OPTIMIZING DENOISING TRAJECTORIES IN DLLMS: A LIGHTWEIGHT EVOLUTIONARY HEURISTIC APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion Large Language Models (dLLMs) have recently emerged as a promising alternative to conventional Auto-Regressive (AR) Large Language Models (LLMs). By leveraging bidirectional attention and parallel decoding, dLLMs enable more efficient generation. However, they require a carefully designed denoising scheduler at inference time (absent during training) whose choice significantly impacts generation quality. While confidence-based heuristic schedulers have shown strong empirical performance, they suffer from two critical failure modes: EOS Overflow and Proximal Bias. Through in-depth analysis of the Transformer's attention patterns, we reveal that these failures stem from certain positions assigning disproportionately high attention weights to invalid tokens (e.g., [MASK] and [EOS]), which produce misleading confidence signals. Building on this insight, we show that valid attention scores provide complementary guidance to conventional confidence-based heuristics, yet no single metric consistently excels across all scenarios, implying that the optimal denoising trajectory is highly context-dependent. To address this problem, we propose a lightweight evolutionary heuristic scheduler optimized using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Our scheduler dynamically integrates multiple heuristic features with a contextual mean-field embedding, while requiring only 393 trainable parameters. Evaluated on LLaDA and Dream across four reasoning and planning benchmarks, our method outperforms strong baselines in most settings, including conventional heuristics, block auto-regressive methods, and recent State-Of-The-Art (SOTA) approaches. To the best of our knowledge, it represents the most parameter-efficient neural scheduler to date. Our code is available at anonymous repository `https://anonymous.4open.science/r/Evo-Denoiser-CE0D`.

1 INTRODUCTION

Auto-Regressive (AR) Large Language Models (LLMs) Achiam et al. (2023); Touvron et al. (2023); Liu et al. (2024) have achieved remarkable success across diverse domains Chen et al. (2026a). However, their inherently sequential generation paradigm suffers from slow decoding speed and error accumulation Arbuzov et al. (2025). In recent years, Diffusion Large Language Models (dLLMs) Nie et al. (2026); Ye et al. (2025b) have emerged as a compelling alternative. By leveraging bidirectional attention and parallel decoding, dLLMs capture full contextual information at every step and enable efficient generation. Recent studies have further demonstrated that discrete dLLMs exhibit scaling laws Nie et al. (2025) and modality expansion capabilities You et al. (2026); Zhu et al. (2025) comparable to those of AR LLMs.

Despite these advantages, dLLMs face a critical train-inference mismatch. During training, dLLMs randomly mask tokens and learn to recover them in parallel under a maximum likelihood Evidence Lower Bound (ELBO) objective (Nie et al., 2026). At inference, generation starts from a fully masked sequence and proceeds through progressive denoising steps. Since the optimal denoising order is never explicitly supervised, the design of the denoising scheduler becomes crucial. Prior work has shown that the choice of scheduling strategy significantly affects final generation quality Huang et al. (2026a); He et al. (2025); Tang et al. (2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Inspired by confidence-based metrics in AR LLMs (e.g., top-1 probability, entropy, and Gini impurity) Chen et al. (2025; 2026c); Kang et al. (2026), recent studies have adopted similar heuristics to guide token denoising in dLLMs Nie et al. (2026); Ben-Hamu et al. (2026). Although selecting the top-k most confident tokens substantially outperforms random ordering, confidence-based schedulers still suffer from two persistent failure modes that notably degrade performance, especially in long-form reasoning and planning tasks:

- **EOS Overflow** Kim et al. (2026b;a); Park et al. (2026): Excessive EOS tokens accumulate in the rightmost part of the sequence, particularly when they are denoised prematurely.
- **Proximal (Local) Bias** Kim et al. (2026b); Piskorz et al. (2026): Once a token is denoised, its neighboring positions receive disproportionately high confidence scores.

To address these issues, a growing body of work has focused on improved denoising schedulers, which can be broadly categorized into three types: (i) manually designed heuristics that are computationally efficient and training-free, yet whose optimality is difficult to verify Cao et al. (2026); Park et al. (2026); (ii) Block-AR (Semi-AR) methods that perform parallel denoising within blocks but sequential decoding across blocks Wu et al. (2025); Zhang et al. (2026), inherently limiting both inference speed and generation quality; and (iii) trainable schedulers based on SFT or RL that learn an auxiliary network Hong et al. (2026); Kim et al. (2026b); Jazbec et al. (2026); Huang et al. (2026a); He et al. (2025), which are effective but incur substantial training costs (e.g., up to 134M parameters Hong et al. (2026)). Despite their diversity, these approaches either rely on fixed empirical rules that fail to adapt to varying contexts, or suffer from prohibitive training overhead. This raises a fundamental question: *Is it possible to achieve adaptive, context-aware denoising scheduling with minimal training cost?*

In this paper, we first conduct an in-depth analysis of the Transformer's attention mechanism, revealing that both EOS Overflow and Proximal Bias stem from certain positions excessively attending to invalid tokens such as [MASK] and [EOS]. Building upon this insight, we demonstrate that valid attention scores serve as a strong complementary signal to conventional confidence-based heuristics. Nevertheless, extensive empirical evidence shows that no single heuristic, whether confidence-based or attention-based, consistently dominates, highlighting that the optimal denoising trajectory is highly context-dependent. To address this challenge, we propose a lightweight evolutionary heuristic scheduler optimized using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) Hansen & Ostermeier (2001); Hansen (2016). Our scheduler dynamically integrates multiple heuristic features with a contextual mean-field embedding, while requiring *only 393 trainable parameters*, making it, to the best of our knowledge, the most parameter-efficient neural scheduler to date. Evaluated on LLaDA Nie et al. (2026) and Dream Ye et al. (2025b) across four challenging reasoning and planning benchmarks, our method outperforms strong baselines in most settings, including conventional heuristics, block-AR methods, and recent State-Of-The-Art (SOTA) approaches, delivering substantial improvements particularly in difficult settings with limited denoising budgets.

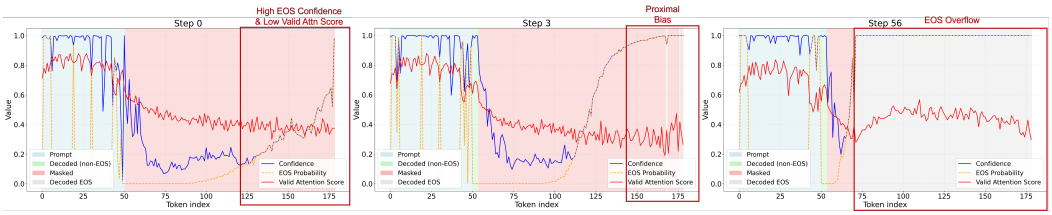
A brief background on dLLMs and the denoising scheduler formulation is provided in Appendix A.

