Analyzing and Bridging the Granularity Gap: Block-Size Curriculum Learning for Diffusion Reasoning Models

Anonymous ACL submission

Abstract

Block diffusion language models accelerate decoding through parallel block-wise denoising, yet whether they can be reliably scaled for long chain-of-thought (CoT) reasoning remains unresolved. To this end, we develop DIFFUREASONER-8B-Thinking, an open-source block diffusion reasoning model, and conduct a systematic study of how training and inference block sizes affect long-CoT reasoning. Our analysis reveals a stark performance disparity: training with large block sizes yields remarkably poor reasoning, whereas small block sizes preserve effective reasoning. To bridge this granularity gap, we propose block-size curriculum learning, which gradually transitions training from fine-grained to coarse-grained block sizes, thereby overcoming this limitation and enabling strong reasoning performance that generalizes across diverse inference block sizes. On mathematical and code reasoning benchmarks, DIFFUREASONER-8B-Thinking achieves results competitive with leading open autoregressive models such as Qwen3-8B. This work establishes a practical foundation for efficient, reasoning-capable diffusion language models. We will release our model and checkpoints.

1 Introduction

Autoregressive (AR) language models have become the dominant paradigm for complex reasoning tasks, owing to their strong sequential coherence and the demonstrated ability to generate extended chains of thought (Google DeepMind, 2026; OpenAI, 2026; DeepSeek-AI, 2026). Yet the strict left-to-right factorization in AR decoding fundamentally limits parallelism during inference. This efficiency bottleneck has motivated increasing interest in alternative architectures that can preserve reasoning quality while enabling greater generation parallelism.

Diffusion language models have emerged as a promising alternative, offering parallel token generation, bidirectional context modeling, and flexible decoding strategies (Sahoo et al., 2024; Gong et al., 2025a; Nie et al., 2025b; Ye et al., 2025). Among these, block diffusion (Arriola et al., 2025; Bie et al., 2025, 2026) provides a principled structural solution that interpolates between pure diffusion and pure autoregressive generation. By partitioning the output sequence into contiguous blocks, block diffusion handles inter-block dependencies autoregressively and intra-block tokens in parallel via bidirectional diffusion denoising, thereby enabling a flexible trade-off between sequential coherence and decoding efficiency. Nevertheless, open-source reasoning-capable diffusion models still lag behind AR counterparts on standard benchmarks (Wang et al., 2025; Cheng et al., 2025), and these models typically exhibit substantial performance degradation as block size increases as shown in Figure 1, undermining the efficiency gains promised by coarse-grained block decoding.

In this work, we study whether larger blocks can be made useful for reasoning in diffusion language models. Specifically, we investigate whether block diffusion models can be trained to reason effectively with large block sizes, and identify the training and decoding strategies that enable this. To answer this question, we train DIFFUREASONER-8B-Thinking, an open 8B-parameter block-diffusion reasoning model, and systematically evaluate how training block size, inference block size, and decoding strategy interact. Rather than treating block size as a fixed architectural constant, we analyze it as a *scaling axis* that jointly affects reasoning quality and decoding efficiency.

Our analysis reveals several key findings: First, naively training with a large block size leads to degradation in reasoning performance while small block sizes preserve reasoning capabilities. Second, block-size curriculum learning gradually transitions training from fine-grained to coarse-grained block sizes, overcoming the degradation associ-

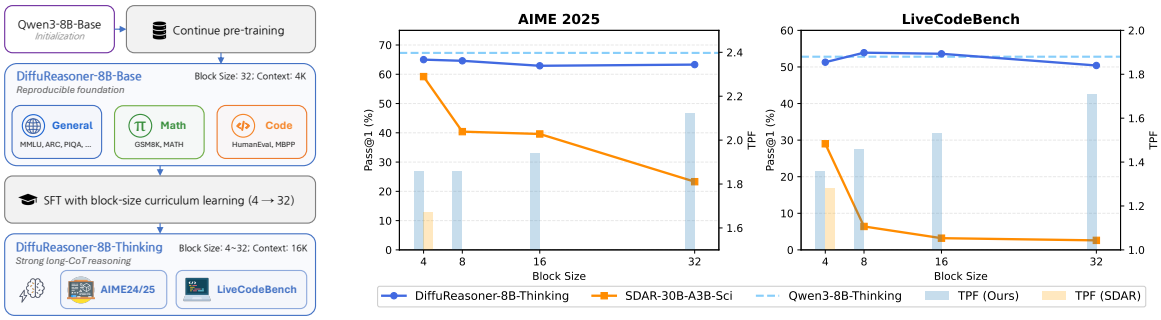

Figure 1: Overview and performance of DIFFUREASONER-8B. **Left:** the training pipeline of DIFFUREASONER-8B-Base and DIFFUREASONER-8B-Thinking through block-size curriculum learning. **Right:** DIFFUREASONER-8B-Thinking maintains high accuracy close to the Qwen3-8B-Thinking baseline while its efficiency (measured by TPF[1]) improves with larger block sizes. In contrast, SDAR-30B-A3B-Sci shows lower efficiency and its accuracy degrades sharply as the block size increases, highlighting the superior balance between performance and computational cost achieved by our model.

ated with large-block training. Third, curriculum-trained models achieve substantial efficiency gains from large-block inference without sacrificing reasoning quality. On mathematical and code reasoning benchmarks, DIFFUREASONER-8B-Thinking achieves performance competitive with leading open autoregressive models such as Qwen3-8B-Thinking (Yang et al., 2025a), substantially outperforming prior block diffusion models of comparable scale.

Nevertheless, we find that these gains remain fundamentally bottlenecked by conservative token-level confidence thresholds during decoding. To analyze this bottleneck, we introduce RelaxedConfidence, an analytical decoding probe that examines how contiguous spans of high-confidence tokens can be committed earlier through relaxed thresholds. This yields average TPF (tokens per forward pass) gains of 22.5% and 54.5% in the thinking and answering phases without compromising reasoning fidelity.

