizing the loss on such data rewards the model for stabilizing earlier tokens before later ones, effectively "baking in" the AR order and making it harder for the model to utilize genuine parallel decoding during inference.

3.4 Current Fast DLMs Reinforce Sequentiality

Finally, we analyze whether specialized "fast" decoding algorithms can unlock genuine parallelism. We evaluate **Fast-dLLM** Wu et al. (2025), a state-of-the-art acceleration method that employs block-wise parallel decoding.

As shown in Table 1, these methods do not reduce sequential dependence; in fact, they exacerbate it. For instance, while standard AO decoding for LLaDA has an ARness of 0.73, applying Fast-dLLM pushes this score up to **0.87**. Similarly, for Dream-7B, the ARness rises to **0.94**, nearly indistinguishable from strict autoregressive decoding (1.00).

This empirical evidence suggests that current "fast" DLMs achieve speedups not by enabling non-sequential generation, but by effectively identifying and accelerating the underlying autoregressive critical path. The parallelism in these systems is gated by the convergence of the prefix, meaning they optimize the execution of the sequential chain rather than eliminating the bottleneck. This diagnosis reinforces our core premise: achieving *true* non-autoregressive parallelism requires revisiting the supervision signal itself, rather than relying solely on inference-time algorithmic optimizations.

4 NAP: Non-Autoregressive Parallel DLMs

4.1 Overview

To bridge the gap between DLM objectives and the sequential nature of reasoning data, we propose **NAP** (*Non-Autoregressive Parallel DLMs*). NAP is a data-decoding co-design framework that breaks the implicit autoregressive lock-in by restructuring both the supervision signal and the inference process. The framework operates on two levels: first, it curates training examples as multiple independent reasoning trajectories rather than a single linear chain, thereby removing the notion of

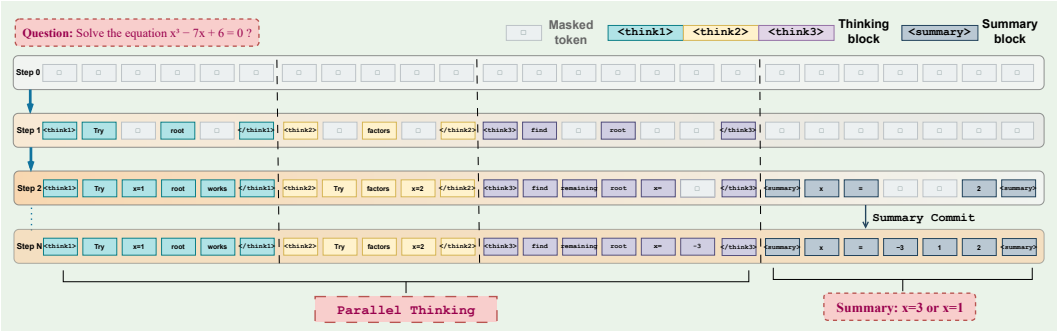


Figure 2: **Overview of the parallel-forced decoding framework**. The model concurrently generates multiple independent reasoning paths within structured thinking blocks. These parallel trajectories are then synthesized into a result within a designated summary block.

a privileged order; second, it employs a **parallel-forced decoding** strategy that explicitly enforces multi-stream updates during inference, preventing the model from collapsing into a sequential critical path.

4.2 Data Curation

Standard chain-of-thought (CoT) data typically encodes a single canonical left-to-right reasoning order, creating a natural mismatch with the objective of parallel DLM decoding. To address this, we curate a dataset $\mathcal{D}_{\text{parallel}}$ whose supervision is inherently parallel.

Generating Parallel Reasoning Traces. Similar to ParaThinker Wen et al. (2025), given a query x, we prompt a strong teacher model to generate P independent reasoning traces $\{r^{(1)}, \ldots, r^{(P)}\}$. We employ a high sampling temperature ($\tau = 1.0$) to induce diverse problem-solving approaches or distinct logical orderings. Unlike standard augmentation which treats these as separate samples, NAP groups them into a single training instance. This ensures that the parallel paths represent truly independent explorations rather than redundant copies.