2 Failure Analysis of Heuristic Schedulers

In this section, we analyze the top-1 probability denoising scheduler as a representative example to illustrate the underlying causes of EOS Overflow and Proximal Bias through the lens of the Transformer's attention mechanism. We also provide both empirical and intuitive evidence demonstrating that conventional single-token heuristic metrics are insufficient for determining the optimal denoising strategy.

2.1 Why Confidence Is Not Enough

We first formally define the heuristic scheduler for dLLM denoising. Generation begins from a fully masked sequence $x_T = [\text{[MASK]}]^L$, where L is the sequence length (the prompt preceding x_T is omitted for simplicity), and at each step t the set of positions that remain masked is $\mathcal{M}_t = \{i \mid x_{t,i} = \text{[MASK]}\}$.



(a) Step 0 (b) Step 3 (c) Step 56

Figure 1: Denoising process visualization on a GSM8K example using LLaDA-8B-Instruct. Different background colors represent prompt (blue), [MASK] (red), and [EOS] (gray) regions. The curves show top-1 probability (blue), valid attention score (red), and EOS probability (yellow).

The dLLM produces intermediate representations $\mathrm{f}_\theta(x_t)$ (e.g., hidden states or logits), which are passed through a scorer $\mathrm{h}(\cdot)$ to obtain denoising scores:

$$s_{t,i} = \mathrm{h}(\mathrm{f}_\theta(x_{t,i})), \tag{1}$$

where $\mathrm{h}(\cdot)$ extracts features such as top-1 probability, entropy, or margin probability. The top-k highest-scoring masked tokens are then selected for denoising:

$$\mathcal{I}_t = \text{Top-K}\big(s_{t,i} \mid i \in \mathcal{M}_t\big). \tag{2}$$

We use the top-1 probability heuristic as a case study to uncover the root causes of EOS Overflow and Proximal Bias, which have not been thoroughly explained in prior work. Since dLLMs are built upon the Transformer architecture, we analyze their behavior through attention patterns. At step t, for the m-th attention head in the n-th layer, the attention matrix is:

$$A_t^{n,m} = \text{Softmax}\left(\frac{Q_t^{n,m}(K_t^{n,m})^\top}{\sqrt{d_k}}\right), \tag{3}$$

where $Q_t^{n,m}$ and $K_t^{n,m}$ are the query and key matrices, respectively, and d_k is the head dimension.

Unlike AR LLMs, dLLM sequences contain many [MASK] tokens and trailing [EOS] tokens, both of which often receive uninformative attention. We therefore define the valid attention score for each token as:

$$a_{t,i}^n = \frac{1}{H}\sum_{m=1}^{H}\sum_j A_t^{n,m}[i,j] \cdot \mathbf{1}\{x_{t,j} \in \mathcal{V}\}, \tag{4}$$

where $\mathcal{V}$ is the set of valid (non-[MASK], non-[EOS]) vocabulary tokens and H is the number of attention heads.

Fig. 1 visualizes the denoising process on a GSM8K sample (here, step i corresponds to time step $T - i$ for readability). The valid attention scores are computed from the middle layer as a representative example. We observe the following:

- **Step 0 (initial):** Both beginning and ending tokens exhibit high top-1 probabilities. While high confidence at the beginning is expected due to strong prompt context, high confidence at the end is surprising. These ending tokens show low valid attention scores but high EOS prediction probabilities, indicating that they assign excessive attention to uninformative [MASK] tokens, resulting in misleadingly high confidence.
- **Step 3:** Several ending tokens have been decoded as [EOS]. Neighboring tokens then exhibit a sharp increase in both top-1 probability and EOS probability, which is a clear manifestation of **Proximal Bias**. Their valid attention scores decrease compared to Step 0, suggesting increased attention to the newly decoded (but uninformative) [EOS] tokens.
- **Step 56:** From Step 3 to Step 56, only ending tokens continue to be decoded as [EOS], while beginning tokens remain largely unchanged. This illustrates the **EOS Overflow** phenomenon, where [EOS] tokens occupy an excessive portion of the sequence.

These observations reveal that both failure modes originate from excessive attention to invalid tokens: ending tokens initially over-attend to [MASK] tokens and decode prematurely as [EOS], and the newly generated [EOS] tokens then attract further attention, causing neighboring tokens to follow suit. This cascading effect leads to severe Proximal Bias and EOS Overflow. Crucially, these

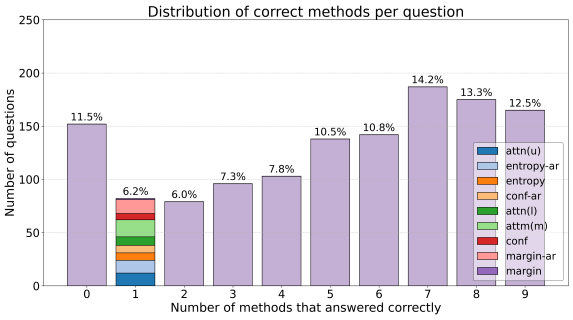


Figure 2: Number of correctly answered GSM8K questions per scheduler (generation length 128, budget 32). Bars give individual and combined coverage; the annotation marks questions solved by exactly one scheduler.

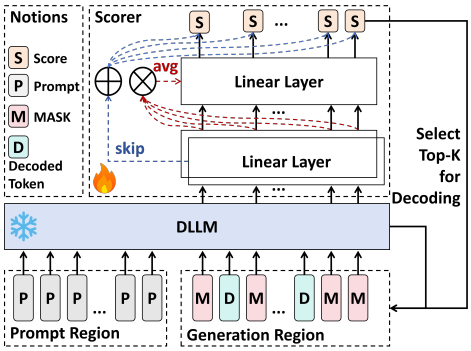


Figure 3: Architecture of the scorer. A linear bypass path (blue) and a two-layer mean-field MLP (red) are summed; the mean-field embedding $\bar{y}_t$ injects the global state of the masked region into the second layer. Total: 393 parameters at $h = 32$.

findings suggest that a purely confidence-based scheduler is fundamentally limited: it cannot distinguish between genuine predictive certainty and attention-driven false confidence, motivating our exploration of alternative signals.

2.2 Why We Need Context with Multiple Heuristics

To further investigate the limitations of single-heuristic schedulers, we evaluate several confidence-based methods on GSM8K with a generation length of 128 and a denoising budget of 32. The tested heuristics include top-1 probability, entropy, margin probability, and their Block-AR variants (block size 32). Motivated by our attention analysis, we also examine valid attention scores computed separately from upper, middle, and lower layers (inspired by prior findings that different layers capture distinct syntactic and semantic information Vig (2019); Jawahar et al. (2019)).