Our contributions are summarized as follows:

- We systematically investigate block size as a scaling axis for diffusion reasoning models, demonstrating that large-block training is brittle at scale, whereas curriculum learning substantially stabilizes large-block generation.
- We release DIFFUREASONER-8B-Thinking, an open block-diffusion reasoning model that achieves competitive reasoning performance on par with Qwen3-8B-Thinking.
- We introduce RelaxedConfidence, an analytical decoding probe revealing that relaxing the commitment threshold for tokens surrounded by reliable neighbors enables earlier and broader parallel decoding.

[1]We regard the TPF of autoregressive models as 1. For SDAR, we report TPF only at block size 4, since its accuracy degrades substantially at larger block sizes.

2 Preliminaries

Full-sequence discrete diffusion processes jointly model all token positions in an input sequence. This provides bidirectional context and flexible decoding orders, but it also couples denoising across the entire sequence, which can slow training convergence on large corpora (Ni et al., 2025; Prabhudesai et al., 2025) and make long-form generation vulnerable to error accumulation. Block diffusion mitigates these issues by combining autoregressive sequence decomposition with local parallel denoising, enabling scalable training and efficient inference (Han et al., 2023; Arriola et al., 2025; Bie et al., 2025; Cheng et al., 2025; Fu et al., 2026).

Let $\mathbf{x}_0$ denote the clean input sequence, and partition it into K contiguous blocks $\mathbf{x}_0 = (b_0^1, b_0^2, \ldots, b_0^K)$, each containing B tokens. We use superscripts to index blocks and subscripts to index diffusion time steps; thus $b_t^k = (x_t^{k,1}, \ldots, x_t^{k,B})$ denotes the state of the k-th block at time t, and b_0^k is its clean state. Block diffusion preserves an autoregressive factorization across blocks while modeling each block conditional with a local bidirectional diffusion process:

$$p_\theta(\mathbf{x}_0) = \prod_{k=1}^{K} p_\theta(b_0^k \mid b_0^{<k}), \qquad (1)$$

where $b_0^{<k} := (b_0^1, \ldots, b_0^{k-1})$. Each conditional $p_\theta(b_0^k \mid b_0^{<k})$ is parameterized by a local *diffusion*

Table 1: Base model performance comparison on standard benchmarks. Best results are shown in bold, and second-best results are underlined.

Model	DIFFUREASONER-8B	Dream-v0-7B	LLaDA-8B	Qwen3-8B	MiMo-7B	OLMo3-7B	Qwen2.5-7B
Type	Block Diff.	Diffusion	Diffusion	AR	AR	AR	AR
General Tasks							
MMLU	75.4	69.5	65.9	**76.9**	69.8	64.5	71.9
ARC-E	**87.0**	83.9	71.8	81.6	76.5	80.6	77.4
ARC-C	**63.5**	59.8	47.5	56.9	51.6	53.5	51.5
HellaSwag	76.6	73.3	72.7	78.6	77.5	74.1	**78.9**
PIQA	**81.2**	75.8	75.8	79.4	79.9	78.6	79.8
WinoGrande	74.2	74.5	74.5	**77.5**	75.9	73.5	76.1
RACE	43.8	**44.7**	38.7	42.3	38.6	40.1	41.9
Math & Science							
GSM8K	83.4	77.2	70.9	**86.5**	73.8	76.5	78.9
MATH	**55.8**	39.6	30.7	52.7	35.8	41.5	41.1
GPQA	**42.9**	36.6	30.4	41.1	–	36.6	35.5
Code							
HumanEval	**69.5**	57.9	32.9	68.9	51.8	43.9	56.7
MBPP	71.7	56.2	39.0	**72.4**	69.2	66.7	63.6

process that processes all tokens within b^k in parallel. The process uses an absorbing corruption procedure with a dedicated absorbing token [MASK]. For the k-th block, the forward process corrupts only the target block while keeping the prefix context clean. Generation reverses this process: starting from a fully masked block state, the model iteratively predicts $p_\theta(x_0^{k,i} \mid b_0^{<k}, b_t^k)$ for masked positions and refines them in parallel, conditioned on fixed preceding blocks $b_0^{<k}$ as context.

Training objective. The objective can be reduced to a weighted cross-entropy on the masked positions for each block (Arriola et al., 2025; Cheng et al., 2025; Bie et al., 2025), conditioned on the clean previous context $b_0^{<k}$. Let $\alpha_t \in (0,1)$ denote the noise schedule, $\alpha_t' := \frac{d\alpha_t}{dt}$, and $w_t := \frac{\alpha_t'}{1-\alpha_t}$ denote the loss weight at time step t. Aggregating over all K blocks yields:

$$\mathcal{L}(\theta) = -\mathbb{E}_{t,b_0,b_t}\left[\sum_{k=1}^{K} w_t \log p_\theta\left(b_0^k \,|\, b_0^{<k}, b_t^k\right)\right], \quad (2)$$

where $t \sim \mathcal{U}(0,1)$, $b_0 = \{b_0^k\}_{k=1}^K$ represents the block sequence of the clean input data, and $b_t = \{b_t^k\}_{k=1}^K$ is its corrupted version at time t. For each block, only masked tokens contribute to the loss with their context provided by clean prefix blocks.

3 DIFFUREASONER-8B-Base

Following previous practice (Gong et al., 2025a; Ye et al., 2025; Cheng et al., 2025; Fu et al., 2026), we first initialize from Qwen3-8B-Base (Yang et al., 2025a) and continually pre-train DIFFUREASONER-8B-Base as a block-diffusion model, establishing a reproducible foundation for subsequent fine-tuning. We use the block-diffusion loss objective in Eq. 2 with block size 32 throughout continual pretraining.

Data Composition. We assemble a pretraining corpus by curating high-quality, open-source datasets used by open-source language models, including OLMo 3 (Olmo et al., 2026) and Nemotron Nano V3 (NVIDIA et al., 2025). The data mixture is designed to enhance reasoning capabilities, and the final training corpus contains 160B tokens.