Summary and Aggregation. To teach the model to resolve conflicts, we construct the summary block S by conditioning the ground-truth answer a on the concatenation of these diverse (and potentially noisy) paths. The final training instance follows the format in Eq. equation 5, as illustrated in Figure 4. In this setup, the model observes multiple parallel paths—some of which may contain errors (e.g., Path 3)—followed invariably by the correct result in S. This supervision forces the model to implicitly learn how to identify valid reasoning streams and filter out noise to match the ground truth, treating the parallel paths as supporting evidence rather than a linear chain. We fine-tune the DLM on this structured data using the standard masked diffusion objective.

4.3 Parallel-Forced Decoding

To enable the model to reason in parallel, we design a decoding canvas that spatially separates reasoning streams and enforce a structure-aware update schedule.

Decoding Canvas. We define a structured output format containing m independent reasoning blocks and one summary block:

$$Y \;=\; \big[\, B_1, R^{(1)},\; B_2, R^{(2)},\; \ldots,\; B_m, R^{(m)},\; B_{\mathrm{S}}, S \,\big], \tag{5}$$

where B_j are fixed textual headers (e.g., "`<think #j>`"), $R^{(j)}$ are free-form reasoning contents for the j-th path, and S is a final summary containing the answer. Given a prompt x, we initialize a canvas of length $L = \sum(|B_j| + L_j) + (|B_{\mathrm{S}}| + L_{\mathrm{S}})$, where fixed headers are clamped and reasoning slots are initialized to `[MASK]`. This layout effectively enforces *conditional independence* between $R^{(i)}$ and $R^{(j)}$ given the prompt, as there is no causal masking order between them in a bidirectional model.

Table 3: Benchmark results on LLaDA-8B-Instruct and Dream-7B-Instruct under different step budgets. Tok/Step denotes the number of tokens decoded per decoding step; larger Tok/Step corresponds to higher decoding parallelism.

Benchmark	Steps	Tok/Step	LLaDA 8B	LLaDA 8B (Long-CoT)	NAP-LLaDA 8B	Dream-7B	Dream-7B (Long-CoT)	NAP-Dream-7B
Mathematics & Scientific								
GSM8K	256	4	46.4	54.1	**56.1** (+2.0)	35.0	46.5	**60.9** (+14.4)
	336	3	54.4	60.9	**63.3** (+2.4)	49.4	56.9	**70.9** (+14.0)
	512	2	62.0	82.0	**82.6** (+0.4)	58.5	66.8	**79.2** (+12.4)
	1024	1	66.5	83.5	**84.1** (+0.6)	68.9	78.0	**83.6** (+5.6)
MATH-500	256	4	17.8	21.4	**26.6** (+5.2)	8.8	16.2	**23.8** (+7.6)
	336	3	20.6	26.6	**35.4** (+8.8)	11.4	25.6	**31.4** (+5.8)
	512	2	28.0	41.2	**43.0** (+1.8)	20.8	40.0	**43.0** (+3.0)
	1024	1	30.4	45.0	**47.0** (+2.0)	35.0	47.4	**49.6** (+2.2)
GPQA	256	4	5.8	**9.8**	**9.8** (+0.0)	1.3	4.2	**5.8** (+1.6)
	336	3	12.5	15.4	**19.0** (+3.6)	5.8	7.3	**10.5** (+3.2)
	512	2	18.8	21.2	**25.9** (+4.7)	14.7	19.4	**22.5** (+3.1)
	1024	1	20.8	23.0	**28.6** (+5.6)	26.1	28.6	**29.5** (+0.9)

Macro-Parallel, Micro-Confidence Updates. Standard arbitrary-order decoding often degenerates into global sequential generation because the model preferentially resolves the immediate next tokens. NAP-D prevents this via a hierarchical schedule. At the macro level, we enforce strict parallelism: the unmasking budget is distributed across *all* m reasoning blocks $\{R^{(1)}, \ldots, R^{(m)}\}$ at every step. This constraint prevents the model from stabilizing upstream paths before initiating downstream ones. At the micro level, within each individual block $R^{(j)}$, we apply a confidence-based strategy (i.e., masking low-confidence tokens). We do not enforce a left-to-right order locally; instead, tokens are committed based on their confidence scores. This combination ensures that the global process is parallel (evolving multiple trajectories simultaneously) while local generation retains the flexibility of non-autoregressive refinement.