Fig. 2 shows the number of correctly answered questions for each scheduler, with special emphasis on questions solved uniquely by one method. Detailed results are provided in Table 1. Our analysis yields several key insights:

- Although individual scheduler accuracies range from 46.2% to 66.6% (Table 1), only 11.5% of questions are answered incorrectly by *all* schedulers (Fig. 2). This indicates that dLLMs possess significantly stronger problem-solving capability than current schedulers can elicit. In other words, if an oracle scheduler existed, the upper-bound performance would be at least 88.5%, which is comparable to or even exceeds some post-training results via SFT or RLZhao et al. (2026); Xie et al. (2026). This highlights the critical importance of optimizing the denoising trajectory.
- Each heuristic exhibits unique strengths: certain questions are solved exclusively by one scheduler but not others. This leads to two important conclusions: (i) no single manually designed heuristic is universally sufficient, and combining multiple complementary heuristics is required; (ii) the optimal denoising strategy is highly context-dependent, as no single fixed heuristic is suitable for all questions.

To further illustrate why context matters, consider the interplay between attention and confidence: a high valid attention score may indicate sufficient information flow, but alone it cannot determine whether the token is ready to be denoised. However, when accompanied by a high top-1 probability, this combined signal provides stronger evidence. Moreover, the overall magnitude of attention scores varies with sequence length, as longer prompts naturally provide richer valid contexts. This observation underscores the necessity of context-aware scheduling: the decision to denoise a token cannot be made in isolation but must account for the global state of the sequence.

3 METHODOLOGY

Motivated by the insights from our failure analysis, we propose an evolutionary heuristic scheduler that learns to dynamically combine multiple heuristics for superior denoising quality. Following the formulation in Eq. (1), we parameterize the scorer as $\mathrm{h}_\Theta(\cdot)$. Directly optimizing the scorer parameters via policy-gradient methods (e.g., PPO Schulman et al. (2017), GRPO Shao et al. (2024)) is prohibitively difficult: the token selection operation in Eq. (2) is discrete and non-differentiable, and backpropagating through the entire dLLM is computationally infeasible. We therefore reformulate scheduler optimization as a black-box problem and adopt CMA-ES Hansen & Ostermeier (2001); Hansen (2016), a zero-order evolutionary strategy that only requires evaluating the final task accuracy as the fitness signal. Although our method does not explicitly model long-term future effects through the Bellman equation as in RL-based schedulers Huang et al. (2026b); Hong et al. (2026), it is not myopic: because the scorer is shared across all denoising steps, optimizing for final answer accuracy implicitly captures temporal dependencies among successive decisions.

3.1 FEATURE CONSTRUCTION

We formalize the input features for our evolutionary scorer. The manual heuristic features are as follows:

- **Top-1 Probability**: The predicted probability of the most likely token:
$$\text{Top-1 Prob} = \max p_\theta(\cdot \mid x_{t,i}). \tag{5}$$
- **Probabilistic Margin**: The difference between the top-1 and top-2 probabilities, which effectively quantifies prediction confidence:
$$\text{Prob Margin} = \max p_\theta(\cdot \mid x_{t,i}) - \overset{(2)}{\max} p_\theta(\cdot \mid x_{t,i}), \tag{6}$$
where $\max^{(2)}$ denotes the second-highest probability.
- **Valid Attention Score**: We extract attention scores from three representative layers, including layer 0 (bottom), $\lfloor l/2 \rfloor$ (middle), and $l-1$ (top), where l is the total number of layers in the dLLM backbone. This choice is motivated by prior findings that different layers capture distinct levels of syntactic and semantic information Vig (2019); Jawahar et al. (2019).

Additionally, inspired by the success of Block-AR methods, we incorporate positional information. We include the relative position $\frac{i}{L}$ and the relative position among currently masked tokens:

$$\text{MASK-Pos} = \frac{\sum_{j \in \mathcal{M}_t} \mathbf{1}\{j < i\}}{|\mathcal{M}_t|}. \tag{7}$$

These features together form a 7-dimensional vector $\kappa_{t,i} \in [0,1]^7$ for each token at step t (1 for top-1 probability, 1 for margin, 3 for attention from different layers, 1 for relative position, and 1 for mask position). In our design, all features have the same range, which is beneficial for training. While this design results from careful empirical selection, future work may explore additional heuristics such as entropy or alternative attention-based signals Guo et al. (2024).

3.2 NETWORK ARCHITECTURE

The architecture of our neural scorer is illustrated in Fig. 3. It employs a dual-path design: a linear skip-connection path (blue) and a two-layer Mean-Field Multilayer Perceptron (MF-MLP) path (red). The final score is computed as:

$$s_{t,i} = \text{Linear}(\kappa_{t,i}) + \text{MF-MLP}(\kappa_{t,i}; \kappa_t). \tag{8}$$

This design is motivated by two key observations from our analysis. First, since the optimal denoising strategy is context-dependent, the scorer must incorporate global contextual information. However, full Transformer-based scorers Hong et al. (2026); Jazbec et al. (2026) introduce too many parameters for effective zero-order optimization. Inspired by Mean-Field Reinforcement Learning (MFRL) Yang et al. (2018), we propose a lightweight MF-MLP that uses the mean embedding of the first-layer hidden representations as contextual input to the second layer. We apply the mean-field operation only after the first layer to minimize parameter count. Second, a purely non-linear

architecture can slow down early-stage optimization. Since individual heuristics already perform reasonably well in many cases, we add a linear bypass path to enable fast initial progress via simple weighted combinations.

Overall, this design reflects our goal of achieving a compact yet effective structure. To capture the mean-field contextual information, a single layer is insufficient, as an affine combination would be identical for all tokens. Thus, we adopt a two-layer MF-MLP, with the mean-field embedding introduced only at the second-layer input to minimize the parameter overhead. Additionally, the linear bypass layer facilitates efficient training under zero-order optimization conditions.

We now formalize the MF-MLP structure. First, a linear layer with ReLU activation and layer normalization produces per-token embeddings:

$$y_{t,i} = \mathrm{Linear}(\kappa_{t,i}) \in \mathbb{R}^h, \tag{9}$$

where h is the hidden dimension. The mean-field embedding over all masked tokens is then computed:

$$\overline{y}_t = \frac{1}{|\mathcal{M}_t|} \sum_{j \in \mathcal{M}_t} y_{t,j} \in \mathbb{R}^h. \tag{10}$$

Finally, the second linear layer produces the contextual score:

$$z_{t,i} = \mathrm{Linear}([y_{t,i}; \overline{y}_t]) \in \mathbb{R}. \tag{11}$$

The final score $s_{t,i}$ is the sum of $z_{t,i}$ and the affine bypass term.