Baselines. We compare DIFFUREASONER-8B-Base against two categories of open-source baselines with settings listed in Appendix A: (i) *Autoregressive base models*, including Qwen3-8B (Yang et al., 2025a), MiMo-7B (Xiaomi et al., 2025), OLMo3-7B (Olmo et al., 2026), and Qwen2.5-7B (Yang et al., 2025b); and (ii) *Diffusion base models*, including the full-sequence diffusion models Dream-v0-7B (Ye et al., 2025) and LLaDA-8B (Nie et al., 2025b).[2]

Reasoning capabilities. DIFFUREASONER-8B-Base demonstrates performance competitive with autoregressive baselines, exhibiting particular strength in reasoning-intensive benchmarks for

[2]We do not compare with LLaDA-2.0 (Bie et al., 2025) and SDAR-8B (Cheng et al., 2025) since their base models are not open-sourced.

Table 2: Comparison of diffusion (LowConfidence) and autoregressive (AR) decoding strategies across varying training and inference block sizes on mathematical reasoning benchmarks.

Training	Inference	AIME 2024		AIME 2025		MATH500	
		LowConfidence	AR	LowConfidence	AR	LowConfidence	AR
Block Size 4	Block Size 4	47.1	52.1	37.5	40.8	86.8	85.6
	Block Size 32	42.5	52.1	39.6	38.8	82.8	84.2
Block Size 32	Block Size 4	20.0	29.2	24.2	22.5	78.8	78.2
	Block Size 32	42.5	48.3	21.3	33.3	80.2	83.4
Curriculum 4→32	Block Size 4	50.0	51.2	43.8	37.6	87.8	85.6
	Block Size 32	48.3	49.2	38.3	36.7	85.2	85.8

math, science, and code generation. Specifically, it outperforms Qwen3-8B-Base on advanced reasoning tasks such as MATH and GPQA, and achieves competitive performance on code generation. DIFFUREASONER-8B-Base surpasses Dream-v0-7B-Base by a substantial margin, improving MATH performance from 39.6 to 55.8 and HumanEval from 57.9 to 69.5. Ablation studies in Appendix B further show that block diffusion consistently yields stronger reasoning performance than ordinary full-sequence diffusion models initialized with the same base model checkpoint.

4 From Empirical Insights to DIFFUREASONER-8B-Thinking

4.1 Pilot Study: Block-Size Trade-offs

To understand when block diffusion can preserve reasoning quality while increasing generation parallelism, we isolate the two endpoints of the block-size design space. Small blocks provide a stronger token-level causal scaffold but offer limited intra-block parallelism; Larger blocks expose more tokens to parallel denoising but place greater pressure on the model to maintain sequential fidelity within each block. We therefore conduct a pilot study on a filtered subset of PromptCoT 2.0 (Zhao et al., 2025), retaining only mathematical examples with context lengths under 4096 tokens, and compare two fixed-granularity training strategies: Block Size 4 and Block Size 32.

Evaluation. We assess these configurations under two decoding strategies. (1) **LowConfidence**: following prior work (Wu et al., 2025b; Cheng et al., 2025; Bie et al., 2025), we unmask tokens whose confidence scores exceed a threshold of $\tau = 0.95$ at each denoising step. (2) **Autoregressive**: As the canonical baseline, we adopt the standard left-to-right generation protocol, where at each step only the leftmost masked token is unmasked, strictly preserving the causal generation order. Evaluation is performed on AIME 2024, AIME 2025, and MATH500 (Hendrycks et al., 2021).

Granularity Alignment Under Diffusion Decoding. When evaluating under LowConfidence diffusion decoding, our results highlight a clear contrast between fine-grained and coarse-grained training. Direct training with large blocks (Block Size 32) suffers immense capability loss compared to its Block Size 4 counterpart, especially under fine-grained inference, where it reaches only 20.0% on AIME 2024 and 24.2% on AIME 2025 with Block Size 4 inference. This degradation demonstrates that coarse-block training forces the model to overfit to intra-block long-range dependencies, rendering it incapable of faithfully decomposing complex reasoning into fine-grained denoising steps. Conversely, Block Size 4 training exhibits remarkable robustness: it not only establishes strong performance in small-block settings but also maintains stable efficacy when evaluated with Block Size 32 inference suggesting that local structures learned from fine-grained training can organically support larger decoding blocks.

Decoding Strategy: Autoregressive vs. Low-Confidence Diffusion. Ni et al. (2026) reveal that arbitrary generation orders in full-sequence diffusion models often cause them to bypass high-uncertainty tokens, thereby collapsing the reasoning solution space, a phenomenon termed the *flexibility trap*. To investigate whether block-wise diffusion encounters a similar trap, we compare decoding behaviors across granularities.

Our evaluation reveals that training granularity strongly affects decoding resilience. With Block Size 4 training, the model achieves high absolute performance across both paradigms. While

Table 3: Performance comparison of open-source models on mathematics and code benchmarks.

Model	Block Size	AIME 2024	AIME 2025	LiveCodeBench
Autoregressive Baselines				
DeepSeek-R1-Distill-Qwen-7B	-	55.5	39.0	37.8
MiMo-7B-RL	-	68.2	55.4	50.7
AceReason-Nemotron-1.1-7B	-	72.6	64.8	52.1
Qwen3-8B-Thinking	-	76.0	67.3	52.8
Diffusion Baselines				
Dream-7B-Instruct	-	0.0	3.3	9.7
LLaDA-8B-Instruct	-	0.0	0.0	-
LLaDA-2.0-Flash (100B-A6B)	32	-	60.0	41.0
SDAR-8B-Chat	4	13.8	10.4	16.4
TraDo-8B-Instruct	-	13.3	10.8	16.1
TraDo-8B-Thinking	8	35.5	17.5	14.2
SDAR-30B-A3B-Sci	4	73.4	59.2	29.0
SDAR-30B-A3B-Sci	8	52.5	40.4	6.4
SDAR-30B-A3B-Sci	16	44.2	39.6	3.2
SDAR-30B-A3B-Sci	32	47.1	23.3	2.6
Ours				
DIFFUREASONER-8B-Thinking	4	73.8	65.0	51.3
DIFFUREASONER-8B-Thinking	8	71.7	64.6	53.9
DIFFUREASONER-8B-Thinking	16	71.7	62.9	53.6
DIFFUREASONER-8B-Thinking	32	68.3	63.3	50.4

AR decoding provides a reliable performance ceiling (e.g., 52.1% on AIME 2024), the model retains strong capabilities under diffusion decoding (47.1%), implying it has successfully internalized robust local causal dependencies. In contrast, coarse-grained Block Size 32 training is substantially weaker under diffusion decoding on complex tasks (e.g., dropping to 21.3% on AIME 2025), while strict AR decoding recovers part of the gap at 33.3%. This asymmetry indicates that direct large-block training fosters an over-reliance on parallel intra-block aggregation, compromising token-level sequential reasoning.