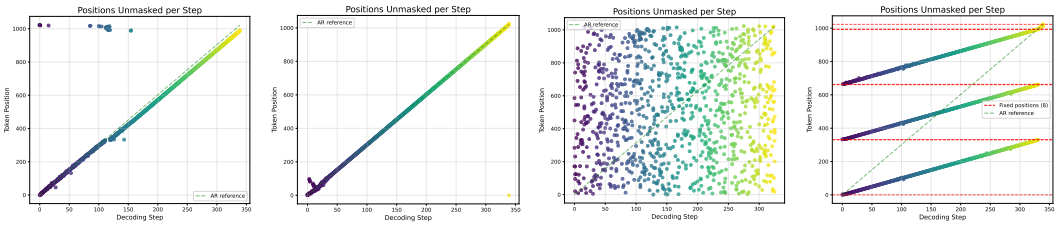
5 EXPERIMENTS

This section evaluates whether our decoding strategy can (i) improve reasoning performance over standard diffusion decoding rules, (ii) reshape the induced generation order as measured by ARness (Section 2.2), and (iii) mitigate order sensitivity in regimes where long-form rationales exhibit strong sequential dependence (Eq. equation 4). Unless otherwise stated, *all* results use the same pretrained masked diffusion model and differ *only* in the decoding rule.

Evaluation protocol. We evaluate on a suite of reasoning benchmarks including GSM8K Cobbe et al. (2021), MATH-500 Lightman et al. (2023), and GPQA Rein et al. (2024). Each example is prompted to produce a thinking path and a final answer in a fixed format; we extract answers with a deterministic parser and report accuracy.

Models and Training. We conduct experiments on two state-of-the-art diffusion language models: `LLaDA-8B-Instruct` Nie et al. (2025c) and `Dream-7B-Instruct` Ye et al. (2025). To validate our proposed method, we fine-tune these base models on the parallel reasoning dataset $\mathcal{D}_{\text{parallel}}$ curated via the pipeline described in Section 4.2. For a fair comparison, we also train a Long-CoT baseline on the same set of reasoning trajectories but serialized in the standard autoregressive format. Crucially, this baseline is evaluated using standard decoding—its optimal inference setting—rather than our parallel strategy, ensuring a strong and fair comparison. Both variants are trained using the standard masked diffusion objective for 3 epochs. We use the AdamW optimizer with a learning rate of 2e-6 and a global batch size of 256. All experiments are conducted on 8 NVIDIA A800 GPUs.

Decoding baselines. We compare several widely used unmasking rules under the common mask-and-predict framework. **AR order** commits the leftmost unresolved tokens at each step (a diffusion realization of left-to-right decoding). **Arbitrary order (AO)** commits the most confident positions. **Random order (Rand)** commits a uniformly random subset at each step, serving as a low-ARness control. Our method generates m multiple independent reasoning paths and a final summary commit on a structured canvas. To ensure a fair budget, PaS-Dec uses the same total token cap L by allocating per-path budgets 330 and a summary budget 32 such that the overall canvas length matches the baseline. The summary block is the only region used for answer extraction and scoring.



(a) LLaDA-8B (AO) (b) Dream-7B (AO) (c) Random (d) Ours

Figure 3: Visualization of decoding dynamics. We plot the token position being unmasked (y-axis) against the decoding step (x-axis). **(a, b)** Despite using confidence-based Arbitrary Order (AO) decoding, standard DLMs (LLaDA and Dream) exhibit a strict linear diagonal pattern, revealing that their behavior collapses into autoregressive (left-to-right) generation. **(c)** Random decoding eliminates AR bias but lacks structure. **(d)** Our method (NAP) breaks the single-stream bottleneck, generating multiple reasoning trajectories simultaneously as evidenced by the distinct parallel diagonal bands.