3.3 Training Process

We employ CMA-ES with task-specific adaptations for training. The fitness function is defined as the average accuracy over a mini-batch:

$$f(\Theta) = \frac{1}{B} \sum_{q \in \mathcal{B}} \mathbf{1}\{\mathrm{correct}(q; \mathrm{h}_\Theta)\}, \tag{12}$$

where $\mathcal{B}$ is a mini-batch of size B sampled from the training set. The batch is fixed within each generation but resampled across generations to reduce overfitting. The complete training procedure is summarized in Algorithm 1 at Appendix C. Since zero-order optimization itself is not the primary contribution of this work, we refer readers to Hansen & Ostermeier (2001); Hansen (2016) for detailed CMA-ES mechanics. The cost of this route is not zero: a full fit draws $100 \times 10 \times B$ backbone generations, all of them inference only, with the backbone frozen and no backward pass. The advantage over trainable schedulers is therefore not that the search is free, but that it needs neither a differentiable surrogate, nor a reference policy, nor any label beyond the correctness signal that evaluation already produces.

4 Experiments

4.1 Experiment Setup

To validate the efficiency and generalization capacity of the proposed method, we evaluate our scheduler on two popular dLLMs, LLaDA-8B-Instruct Nie et al. (2026) and Dream-7B-Instruct Ye et al. (2025b), and on four reasoning and planning tasks spanning mathematics and logic: GSM8K Cobbe et al. (2021), Math Hendrycks et al. (2021), Countdown Pan et al. (2025), and StrategyQA Geva et al. (2021). For all datasets, we follow the standard training and test splits, and for Math we use the Math-500 subset as the test set for efficient evaluation. Detailed introductions to these datasets are provided in Appendix E.

In addition to conventional heuristic and Block-AR schedulers, we compare against four recent approaches: Early Decision Matters (EDM, 5M parameters) (Kim et al., 2026b), which trains a scorer to guide the initial denoising steps, and for which we adopt the original evaluation protocol and report the results from the original paper (Kim et al., 2026b) since its dataset construction is nontrivial; CCD (Chen et al., 2026b), which uses historical average confidence scores for more consistent guidance; the AGDO Scheduler (Deng et al., 2026), which first selects candidates based on last-layer

Table 1: Model Performance on LLaDA-8B-Instruct Nie et al. (2026). In the table, 'T' denotes the step budget, 'L' denotes the sequence length, 'B' denotes the block size, and 'Attn (u/m/l)' indicates that the attention score is computed from the upper, middle, or lower layer, respectively. **Bold** indicates the best performance and underline denotes the second best one. These conventions apply to all subsequent tables. The results reported for EDM are reproduced from the original paper Kim et al. (2026b).

Type	Scheduler	GSM8K				Math			
		L=128		L=256		L=128		L=256	
		T=16	T=32	T=16	T=32	T=16	T=32	T=16	T=32
Heuristics	Top-1 Prob	43.2	54.0	40.4	47.5	17.8	21.0	13.2	18.6
	Entropy	35.9	46.2	43.2	44.0	17.2	13.0	12.2	15.2
	Prob Margin	50.2	54.9	45.6	50.0	19.8	20.0	17.2	19.8
Block-AR (B=32)	Top-1 Prob	42.6	60.5	9.8	46.8	11.8	23.0	4.8	17.4
	Entropy	32.8	57.0	6.7	35.1	11.8	19.2	4.0	11.2
	Prob Margin	47.6	66.6	12.8	52.5	15.2	24.0	5.4	16.8
Recent Works	EDM (5M params)	–	56.8	–	–	–	22.8	–	–
	CCD	43.5	52.5	37.0	48.7	18.2	19.4	12.2	17.4
	AGDO Scheduler	43.4	54.4	40.3	47.4	17.6	21.4	13.2	19.0
	Suffix Anchor	54.9	56.7	45.7	50.9	18.0	21.8	16.8	19.2
Proposed	Attn (u)	32.5	53.5	9.6	33.2	5.8	18.0	6.4	11.8
	Attn (m)	38.2	59.1	40.2	50.8	10.2	17.8	12.4	17.2
	Attn (l)	23.6	54.2	5.6	30.5	4.6	17.0	2.6	7.2
	Evolution (393 params)	**58.5**	**67.6**	**52.6**	**60.8**	**21.4**	**27.2**	**25.0**	**27.8**

Type	Scheduler	Countdown				StrategyQA			
		L=128		L=256		L=128		L=256	
		T=16	T=32	T=16	T=32	T=16	T=32	T=16	T=32
Heuristics	Top-1 Prob	39.8	46.0	3.5	20.7	63.0	65.9	36.8	56.8
	Entropy	39.5	40.6	7.4	27.0	62.7	65.5	33.5	51.4
	Prob Margin	40.3	48.4	6.6	18.8	63.2	65.9	41.6	57.9
Block-AR (B=32)	Top-1 Prob	24.2	36.3	4.7	9.0	55.0	62.9	26.9	42.4
	Entropy	16.8	28.1	3.5	9.8	47.6	61.6	27.4	44.0
	Prob Margin	26.6	39.8	3.5	11.3	53.1	62.6	23.0	40.6
Recent Works	EDM (5M params)	–	43.8	–	–	–	–	–	–
	CCD	37.9	43.0	4.7	17.6	63.9	**66.4**	30.1	57.2
	AGDO Scheduler	40.6	45.7	5.5	20.7	62.3	65.6	37.3	53.6
	Suffix Anchor	40.6	42.2	33.2	**45.7**	57.6	60.4	**58.1**	59.1
Proposed	Attn (u)	17.2	21.5	1.6	9.8	44.8	55.7	22.7	31.7
	Attn (m)	14.8	25.4	6.6	7.8	55.6	61.9	25.6	49.3
	Attn (l)	13.3	23.0	0.4	3.9	57.9	59.7	28.9	48.9
	Evolution (393 params)	**50.0**	**53.9**	**35.2**	40.6	**65.5**	65.9	54.1	**66.5**

valid attention scores and then determines which tokens to denoise using Top-1 probability; and Suffix Anchor (Park et al., 2026), which appends a fixed prompt at the end of the generation region to mitigate EOS overflow. We single out AGDO (Deng et al., 2026) because it is the closest prior work to use an attention signal, and the two approaches differ in kind: AGDO fixes one attention term at one point in the pipeline, so its behaviour is set by design, whereas our scorer learns the weight of attention relative to the other features and the balance adapts as unmasking proceeds. Attention is thus informative but not self-sufficient. On LLaDA, AGDO exceeds plain probability margin in only 3 of 16 settings, with means of 36.8 against 38.8, while our learned scorer reaches 48.3 on the same grid. We exclude two neural schedulers from comparison, since Jazbec et al. (2026) adopts an adaptive denoising speed that is incompatible with our fixed-budget setting, and UPO (Hong et al., 2026) supports only one token per step, which is mismatched with our fast-denoising protocol. We evaluate under generation length $L \in \{128, 256\}$ and denoising budget $T \in \{16, 32\}$. Performance gaps between schedulers diminish with large budgets and pure AR decoding can be competitive Ben-Hamu et al. (2026); Jazbec et al. (2026); Luxembourg et al. (2026); Ni et al. (2026), so we focus on the more challenging fast-denoising regime, which better aligns with the parallel decoding strength of dLLMs. For CMA-ES we use a batch size of 100, a maximum of 10 generations, and an initial step size of 0.2, with all other parameters at pycma defaults Hansen et al. (2019). All experiments are conducted on a server with an Intel Xeon Platinum 8558P CPU and 8 NVIDIA H20 GPUs.