4.2 Block-Size Curriculum Learning

The pilot study exposes a granularity dilemma in block diffusion language models. To bridge this gap, we use a **block-size curriculum learning** framework for supervised fine-tuning our continually pretrained DIFFUREASONER-8B-Base (§3).

Starting from this foundation, the curriculum first trains with Block Size 4 to acquire robust local causal dependencies and fine-grained reasoning patterns, and then exposes the model to larger blocks. In the pilot study, this second stage uses Block Size 32 to stress-test the block sizes.

Curriculum Learning Unifies Granularity Robustness. As shown by the curriculum rows in Table 2, the progressive curriculum (Block Size $4 \rightarrow 32$) emerges as the most robust diffusion-decoding performance across inference granularities. On AIME 2024, the curriculum model yields 50.0% under Block Size 4 inference and 48.3% under Block Size 32 inference, surpassing both fixed-block baselines. Similar trends are established on AIME 2025. Although the curriculum approach slightly trades off peak AR performance in some settings (e.g., $52.1\% \rightarrow 51.2\%$ on AIME 2024 compared to pure Block Size 4 training), it improves overall robustness. These findings suggest that gradual exposure to increasing block sizes preserves the fine-grained precision essential for small-block parallel denoising, while improving stability under large-block generation, reducing the need for maintaining multiple granularity-specific models.

4.3 DIFFUREASONER-8B-Thinking

Building upon the curriculum framework in Section 4.2, we train our final model, DIFFUREASONER-8B-Thinking, on the complete Prompt-CoT 2.0 dataset (Zhao et al., 2025), encompassing approximately 4.8 million samples with context length 16384 tokens, spanning mathematical reasoning and code generation tasks. We initiate training with block size 4 and then transition to mixed-granularity training with block sizes randomly sampled from $\{4, 8, 16, 32\}$.

Baselines. We compare against three categories of open-source models: (1) **Autoregressive baselines**: DeepSeek-R1-Distill-Qwen-7B (DeepSeek-

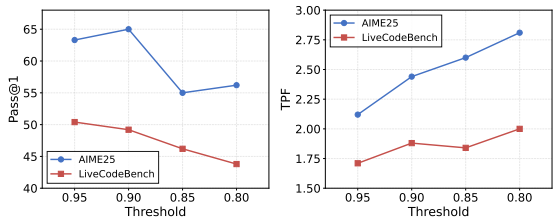

Figure 2: Effect of confidence threshold on reasoning accuracy and decoding efficiency (Block Size=32).

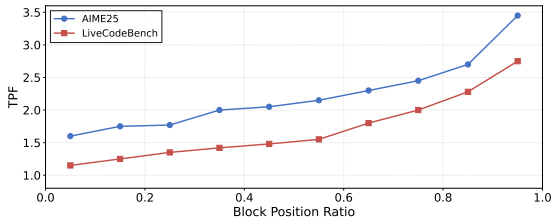

Figure 3: Tokens per forward pass across relative block positions under LowConfidence decoding (τ = 0.95).

AI, 2025), MiMo-7B-RL (Xiaomi et al., 2025), AceReason-Nemotron-1.1-7B (Chen et al., 2025), and Qwen3-8B-Thinking (Yang et al., 2025a); (2) **Full-attention diffusion baselines**: LLaDA-8B (Nie et al., 2025b) and Dream-7B (Ye et al., 2025); (3) **Block diffusion baselines**: General-purpose instruct models, including LLaDA-2.0-Flash (Bie et al., 2025), TraDo-8B-Instruct (Wang et al., 2025), and SDAR-8B-Chat (Cheng et al., 2025), alongside reasoning-optimized variants with extended chain-of-thought capabilities, TraDo-8B-Thinking (Wang et al., 2025) and SDAR-30B-A3B-Sci (Cheng et al., 2025).

Evaluation. We assess mathematical reasoning through accuracy on AIME 2024 and AIME 2025, and measure code generation proficiency via pass@1 on LiveCodeBench (Oct 2024–May 2025) (Jain et al., 2024). To assess architectural generalizability across varying block sizes, both SDAR-30B-A3B-Sci and our model are comprehensively evaluated at inference block sizes of 4, 8, 16, and 32. Remaining diffusion models are evaluated at their native training block sizes. All diffusion models are decoded using the LowConfidence strategy with threshold $\tau = 0.95$.

Comparison with Autoregressive Baselines. DIFFUREASONER-8B-Thinking achieves competitive performance against 8B-scale autoregressive reasoning models. Through continual pretraining and block-size curriculum learning, our model substantially narrows the gap with its autoregressive counterpart Qwen3-8B-Thinking. These results demonstrate that diffusion-based architectures, when properly trained, offer a viable alternative to conventional autoregressive approaches for complex reasoning tasks.

Comparison with Diffusion Baselines. Our model establishes a new performance frontier among open-source diffusion language models. Full-sequence diffusion language models such as LLaDA and Dream struggle to produce meaningful results on complex reasoning benchmarks. Among block diffusion approaches, DIFFUREASONER-8B-Thinking outperforms same-scale reasoning variants such as TraDo-8B-Thinking by a large margin, while surpassing substantially larger models including LLaDA-2.0-Flash (100B-A6B) and SDAR-30B-A3B-Sci across both mathematical and code benchmarks. The advantage is particularly pronounced on LiveCodeBench, where our model achieves a pass@1 score of 51.3% compared with SDAR-30B-A3B-Sci's 29.0%.