5.1 Main Results

Table 3 summarizes the performance across three benchmarks. Across all benchmarks and step budgets, our method achieves higher accuracy than both the Base model and the Long-CoT baseline. For instance, on GSM8K with Dream-7B (1024 steps), NAP-Dream-7B reaches 83.6%, surpassing the Long-CoT model (78.0%) despite using the same amount of compute and training data. This suggests that organizing reasoning into parallel streams is a more effective supervision signal for DLMs than forcing a single long chain.

The most significant advantage of NAP appears in the low-step regime, e.g., 256 steps (4x parallel), where the model must generate more than one token per forward pass. Standard `Long-CoT` models degrade sharply as parallelism increases. On Dream-7B/GSM8K, accuracy drops from 78.0% (1024 steps) to 46.5% (256 steps). This confirms that standard supervision creates a dependency on sequential stability; when forced to hurry, the reasoning collapses. In the same setting, NAP-Dream-7B maintains strong accuracy at 60.9%, compared to 46.5% of the Long-CoT baseline, thereby retaining substantially more capability. Notably, the gap between NAP and Long-CoT widens as parallel decoding is made more aggressive, increasing from +5.6% at 1024 steps to **+14.4%** at 256 steps. This result validates our core hypothesis: by training on data that lacks a privileged order, the model learns to be less reliant on the immediate left-side context, enabling effective Non-AR parallel decoding.

To further understand how NAP achieves these results, we analyze the relationship between performance and the sequential nature of generation (ARness). As shown in Figure 3, standard models (LLaDA/Dream) using Arbitrary Order (AO) decoding exhibit a strict diagonal pattern. Even though they can decode anywhere, they effectively collapse into a left-to-right process (High ARness). In contrast, NAP (Figure 3(d)) displays distinct parallel bands, confirming that multiple reasoning trajectories are being generated simultaneously.

6 Conclusion

In this work, we argue that the struggle of Diffusion Language Models (DLMs) to achieve genuine parallel decoding stems largely from the implicit sequentiality of standard training data. Our proposed method, NAP, demonstrates that aligning supervision with parallel decoding dynamics effectively mitigates this autoregressive collapse. By training on parallel reasoning trajectories and enforcing multi-stream updates, NAP decouples reasoning capability from sequential order, achieving superior performance in high-parallelism regimes while significantly reducing global ARness. These results suggest that unlocking the full potential of non-autoregressive generation requires moving beyond decoding heuristics to fundamentally rethink how we structure supervision for parallel reasoning.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 17981–17993, 2021a.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021b.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

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 (Volume 1: Long Papers)*, pp. 12042–12059, 2025.

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

Hengyu Fu, Baihe Huang, Virginia Adams, Charles Wang, Venkat Srinivasan, and Jiantao Jiao. From bits to rounds: Parallel decoding with exploration for diffusion language models. *arXiv preprint arXiv:2511.21103*, 2025.

Yifeng Gao, Ziang Ji, Yuxuan Wang, Biqing Qi, Hanlin Xu, and Linfeng Zhang. Self speculative decoding for diffusion large language models. *arXiv preprint arXiv:2510.04147*, 2025.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*, 2024.

Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation. *arXiv preprint arXiv:2506.20639*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Daniel Israel, Guy Van den Broeck, and Aditya Grover. Accelerating diffusion llms via adaptive parallel decoding. *CoRR*, abs/2506.00413, 2025.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjae Lee, Yuchen Zeng, Shuibai Zhang, Coleman Hooper, Yuezhou Hu, Hyung Il Koo, Nam Ik Cho, et al. Parallelbench: Understanding the trade-offs of parallel decoding in diffusion llms. *arXiv preprint arXiv:2510.04767*, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Yi Liang, Soroush Vosoughi, and Shiwei Liu. Diffusion language models know the answer before decoding. *arXiv preprint arXiv:2508.19982*, 2025.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

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. *arXiv preprint arXiv:2506.06295*, 2025.

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

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. Estimating the carbon footprint of bloom, a 176b parameter language model. *Journal of machine learning research*, 24(253):1–15, 2023.

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.

Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, et al. The flexibility trap: Rethinking the value of arbitrary order in diffusion language models. 2026.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16784–16804. PMLR, 2022.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *CoRR*, abs/2502.09992, 2025a.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025b.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025c. doi: 10.48550/arXiv.2502.09992. URL https://arxiv.org/abs/2502.09992.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Avishek Joey Bose, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685. IEEE, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024a.