Table 2: Ablation and Generalization Experiment of LLaDA-8B-Instruct Nie et al. (2026) in Math Hendrycks et al. (2021)

Scheduler	L=128		L=256	
	T=16	T=32	T=16	T=32
Proposed	<u>21.4</u>	**27.2**	**25.0**	**27.8**
w/o MF	21.0	<u>25.8</u>	19.6	24.0
w/o skip	<u>21.4</u>	24.0	20.0	23.4
GSM8K	**22.6**	24.8	20.6	<u>24.6</u>
Countdown	14.0	18.0	20.4	15.8
StrategyQA	14.2	22.0	18.2	16.6
Dream-7B-Instruct	20.4	23.4	22.0	21.2
L=128	–	–	21.4	24.4
L=256	20.6	24.2	–	–
T=16	–	24.4	–	24.6
T=32	20.0	–	<u>22.6</u>	–

4.2 Experiment Results

The results on LLaDA-8B-Instruct are presented in Table 1, while those on Dream-7B-Instruct are deferred to Appendix F, as they lead to similar conclusions. Overall, our proposed evolutionary heuristic scheduler performs strongly across the board, outperforming all baselines in the majority of settings. The advantage of our method becomes particularly pronounced on more challenging tasks and under tighter budgets. For instance, on LLaDA at the most challenging MATH configuration ($T = 16$, $L = 256$, i.e., 16 tokens denoised per step), our scheduler achieves 25.0% accuracy against 17.2% for the best baseline. Countdown separates the schedulers most sharply: at $L = 256$ and $T = 16$ every heuristic and Block-AR baseline collapses below 10%, the strongest recent method holds at 33.2%, and our scheduler stays above 35%.

Moreover, we observe that attention-based heuristics exhibit inconsistent performance. Middle-layer attention scores generally outperform those from upper or lower layers, aligning with recent findings that the middle layers of Transformers serve as a "global workspace" for reasoning Gurnee et al. (2026). However, valid attention scores alone are far from sufficient, as they frequently underperform simple confidence-based heuristics. This observation reinforces our earlier claim (Fig. 2) that attention signals offer complementary rather than competitive value: they excel in certain niche scenarios but require combination with other heuristics to achieve robust performance.

Two settings deserve comment. On Countdown with $L = 256$ and $T = 32$, Suffix Anchor reaches 45.7 against our 40.6, and on StrategyQA with $L = 256$ and $T = 16$ it reaches 58.1 against 54.1. Suffix Anchor suppresses early [EOS] emission with a fixed prompt suffix, which is precisely the failure our analysis identifies, and these are the two windows where the [EOS] region has the most room to grow before the budget is exhausted. That a hand-coded structural edit can match a learned scheduler in the cells where the diagnosed failure has the most room to develop is consistent with the mechanism we describe, and it points to the natural complement to our method: a structural prior, rather than a larger network.

4.3 Ablation and Generalization Study

In this section, we take LLaDA-8B-Instruct Nie et al. (2026) and the Math dataset Hendrycks et al. (2021) as a case study to conduct ablation and generalization analyses. The core design of our method lies in the network architecture, so we evaluate the impact of removing the mean-field embedding and the skip linear layer, and we test generalization across source datasets (GSM8K, Countdown, and StrategyQA), backbone models (Dream-7B-Instruct), and inference settings (generation length L and denoising budget T). Since exhaustively evaluating all combinations would be computationally prohibitive, we select the most challenging dataset, Math, to make the conclusions more compelling and representative. A comparison between methods under higher decoding budgets is provided in Appendix F.

The experimental results are presented in Table 2. For the ablation study, removing either the mean-field embedding (w/o MF) or the skip linear layer (w/o skip) degrades performance to varying extents, and the degradation patterns differ across configurations. Eliminating the mean-field embedding hurts more when the sequence length is larger, aligning with our expectation that longer sequences require stronger global contextual aggregation. The skip linear layer supports zero-order optimization by reducing the difficulty of optimizing a purely non-linear structure.

Regarding the generalization experiments, transferring the scheduler to different datasets leads to varied degradation. The GSM8K-trained scheduler performs best among cross-dataset transfers, even slightly outperforming the Math-trained one under $T = 16, L = 128$, which is expected given the distributional similarity between the two math reasoning datasets. In contrast, schedulers trained on Countdown or StrategyQA suffer more substantial drops, suggesting that the scheduler learns domain-specific priors rather than purely universal heuristics. Nevertheless, since the scheduler inputs consist of effective manually designed heuristics, its performance remains at least comparable to those heuristic-based schedulers, avoiding complete failure even under domain shifts. When transferring to a different backbone (Dream-7B-Instruct) or unseen inference configurations (length and budget), we observe milder degradation, with results remaining highly competitive against prior work (Table 1). Notably, the degradation from cross-dataset shifts is consistently larger than that from cross-configuration shifts, indicating that the scheduler captures a combination of domain-specific scheduling priors and universal heuristics. This suggests that mixed-dataset training could be a promising direction to improve robustness, which we leave for future work.

5 Conclusion

In this paper, we analyze the failure modes of confidence-based heuristic schedulers in dLLMs. Through attention-based analysis, we reveal that EOS Overflow and Proximal Bias arise from excessive attention to invalid tokens such as [MASK] and [EOS]. Building on this insight, we show that valid attention scores serve as a strong complementary signal, yet no single heuristic suffices due to the highly context-dependent nature of optimal denoising trajectories. To address this challenge, we propose a lightweight evolutionary heuristic scheduler optimized via CMA-ES. By dynamically integrating multiple heuristics with a mean-field contextual embedding, our method achieves strong performance using only 393 trainable parameters, serving as the most parameter-efficient neural scheduler for dLLMs to date. Experiments across four challenging reasoning and planning tasks on LLaDA and Dream demonstrate gains over strong baselines in most settings, especially in difficult settings with limited denoising budgets. While the learned scheduler exhibits some domain specificity, this opens promising avenues for future work on mixed-dataset training and adaptive denoising speed designs to further improve generalization and efficiency.