Robustness to Inference Block Sizes. As shown in Table 3, DIFFUREASONER-8B-Thinking maintains stable performance across block sizes from 4 to 32, with only modest degradation on mathematical reasoning and consistently strong code generation. This resilience contrasts sharply with SDAR-30B-A3B-Sci, whose performance degrades substantially when inference block sizes exceed its training configuration. These results demonstrate that our curriculum learning framework decouples reasoning quality from inference block granularity, enabling greater generation parallelism without the severe accuracy penalties observed in prior block diffusion models.

5 Analysis

5.1 Efficiency Analysis

To better reflect the decoding efficiency, we measure tokens per forward pass (TPF) as the number of generated tokens divided by the total number of forward passes, counting both token-prediction passes and KV-cache recomputation steps.

Block Size. Figure 1 characterizes the accuracy–efficiency trade-off as the inference block size varies. TPF improves monotonically with larger block sizes across both benchmarks, indicating that coarser decoding configurations can effectively reduce the number of forward passes required for

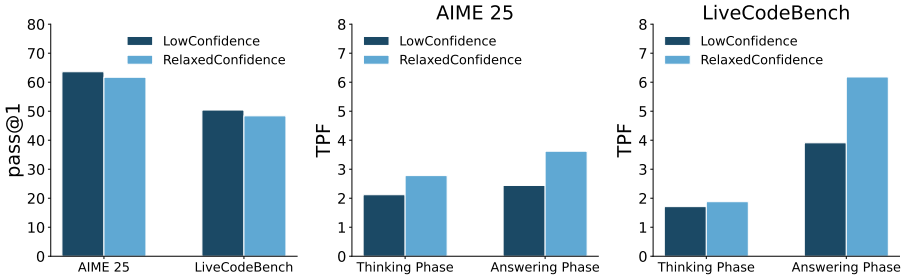

Figure 4: Pass@1 and throughput comparison between LowConfidence and RelaxedConfidence decoding. We define the answering phase as the final solution-writing segment for AIME and the final code block for LiveCodeBench.

generation. At the same time, task accuracy remains relatively stable.

Threshold. Figure 2 examines the effect of the confidence threshold on accuracy and TPF. Lowering the threshold permits more tokens to be decoded per forward pass, thereby improving TPF, yet risks accepting low-confidence predictions that compromise reasoning fidelity. Both benchmarks exhibit a general degradation trend as the threshold becomes more aggressive, with code generation showing particular sensitivity to token-level errors.

Thinking Process. We analyze decoding efficiency during the thinking and answering phases under LowConfidence decoding with a fixed confidence threshold of 0.95. Figure 3 illustrates the average TPF within each block as a function of its relative position in the generated response.

TPF exhibits a monotonic upward trend on both AIME 2025 and LiveCodeBench benchmarks. In early-stage blocks (relative position ≈ 0.0), the model commits only $\sim$1.2–1.6 tokens per forward pass, as the majority of tokens fall below the conservative threshold due to the scarcity of reasoning context. As generation progresses toward later stages (relative position ≈ 1.0), TPF rises to $\sim$2.8–3.5, since the accumulated reasoning trace furnishes sufficient contextual information to support higher degrees of parallel token generation.

5.2 Reasoning Acceleration Analysis

Our step-by-step analysis reveals that standard confidence-threshold decoding is often overly conservative, as contiguous token spans typically stabilize before their individual confidence scores exceed the threshold. This raises the question of whether such implicitly stable tokens can be committed earlier without sacrificing accuracy. To investigate this, we introduce RelaxedConfidence as an analytical decoding strategy.

Formulation. For a masked token at position i, let $\mathcal{N}_r(i) = \{j : 0 < |i-j| \leq r\}$ denote the valid positions within radius r, excluding i itself. Let $\mathcal{D}$ be the set of already decoded positions, and define the reliable-position set $\mathcal{R} = \mathcal{D} \cup \{j : c_j \geq \kappa\}$, where c_j is the model confidence for position j and κ is the neighbor-support threshold. With proximity weights $w_{i,j} = r - |i-j| + 1$, we compute a spatial support score and position-specific commitment threshold as

$$S_i = \frac{\sum_{j \in \mathcal{N}_r(i) \cap \mathcal{R}} w_{i,j}}{\sum_{j \in \mathcal{N}_r(i)} w_{i,j}}, \quad \tau_i = \tau - (\tau - \tau_{\min}) S_i.$$

Here τ is the standard LowConfidence threshold and $\tau_{\min}$ is the minimum threshold floor. A token at position i is committed when $c_i \geq \tau_i$. Thus, isolated tokens with $S_i = 0$ must satisfy the full threshold $\tau_i = \tau$, while well-supported tokens with $S_i = 1$ can be committed at the relaxed threshold $\tau_i = \tau_{\min}$. This interpolation allows RelaxedConfidence to decode contiguous spans more aggressively when their local context is reliable, while preserving the full threshold for isolated positions.

Acceleration Without Degradation. Figure 4 summarizes the impact of RelaxedConfidence on reasoning quality and decoding throughput compared to the standard low-confidence re-masking baseline. The leftmost panel reveals that RelaxedConfidence incurs only marginal degradation in accuracy on AIME25 and LiveCodeBench. This suggests that spatially informed threshold relaxation enables accelerated parallel decoding without compromising reasoning ability, likely because it primarily commits tokens that are well-supported by their local context.

Throughput Gains in Thinking and Answering. To localize where RelaxedConfidence accelerates decoding, we partition responses into a *thinking* phase and an *answering* phase. During thinking,

RelaxedConfidence raises TPF modestly—from $\sim$2.1 to $\sim$2.8 on AIME 25 ($+33\%$) and from $\sim$1.7 to $\sim$1.9 on LiveCodeBench ($+12\%$). Gains are substantially larger during answering: on AIME 25, TPF rises from $\sim$2.4 to $\sim$3.6 ($+50\%$); on LiveCodeBench, from $\sim$3.9 to $\sim$6.2 ($+59\%$). This reflects the more structured and lexically constrained nature of final responses, where contiguous token spans more readily reach implicit stability, allowing the relaxation criterion to commit longer coherent spans per pass.