Subham S. Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a.

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

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *CoRR*, abs/1503.03585, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

OpenR1 Team. Openr1-math-220k: A large-scale math reasoning dataset. `https://huggingface.co/datasets/open-r1/OpenR1-Math-220k`, 2025. Accessed 2025.

Xu Wang, Chenkai Xu, Yijie Jin, Jiachun Jin, Hao Zhang, and Zhijie Deng. Diffusion llms can do faster-than-ar inference via discrete diffusion forcing. *arXiv preprint arXiv:2508.09192*, 2025.

Hao Wen, Yifan Su, Feifei Zhang, Yunxin Liu, Yunhao Liu, Ya-Qin Zhang, and Yuanchun Li. Parathinker: Native parallel thinking as a new paradigm to scale llm test-time compute. *arXiv preprint arXiv:2509.04475*, 2025.

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. *arXiv preprint arXiv:2505.22618*, 2025.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *CoRR*, abs/2505.15809, 2025.

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

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

Yichen Zhu, Weiyu Chen, James Kwok, and Zhou Zhao. Spmdm: Enhancing masked diffusion models through simplifing sampling path. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A Appendix

B Related Work

B.1 Diffusion Language Models

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021), best known for their success in image generation (Rombach et al., 2022; Nichol et al., 2022; Saharia et al., 2022), are increasingly studied as a non-autoregressive alternative for text generation. Bringing diffusion from continuous variables to discrete tokens can be formalized by treating the forward corruption as a Markov process over a finite vocabulary: D3PM (Austin et al., 2021a) instantiates this idea with discrete-time transition matrices, while subsequent work extends it to continuous time through CTMC formulations (Campbell et al., 2022). A particularly practical family is masked diffusion, which can be viewed as an absorbing-state construction in the D3PM lineage and operates directly in token space via random masking (Shi et al., 2024a). This paradigm has produced strong results across scales, from smaller models such as MDLM (Sahoo et al., 2024b) and RADD (Ou et al., 2025) to large systems like LLaDA (Nie et al., 2025a) and Dream (Ye et al., 2025). Beyond text-only settings, MMaDA (Yang et al., 2025) further generalizes large diffusion models to multimodal generation with a shared probabilistic view and modality-agnostic architecture, while the broader literature highlights potential benefits such as parallelizable decoding and flexible (non left-to-right) generation orders that may be useful for complex reasoning.

B.2 Decoding Order and Sampling Schedules

A key degree of freedom in masked diffusion language models is the sampling path—which positions are updated (or committed) at each refinement step and in what order. Rather than being a mere implementation detail, several works treat order as an explicit control knob for quality/efficiency trade-offs. P2 (Peng et al., 2025) cast order selection as a planning problem, where a separate planner chooses which tokens to denoise at each step, decoupling where/when to update from how to update. Prophet (Li et al., 2025) further leverages model confidence to early-commit, switching from iterative refinement to one-shot completion when the top-2 gap indicates convergence. Order-awareness has also been pushed into training, e.g., by encouraging simpler and more coherent sampling paths (Zhu et al.). Meanwhile, Ni et al. (2026) caution that arbitrary-order flexibility can be a double-edged sword: models may preferentially resolve low-uncertainty tokens and bypass high-uncertainty branching points, collapsing the effective reasoning space, suggesting that constraining or regularizing generation order can sometimes improve reasoning.

B.3 Ablation Studies

We investigate the individual contributions of the supervision data and the decoding strategy using Dream-7B on the GSM8K benchmark.