AI Use Statement

In this work, we used generative AI tools for paper polishing, including grammar and wording checks, verification and support of mathematical derivations, and coding. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

Ethics Statement

This work adheres to the principles outlined in the ICLR Code of Ethics.

Reproducibility Statement

The code for this paper is available at `https://anonymous.4open.science/r/Evo-Denoiser-CE0D`.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Mikhail L Arbuzov, Sisong Bei, Ziwei Dong, Dmitri Kalaev, and Alexey A Shvets. Beyond exponential decay: Rethinking error accumulation in large language models. *arXiv preprint arXiv:2505.24187*, 2025.

Marianne Arriola, Aaron Gokaslan, Justin Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, volume 2025, pp. 50726–50753, 2025.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas S Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *Advances in Neural Information Processing Systems*, 38:55981–56007, 2026.

Mingyu Cao, Alvaro HC Correia, Christos Louizos, Shiwei Liu, and Lu Yin. Search or accelerate: Confidence-switched position beam search for diffusion language models. *International Conference on Machine Learning*, 2026.

Haolong Chen, Hanzhi Chen, Zijian Zhao, Kaifeng Han, Guangxu Zhu, Yichen Zhao, Ying Du, Wei Xu, and Qingjiang Shi. An overview of domain-specific foundation model: key technologies, applications and challenges. *Science China Information Sciences*, 69(1):111301, 2026a.

Kecheng Chen, Ziru Liu, Xijia Tao, Hui Liu, Xinyu Fu, Suiyun Zhang, Dandan Tu, Lingpeng Kong, Rui Liu, and Haoliang Li. Beyond confidence: Adaptive and coherent decoding for diffusion language models. *International Conference on Learning Representations*, 2026b.

Shi-Ting Chen, Zijian Zhao, and Jinsong Chen. Confident RAG: Enhancing the performance of LLMs for mathematics question answering through multi-embedding and confidence scoring. In *ICLR 2026 Workshop on Logical Reasoning of Large Language Models*, 2026c.

Tingwei Chen, Jiayi Chen, Zijian Zhao, Haolong Chen, Liang Zhang, and Guangxu Zhu. First token probability guided rag for telecom question answering. *arXiv preprint arXiv:2501.06468*, 2025.

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

Jia Deng, Junyi Li, Wayne Xin Zhao, Jinpeng Wang, Hongyu Lu, and Ji-Rong Wen. Beyond fully random masking: Attention-guided denoising and optimization for diffusion language models. *arXiv preprint arXiv:2606.12273*, 2026.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. Attention score is not all you need for token importance indicator in kv cache reduction: Value also matters. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21158–21166, 2024.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models, July 2026. URL `https://transformer-circuits.pub/2026/workspace/index.html`. Accessed: 2026.

Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. CMA-ES/pycma on Github. Zenodo, DOI:10.5281/zenodo.2559634, February 2019. URL https://doi.org/10.5281/zenodo.2559634.

Haoyu He, Katrin Renz, Yong Cao, and Andreas Geiger. Mdpo: Overcoming the training-inference divide of masked diffusion language models. *arXiv preprint arXiv:2508.13148*, 2025.

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

Chunsan Hong, Seonho An, Min-Soo Kim, and Jong Chul Ye. Improving discrete diffusion unmasking policies beyond explicit reference policies. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=on6cb46OhD.

Zemin Huang, Zhiyang Chen, Zijun Wang, Tiancheng Li, and Guo-Jun Qi. Reinforcing the diffusion chain of lateral thought with diffusion language models. *Advances in Neural Information Processing Systems*, 38:152677–152710, 2026a.

Zemin Huang, Yuhang Wang, Zhiyang Chen, and Guo-Jun Qi. Don't settle too early: Self-reflective remasking for diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=BsZeTuB5fD.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does bert learn about the structure of language? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 3651–3657, 2019.

Metod Jazbec, Theo X Olausson, Louis Béthune, Pierre Ablin, Michael Kirchhof, João Monteiro, Victor Turrisi, Jason Ramapuram, and Marco Cuturi. Learning unmasking policies for diffusion language models. *International Conference on Machine Learning*, 2026.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. *Advances in neural information processing systems*, 38:19720–19745, 2026.

BumJun Kim, Dongjae Jeon, Dueun Kim, Wonje Jeung, and Albert No. Rainbow padding: Mitigating early termination in instruction-tuned diffusion LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=cznTlh7Msz.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *International Conference on Machine Learning*, pp. 30749–30768. PMLR, 2025.

Jiyeon Kim, Sungik Choi, Yongrae Jo, Moontae Lee, and Minjoon Seo. Early decisions matter: Proximity bias and initial trajectory shaping in non-autoregressive diffusion language models. *International Conference on Machine Learning*, 2026b.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Guanxi Lu, Hao Mark Chen, Yuto Karashima, Zhican Wang, Daichi Fujiki, and Hongxiang Fan. Adablock-dLLM: Semantic-aware diffusion LLM inference via adaptive block size. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0Cv9PwL7cI.

Omer Luxembourg, Haim Permuter, and Eliya Nachmani. Plan for speed: Dilated scheduling for masked diffusion language models. *International Conference on Machine Learning*, 2026.

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

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. In *International Conference on Learning Representations*, volume 2025, pp. 82974–82997, 2025.

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. *Advances in Neural Information Processing Systems*, 38:50608–50646, 2026.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero. https://github.com/Jiayi-Pan/TinyZero, 2025. Accessed: 2025-01-24.

Jungwon Park, Jimyeong Kim, Jungmin Ko, Nojun Kwak, and Wonjong Rhee. When confidence misleads: Suffix anchoring and anchor-proximity confidence modulation for diffusion language models. *arXiv preprint arXiv:2605.28181*, 2026.

Julianna Piskorz, Cristina Pinneri, Alvaro Correia, Motasem Alfarra, Risheek Garrepalli, and Christos Louizos. Masks can be distracting: On context comprehension in diffusion language models. In *Workshop on Scientific Methods for Understanding Deep Learning*, 2026. URL https://openreview.net/forum?id=y6Nvum4WwO.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Luhan Tang, Longxuan Yu, Shaorong Zhang, and Greg Ver Steeg. Is your diffusion sampler actually correct? a sampler-centric evaluation of discrete diffusion language models. *International Conference on Machine Learning*, 2026.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Jesse Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 37–42, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-3007. URL https://www.aclweb.org/anthology/P19-3007.

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.

Shaoan Xie, Lingjing Kong, Xiangchen Song, Xinshuai Dong, Guangyi Chen, Eric Xing, and Kun Zhang. Advancing reasoning in diffusion language models with denoising process rewards. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42703–42720, 2026.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In *International conference on machine learning*, pp. 5571–5580. PMLR, 2018.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. In *International Conference on Learning Representations*, volume 2025, pp. 77875–77898, 2025a.