6 Related Work

6.1 Diffusion Language Models

Diffusion language models generate text through iterative denoising over discrete token vocabularies (Austin et al., 2021a; Hoogeboom et al., 2021; Campbell et al., 2022) or continuous representations (Li et al., 2022; Gong et al., 2023; Han et al., 2023; Jo and Hwang, 2025). Training objectives and parameterizations for discrete diffusion have been extensively refined (Zheng et al., 2024; Lou et al., 2024; Shi et al., 2024; Zhao et al., 2024; Sahoo et al., 2024), with recent work scaling these models toward the modern LLM regime: Nie et al. (2025a) scaled to billion-parameter tasks; Gong et al. (2025a) adapted pretrained AR checkpoints to diffusion; Nie et al. (2025b) trained the 8B-parameter LLaDA from scratch; and Ye et al. (2025) demonstrated competitive large-scale diffusion LMs initialized from AR checkpoints.

However, full-sequence diffusion struggles to maintain global coherence over long contexts, particularly for chain-of-thought reasoning (Wang et al., 2025). Earlier approaches (Han et al., 2023; Wu et al., 2023; Ye et al., 2024) introduced semi-autoregressive or position-dependent denoising mechanisms for text generation. Block diffusion (Arriola et al., 2025) formalizes this idea by partitioning a sequence into contiguous blocks with autoregressive factorization and denoising tokens within each block in parallel. This architecture interpolates between pure AR generation and full-sequence diffusion. LLaDA2.0 (Bie et al., 2025), FastDLLM v2 (Wu et al., 2025a), and Efficient-DLM (Fu et al., 2026) extended this paradigm to larger-scale settings but lack long chain-of-thought reasoning. T* (Xia et al., 2026) and TraDo (Wang et al., 2025) studies progressive block scaling for masked diffusion models, increasing block size stage by stage to improve mathematical reasoning under larger blocks. Our work differs in treating block size as a deployment-time quality-efficiency variable and providing an in-depth analysis of the interplay between block size and decoding strategy.

6.2 Reasoning with Language Models

Reasoning in autoregressive language models has been substantially advanced by chain-of-thought techniques along with fine-tuning and reinforcement learning (OpenAI et al., 2026; DeepSeek-AI, 2025; Xiaomi et al., 2025; Yang et al., 2025a). Existing RL methods for diffusion language models spanning token-level likelihood surrogates (Zhao et al., 2026; Tang et al., 2025; Wang et al., 2026; Gong et al., 2025b) and sequence-level objectives (Zhu et al., 2025; Ou et al., 2025) have shown promise for improving reasoning, yet their evaluation remains confined to elementary math benchmarks such as GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021), leaving more complex real-world reasoning challenges such as AIME largely unexplored. Previous block diffusion reasoners lag behind autoregressive counterparts (Cheng et al., 2025; Wang et al., 2025) and face performance collapse when using large block sizes. Our proposed block-size curriculum mitigates the instability of direct large-block training, preserves the efficiency benefits of coarse-grained parallel decoding, and yields a strong open 8B block-diffusion reasoning model competitive with same-scale AR reasoning models.

7 Conclusion

We present DIFFUREASONER-8B, an open block-diffusion reasoning model that matches leading autoregressive baselines on math and code benchmarks. We show that fixed large-block training causes performance collapse while curriculum learning of block sizes stabilizes large-block reasoning. We further demonstrate that inference efficiency remains bottlenecked by conservative per-token commitment thresholds, and propose RelaxedConfidence, a context-aware decoding strategy that relaxes thresholds based on neighboring token confidence to accelerate parallel generation without compromising reasoning fidelity. We release our model and checkpoints to support future research on efficient diffusion-based reasoning.

Limitations

While block-size curriculum learning effectively stabilizes large-block training and decouples reasoning quality from inference granularity, we do not explore dynamic or variable-length block sizes that respect semantic or syntactic boundaries, such as sub-problem delimiters, code blocks, or natural paragraph breaks. Allowing the model to adaptively choose block granularity rather than fixing it at predetermined widths could further push the Pareto frontier between decoding efficiency and modeling quality. We leave the investigation of boundary-aware, content-adaptive block partitioning to future work.

Due to computational constraints, our model is primarily limited to mathematical reasoning and code generation. We do not systematically assess the proposed curriculum on broader tasks such as tool use (Yao et al., 2025) and coding agents (Jimenez et al., 2024). Generalizing block-size curriculum learning to these more diverse reasoning domains remains an important direction for future work.

References

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. 2025. Block diffusion: Interpolating between autoregressive and diffusion language models. *Preprint*, arXiv:2503.09573.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021a. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021b. Program synthesis with large language models. *Preprint*, arXiv:2108.07732.

Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, Zhuocheng Gong, Yanmei Gu, Jian Guan, Kaiyuan Guan, Hongliang He, Zenan Huang, Juyong Jiang, Zhonghui Jiang, Zhenzhong Lan, and 31 others. 2026. Llada2.1: Speeding up text diffusion via token editing. *Preprint*, arXiv:2602.08676.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, Chengxi Li, Chongxuan Li, Jianguo Li, Zehuan Li, Huabin Liu, Ling Liu, Guoshan Lu, Xiaocheng Lu, Yuxin Ma, and 12 others. 2025. Llada2.0: Scaling up diffusion language models to 100b. *Preprint*, arXiv:2512.15745.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. Acereason-nemotron: Advancing math and code reasoning through reinforcement learning. *Preprint*, arXiv:2505.16400.

Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and 1 others. 2025. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. 2024. Flex attention: A programming model for generating optimized attention kernels. *Preprint*, arXiv:2412.05496.