The Necessity of Data-Decoding Co-design. We first isolate the impact of our proposed decoding method versus the parallel-aligned data. As shown in Table 4, applying our Parallel-Forced Decoding strategy to a standard base model that has not been trained with our data leads to a larger performance

[Input Query]
A shirt costs \$50 and is on sale for 20% off. What is the final price?
[Model Output]
⟨think 1⟩ Method: Calculate discount amount first. Discount = $50 \times 0.20 = 10$. Final Price = $50 - 10 = 40$. ⟨/think 1⟩
⟨think 2⟩ Direct multiplier. Since it's 20% off, we pay $100\% - 20\% = 80\%$. Final Price = $50 \times 0.8 = 40$. ⟨/think 2⟩
⟨think 3⟩ 20% of 50 is 10. Final Price = $50 - 10 = 30$. **[Calculation Error]** ⟨/think 3⟩
⟨summary⟩ By analyzing multiple reasoning processes above, I concluded that: The final answer is $\boxed{40}$. ⟨/summary⟩

Figure 4: A compact training instance. The model generates parallel paths (including distinct methods and a noisy path) and aggregates them into a correct summary.

drop than standard Arbitrary Order (AO) decoding. This suggests that without training support, the original Dream-7B struggles to handle the fragmented context of simultaneous generation. In addition, the decoding strategy becomes critical when parallelism is high. Specifically at the aggressive 256-step budget, our Parallel-Forced decoding outperforms AO (60.9% vs. 57.4%). This confirms that while the data provides the foundational reasoning capability, aligning the decoding strategy is essential to maintain robustness when forcing the model to generate multiple tokens in parallel.

Table 4: GSM8K accuracy using Dream-7B. Simply applying parallel decoding to a base model hurts performance; gains require aligned supervision.

Training Data	Decoding	256	512	1024
Base (Pretrained)	AO	35.0	58.5	68.9
Base (Pretrained)	**Parallel-Forced**	31.0	52.6	60.2
NAP (Ours)	AO	57.4	78.9	**85.1**
NAP (Ours)	**Parallel-Forced**	**60.9**	**79.2**	83.6

Impact of Parallel Width (m). We further analyze how the number of parallel reasoning paths affects performance while keeping the total token budget constant. As detailed in Table 5, increasing the number of reasoning paths from a single chain ($m = 1$) to three ($m = 3$) provides consistent accuracy gains across both model families. Specifically, NAP-Dream sees a substantial improvement from 75.4% to 83.6%, while NAP-LLaDA rises from 79.4% to 84.1%. This monotonic trend supports the view that NAP benefits from an "internal ensemble" effect, where the final summary block effectively aggregates insights from multiple diverse trajectories generated in parallel to derive a more robust answer.

Table 5: Accuracy on GSM8K with varying m. Total token budget is fixed.

Method	1 Path	2 Paths	3 Paths
NAP-Dream	75.4	78.9	83.6
NAP-LLaDA	79.4	82.6	84.1

Intrinsic Parallelism of Curated Data. To verify that our data curation pipeline effectively reduces the autoregressive bottleneck, we analyze the Sequential Dependence (SeqDep) of our constructed dataset $\mathcal{D}_{\text{parallel}}$. As illustrated in Figure 5, the SeqDep score remains remarkably stable (mean ≈ 12) even as the sequence length grows from 500 to over 1000 tokens. Unlike standard long-chain reasoning (as shown in Section 3), where dependence often escalates with depth, our parallel-structured data maintains a consistent level of information density. This "flat" dependency profile confirms that the reasoning trajectories within our data possess high conditional independence, providing the necessary learning signal for the model to perform effective parallel updates during inference.

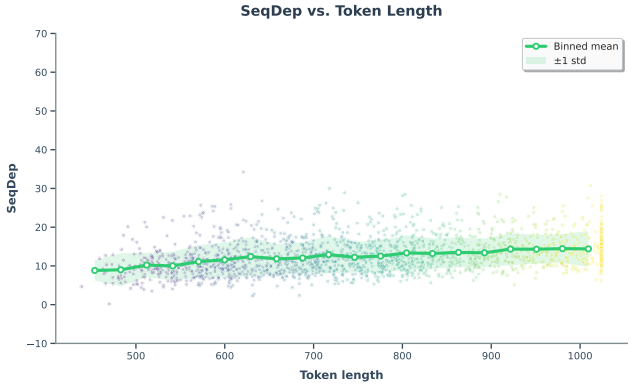


Figure 5: **SeqDep Analysis on $\mathcal{D}_{\text{parallel}}$.** We visualize the Sequential Dependence (SeqDep) of our curated parallel reasoning data against token length. The green curve (binned mean) shows that SeqDep remains stable and relatively low across varying lengths.