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*, 2025b.

Zebin You, Shen Nie, Xiaolu Zhang, JUN ZHOU, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10093–10105, 2026.

Yu Zhang, Xinchen Li, Jialei Zhou, Hongnan Ma, Zhongwei Wan, Yiwei Shi, Duoqian Miao, Qi Zhang, and Longbing Cao. Swordsman: Entropy-driven adaptive block partition for efficient diffusion language models. *International Conference on Machine Learning*, 2026.

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

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. In *International Conference on Learning Representations*, volume 2025, pp. 63186–63227, 2025.

Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, et al. Llada-moe: A sparse moe diffusion language model. *arXiv preprint arXiv:2509.24389*, 2025.

APPENDIX CONTENTS

A Background of dLLMs **15**

A.1 Diffusion Language Models 15

A.2 Denoising Scheduler Formulation 15

B Literature Review **15**

C Training Algorithm **16**

D Analytic Properties of the Scorer **17**

E Dataset Introduction **18**

F Expanded Experiment Results **18**

A Background of dLLMs

A.1 Diffusion Language Models

dLLMs extend diffusion modeling to discrete text generation, offering a compelling alternative to conventional AR LLMs. Unlike AR models that generate tokens sequentially from left to right, dLLMs begin with a fully masked sequence and iteratively denoise tokens in parallel using bidirectional attention. This design enables full-context modeling at every step and provides competitive performance, often with significantly faster inference for long sequences due to parallel decoding. Representative models such as LLaDA Nie et al. (2026) and Dream Ye et al. (2025b) have demonstrated strong results while benefiting from inherent parallelism and bidirectional reasoning.

The forward (noising) process gradually corrupts a clean sequence $\mathbf{x}^0$ by replacing tokens with the special [MASK] token. Let $\beta_t \in (0, 1)$ denote the instantaneous masking rate at time t. The marginal probability that a token remains unmasked at time t is $\alpha_t = \exp(-\int_0^t \beta_s ds)$, which decreases monotonically from 1 to 0, and the per-token transition kernel $q(x_i^t \mid x_i^{t-1})$ equals β_t when $x_i^{t-1} \neq$ [MASK] and 1 otherwise, so that the masked state is absorbing.

The reverse (denoising) process is parameterized by a Transformer network p_θ, which predicts the original tokens for masked positions. The model is trained by maximizing the ELBO on the data likelihood, which reduces to a weighted masked cross-entropy objective:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{t\sim\mathcal{U}(0,1),\, \mathbf{x}^0,\, \mathbf{x}^t\sim q}\left[\frac{|\dot{\alpha}_t|}{1-\alpha_t}\sum_{i:x_i^t=\text{[MASK]}} -\log p_\theta(x_i^0 \mid \mathbf{x}^t)\right], \tag{13}$$

where $\dot{\alpha}_t = d\alpha_t/dt$.

A.2 Denoising Scheduler Formulation

At inference, generation starts from a fully masked sequence $\mathbf{x}^T$ and progressively recovers tokens over multiple steps. The network p_θ estimates the clean data distribution $p_\theta(\mathbf{x}^0 \mid \mathbf{x}^t)$. Given a noisy sample $\mathbf{x}^t$, we first draw a predicted clean sequence $\tilde{\mathbf{x}}^0 \sim p_\theta(\cdot \mid \mathbf{x}^t)$, and then sample the previous state from the posterior:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathbb{E}_{\tilde{\mathbf{x}}^0\sim p_\theta(\cdot|\mathbf{x}^t)}\Big[q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \tilde{\mathbf{x}}^0)\Big]. \tag{14}$$

However, computing $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ directly is computationally intractable due to the expectation over the vocabulary space. In practice, we mostly rely on a denoising scheduler to approximate this reverse step. Importantly, the inference-time denoising schedule (e.g., the number of iterations and the re-masking strategy) is not explicitly supervised during training, as the ELBO objective jointly supervises all time steps. Recent studies Tang et al. (2026) have shown that the choice of this schedule significantly affects generation quality, making the design of effective denoising schedulers a critical challenge for dLLMs. A detailed related works review is provided at Appendix B.

B Literature Review

Early work, inspired by the concept of confidence in AR LLMs Kang et al. (2026), designs denoising schedulers under the assumption that tokens with higher confidence are more likely to be correctly predicted, and that revealing them first can provide reliable guidance for subsequent tokens. Metrics such as top-1 probability Wu et al. (2025), entropy Ben-Hamu et al. (2026), and margin probability Kim et al. (2025) have been introduced and shown to outperform random denoising Zheng et al. (2025). Building on these, subsequent methods propose further empirical improvements. For instance, (Chen et al., 2026b) observed inconsistency in individual token predictions and proposed CCD, which uses historical average confidence as a more stable denoising metric. (Cao et al., 2026) noted that average confidence over the entire denoising process correlates with final accuracy and introduced PBS, which applies beam search to retain trajectories with the highest cumulative confidence. To address the EOS Overflow issue, (Nie et al., 2026) and (Park et al., 2026) proposed directly suppressing [EOS] tokens or down-weighting confidence near sentence endings. However, these approaches may result in overly long outputs and lack flexibility.

To prevent [EOS] tokens from appearing too early and interfering with generation, Block-AR has emerged as an alternative paradigm Arriola et al. (2025). By allowing arbitrary diffusion within blocks while decoding blocks auto-regressively, Block-AR achieves better contextual consistency, yet introduces new challenges. For example, (Lu et al., 2026) and (Zhang et al., 2026) observed that block size is a critical hyper-parameter affecting generation quality, and more importantly, splitting a sentence or semantic paragraph across blocks can cause significant performance degradation. To address this, (Lu et al., 2026) and (Zhang et al., 2026) proposed AdaBlock and Swordsman, respectively, which adaptively determine block boundaries based on confidence-related metrics, following empirical heuristics. Moreover, (Kim et al., 2026b; Ye et al., 2025a) also argue that reintroducing AR constraints in dLLMs could limit their potential for complex reasoning and planning tasks.

As noted by (Chen et al., 2026b), most of the above methods rely on empirical heuristics with limited theoretical guarantees. As a result, another line of work aims to train a neural scheduler jointly with the dLLM backbone to learn optimal denoising trajectories by RL or from offline data. For instance, (Huang et al., 2026a) proposed LLaDOU, which first formulates denoising as a Markov Decision Process (MDP) and employs the Plackett–Luce model to model token selection probabilities at each denoising step. They further introduced RemeDi, incorporating a pre-training warmup with a remasking mechanism Huang et al. (2026b) to improve performance. To reduce training costs, recent efforts focus on training the scheduler while keeping the dLLM backbone frozen, which also achieves competitive results. For example, (Hong et al., 2026) and (Jazbec et al., 2026) both adopt GRPO Shao et al. (2024) to train a Transformer-based scheduler, taking as input either the hidden features from the dLLM or the prediction probabilities of each token, respectively. Meanwhile, (Kim et al., 2026b) train a Transformer to predict whether an initial trajectory is likely to lead to a correct answer using offline data, and then use it to score and guide early inference steps. Overall, these methods require Transformers with parameter counts ranging from 300K to 134M, incurring additional training and inference costs. In contrast, our proposed scheduler is a carefully designed network with only 393 parameters, making it, to the best of our knowledge, the most lightweight neural scheduler to date.