Yonggan Fu, Lexington Whalen, Zhifan Ye, Xin Dong, Shizhe Diao, Jingyu Liu, Chengyue Wu, Hao Zhang, Enze Xie, Song Han, Maksim Khadkevich, Jan Kautz, Yingyan Celine Lin, and Pavlo Molchanov. 2026. Efficient-dlm: From autoregressive to diffusion language models, and beyond in speed. *Preprint*, arXiv:2512.14067.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. 2025a. Scaling diffusion language models via adaptation from autoregressive models. *International Conference on Learning Representations*.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. DiffuSeq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations, ICLR*.

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

Google DeepMind. 2026. Gemini 3.1 Pro model card.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. 2023. SSD-LM: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

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

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. 2024. Swe-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jaehyeong Jo and Sung Ju Hwang. 2025. Continuous diffusion model for language modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 4328–4343. Curran Associates, Inc.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion language modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning, ICML*.

Jinjie Ni, Qian Liu, Longxu Dou, Chao Du, Zili Wang, Hang Yan, Tianyu Pang, and Michael Qizhe Shieh. 2025. Diffusion language models are super data learners. *arXiv preprint arXiv:2511.03276*.

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

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. 2025a. Scaling up masked diffusion models on text. *Preprint*, arXiv:2410.18514.

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

NVIDIA, :, Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, Aleksandr Shaposhnikov, Alex Kondratenko, Alexander Bukharin, Alexandre Milesi, Ali Taghibakhshi, Alisa Liu, Amelia Barton, and 295 others. 2025. Nemotron 3 nano: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning. *Preprint*, arXiv:2512.20848.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, and 50 others. 2026. Olmo 3. *Preprint*, arXiv:2512.13961.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 246 others. 2026. Openai o1 system card. *Preprint*, arXiv:2412.16720.

OpenAI. 2026. GPT-5.4 thinking system card. Technical report, OpenAI. System card and technical report.

Jingyang Ou, Jiaqi Han, Minkai Xu, Shaoxuan Xu, Jianwen Xie, Stefano Ermon, Yi Wu, and Chongxuan Li. 2025. Principled rl for diffusion llms emerges from a sequence-level perspective. *arXiv preprint arXiv:2512.03759*.

Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. 2025. Diffusion beats autoregressive in data-constrained settings. *arXiv preprint arXiv:2507.15857*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. 2024. Simplified and generalized masked diffusion for discrete data. *Neural Information Processing Systems*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

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

Chenyu Wang, Paria Rashidinejad, DiJia Su, Song Jiang, Sid Wang, Siyan Zhao, Cai Zhou, Shannon Zejiang Shen, Feiyu Chen, Tommi Jaakkola, Yuandong Tian, and Bo Liu. 2026. Spg: Sandwiched policy gradient for masked diffusion language models. *Preprint*, arXiv:2510.09541.

Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. 2025. Revolutionizing reinforcement learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. 2025a. Fast-dllm v2: Efficient block-diffusion llm. *Preprint*, arXiv:2509.26328.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2025b. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *Preprint*, arXiv:2505.22618.

Tong Wu, Zhihao Fan, Xiao Liu, Yeyun Gong, Yelong Shen, Jian Jiao, Hai-Tao Zheng, Juntao Li, Zhongyu Wei, Jian Guo, Nan Duan, and Weizhu Chen. 2023. Ar-diffusion: Auto-regressive diffusion model for text generation. *Preprint*, arXiv:2305.09515.

Hanchen Xia, Baoyou Chen, Yutang Ge, Guojiang Zhao, and Siyu Zhu. 2026. $T^{\star}$: Progressive block scaling for masked diffusion language models through trajectory aware reinforcement learning. *Preprint*, arXiv:2601.11214.

LLM-Core Xiaomi, :, Bingquan Xia, Bowen Shen, Cici, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, Liang Zhao, Peidian Li, Peng Wang, Shihua Yu, Shimao Chen, Weikun Wang, Wenhan Ma, and 46 others. 2025. Mimo: Unlocking the reasoning potential of language model – from pretraining to posttraining. *Preprint*, arXiv:2505.07608.

Zhihui Xie, Jiacheng Ye, Lin Zheng, Jiahui Gao, Jingwei Dong, Zirui Wu, Xueliang Zhao, Shansan Gong, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. Dream-coder 7b: An open diffusion language model for code. *Preprint*, arXiv:2509.01142.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025b. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. 2025. {τ}-bench: A benchmark for \underline{T}ool-\underline{A}gent-\underline{U}ser interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and Lingpeng Kong. 2024. Diffusion of thought: Chain-of-thought reasoning in diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 105345–105374. Curran Associates, Inc.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Lingxiao Zhao, Xueying Ding, Lijun Yu, and Leman Akoglu. 2024. Improving and unifying discrete&continuous-time discrete denoising diffusion. *arXiv preprint arXiv:2402.03701*.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. 2026. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *Advances in Neural Information Processing Systems*, 38:56729–56762.

Xueliang Zhao, Wei Wu, Jian Guan, Zhuocheng Gong, and Lingpeng Kong. 2025. Promptcot 2.0: Scaling prompt synthesis for large language model reasoning. *arXiv preprint arXiv:2509.19894*.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. 2024. A reparameterized discrete diffusion model for text generation. In *Conferenec on Language Modeling, COLM, October 7-9, 2024, Philadelphia, PA*.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *Preprint*, arXiv:2505.19223.

A Implementation Details

Training. We extend Megatron-LM (Shoeybi et al., 2019) to support block diffusion training with a no-shift data loader (Ni et al., 2025; Fu et al., 2026) that uses input sequences directly as supervision targets. A noise injection module stochastically masks tokens while enforcing at least one masked token per block to prevent degenerate cases. The resulting interleaved sequences are processed via FlexAttention (Dong et al., 2024), which compiles the structured sparse attention pattern into optimized kernels, with training loss computed exclusively on masked positions.

In §4.1, we train for 4 epochs in total. For the curriculum setting, the first 3 epochs use block size 4, followed by 1 epoch with block size 32. In §4.3, we first train with block size 4 for 3 epochs, then transition to mixed-granularity training with block sizes randomly sampled from $\{4, 8, 16, 32\}$. We intentionally preserve small block sizes in this final stage, as we observe that training solely with large blocks on longer thinking traces causes severe performance degradation during small-block inference.

Evaluation of base models. We evaluate DIFFUREASONER-8B-Base on a comprehensive suite of standard benchmarks spanning general knowledge, reasoning, mathematics, and code generation. *General tasks* include MMLU (Hendrycks et al., 2020) for multi-subject understanding, ARC-Easy and ARC-Challenge (Clark et al., 2018) for commonsense reasoning, HellaSwag (Zellers et al., 2019) for sentence completion, PIQA (Bisk et al., 2020) for physical reasoning, WinoGrande (Sakaguchi et al., 2021) for pronoun disambiguation, and RACE (Lai et al., 2017) for reading comprehension. *Mathematical and scientific reasoning* is assessed through GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2020) for mathematical problem-solving, and GPQA (Rein et al., 2023) for graduate-level science questions. *Code generation* capabilities are measured via HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021b). We follow the evaluation framework in Dream-7B (Ye et al., 2025) to evaluate our model on all benchmarks.

Evaluation of reasoning models. We adopt SGLang[3] as the inference engine. All baselines are reimplemented in our codebase, except for Dream

[3] https://github.com/sgl-project/sglang

Table 4: Comparison of full-sequence and block diffusion continually pretrained from Qwen3-8B on math and code reasoning benchmarks.

Model	GSM8K	MATH	HumanEval	MBPP
Qwen3-8B	86.5	52.7	65.8	68.8
DIFFUREASONER-8B-Base	83.5	51.2	62.8	60.4
w/o block diffusion	75.5	40.6	31.1	41.6

and LLaDA, whose results are taken from prior works (Wang et al., 2025; Xie et al., 2025). The maximum generation length is set to 24K for both autoregressive and block diffusion baselines. In pilot study (§4.1), it is set as 8192 tokens. We set the temperature to 0.8 for autoregressive baselines and to 0 for diffusion baselines, as the latter employ threshold-based decoding. We run each baseline 8 times and report the average to reduce randomness.

RelaxedConfidence. In RelaxedConfidence, we set the standard threshold to $\tau = 0.95$, the neighborhood radius to $r = 4$, and fix both the neighbor-support threshold κ and the minimum relaxed threshold $\tau_{\min}$ at 0.7.

Data and risks. We strictly comply with the licensing terms of Nemotron nano v3 (NVIDIA Nemotron Open Model License), OLMo 3, and PromptCoT 2.0 (both Apache 2.0). Our dataset contains no personal or personally identifiable information (PII). Given our exclusive focus on mathematical and programming data, we consider our model's risk profile to be minimal.

Usage of AI. AI tools assisted in manuscript preparation and code development; all generated content was rigorously reviewed by the authors.

B Continual Pretraining Ablation

We initialize both full-sequence and block diffusion from the same base model, Qwen3-8B (Yang et al., 2025a), and continually pretrain each variant on 100B high-quality tokens drawn from the Olmo3 mid-training corpus. We evaluate both models on standard math and code reasoning benchmarks.

As shown in Table 4, full-sequence diffusion lags behind block diffusion by a considerable margin on average across all evaluated tasks. These results demonstrate that block diffusion is substantially more effective for reasoning-intensive scenarios.

C Base Model Superiority

We conduct supervised fine-tuning with block size 4 for both DIFFUREASONER-8B and SDAR-8B (Cheng et al., 2025), employing identical hyperparameters and the PromptCoT 2.0 dataset. Figure 5 demonstrates that our pretrained base model substantially outperforms the baseline, achieving 67.1% on aime24 and 58.3% on aime25 compared to SDAR-8B's 64.2% and 47.0% respectively. These results correspond to absolute improvements of 2.9 and 11.3 percentage points, validating the superior quality of our pretraining recipe.

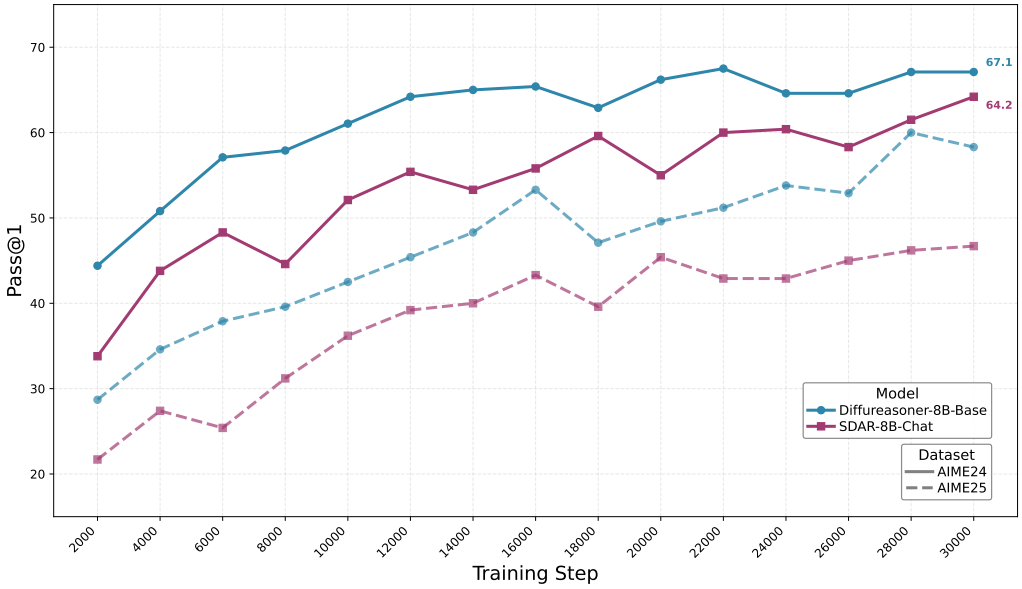

Figure 5: SFT convergence curves of DIFFUREASONER-1-8B vs. SDAR-8B on AIME 2024 and AIME 2025 (block size = 4). Solid lines denote AIME 2024 results; dashed lines denote AIME 2025 results.