C Training Algorithm

The detailed training process of our neural scorer is provided at Algorithm 1.

Algorithm 1 CMA-ES Training for the Neural Scorer

Require: Model $\mathrm{h}_\Theta(\cdot)$ with parameters $\Theta \in \mathbb{R}^D$; training set $\mathcal{D}_{\text{train}}$; population size λ; initial step size σ_0; batch size B; maximum generations $G_{\max}$.
Ensure: Optimized parameters Θ^*.

Initialize Θ_0 randomly (e.g., Xavier uniform).
Initialize CMA-ES parameters: mean $\mathbf{m} \leftarrow \Theta_0$, step size $\sigma \leftarrow \sigma_0$, covariance $\mathbf{C} \leftarrow \mathbf{I}$.
$g \leftarrow 0$.
while $g < G_{\max}$ and not converged **do**
    Sample λ candidate solutions $\{\Theta_i\}_{i=1}^{\lambda} \sim \mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$.
    Sample a fixed mini-batch $\mathcal{B} \subset \mathcal{D}_{\text{train}}$ of size B.
    for $i = 1$ to λ **do**
        Assign Θ_i to scorer h_{Θ_i}.
        Generate answers for all questions in $\mathcal{B}$ using h_{Θ_i}.
        Compute fitness $f_i \leftarrow \frac{1}{B}\sum_{q\in\mathcal{B}} \mathbf{1}\{\text{correct}(q)\}$.
    end for
    Sort candidates by fitness: $f_{(1)} \geq \cdots \geq f_{(\lambda)}$.
    Select top μ candidates ($\mu = \lambda/2$).
    Update mean: $\mathbf{m} \leftarrow \sum_{i=1}^{\mu} w_i \Theta_{(i)}$, with weights $w_i = \frac{\log(\mu+1)-\log(i)}{\sum_{j=1}^{\mu}(\log(\mu+1)-\log(j))}$.
    Update covariance matrix $\mathbf{C}$ and step size σ using standard CMA-ES rules.
    $g \leftarrow g + 1$.
end while
return $\Theta^* \leftarrow \mathbf{m}$.

D Analytic Properties of the Scorer

This appendix collects three short results on what the scorer parameterization can express, together with the notation they need. Throughout, $\kappa_{t,i} \in [0,1]^d$ denotes the feature vector of position i at step t, $\mathcal{M}_t$ the set of masked positions, and $A(\cdot)$ the accuracy a scheduler attains on a fixed evaluation set. We write $h_j(\kappa_{t,i}) = \kappa_{t,j}$ for the scheduler that ranks positions by the j-th feature alone.

P1: the linear path contains every single heuristic. For a fixed j, set $w = e_j$, the j-th standard basis vector, and $b = 0$, so that $w^\top \kappa_{t,i} = \kappa_{t,j} = h_j(\kappa_{t,i})$ at every position. The linear path then induces the same ranking as feature j at every step, and since the top-k selection in Eq. (2) is a function of that ranking alone, the two schedulers unmask the same positions in the same order and therefore attain the same accuracy on any evaluation set. Writing $\mathcal{W}$ for the linear-path weight space, this gives $\sup_{w \in \mathcal{W}} A(w) \geq \max_j A(e_j)$. Every heuristic we consider is therefore a feasible point of the search rather than a competitor it has to displace, and the contextual term enlarges the family from there.

P2: depth is what lets the context reach the score. Let $\bar{\kappa}_t = \frac{1}{|\mathcal{M}_t|} \sum_{j \in \mathcal{M}_t} \kappa_{t,j}$ be any permutation-invariant summary of the masked positions' features, and consider a scorer that reads the context through a single affine map, $s_{t,i} = w^\top [\kappa_{t,i}; \bar{\kappa}_t] + b$. Expanding gives $s_{t,i} = w_\kappa^\top \kappa_{t,i} + w_{\bar{\kappa}}^\top \bar{\kappa}_t + b$, in which the last two terms carry no index i and act as a common additive shift c_t. Since adding a constant to every score preserves their relative order, we have $\mathrm{TopK}_i(\tilde{s}_{t,i} + c_t) = \mathrm{TopK}_i(\tilde{s}_{t,i})$ for every value of $\bar{\kappa}_t$ and every k.

This identity is what fixes the architecture. The pooled summary is a function of the whole masked set, so a scorer can exploit it only if it passes through a map that is nonlinear in the context, and the two-layer scorer does exactly that: the second layer reads $[y_{t,i}; \bar{y}_t]$, where $\bar{y}_t$ depends on $\bar{\kappa}_t$ through the nonlinearity of the first layer. The shift that cancels in the top-k selection is then the constant part of that map, while the state dependence enters in the varying part, which is how one fixed parameter vector produces a comparison direction that adapts across steps. Reading the mean field through a single affine layer would annihilate the term entirely, so the depth of the scorer is not incidental to carrying the context, it is the mechanism by which the context becomes usable.

P3: mean-field approximation of the context. Let $Y_t \in \mathbb{R}^{m \times h}$ stack the first-layer embeddings $y_{t,j}$ of the masked positions, let $\bar{y}_t$ be their mean, and let Φ be permutation-invariant and L_Φ-Lipschitz with respect to the Frobenius norm on the stacked matrix. Then

$$\big\|\Phi(Y_t) - \Phi(\bar{y}_t \mathbf{1}^\top)\big\|_2 \;\leq\; L_\Phi \Big(\sum_{j \in \mathcal{M}_t} \|y_{t,j} - \bar{y}_t\|_2^2 \Big)^{1/2} \;\leq\; L_\Phi \sqrt{|\mathcal{M}_t|}\, \max_{j \in \mathcal{M}_t} \|y_{t,j} - \bar{y}_t\|_2 .$$

Let $\bar{Y} = \bar{y}_t \mathbf{1}^\top$ denote the matrix whose rows all equal the mean. The Lipschitz property gives $\|\Phi(Y_t) - \Phi(\bar{Y})\|_2 \leq L_\Phi \|Y_t - \bar{Y}\|_F$, and row j of $Y_t - \bar{Y}$ is $y_{t,j} - \bar{y}_t$, so the Frobenius norm is $\big(\sum_j \|y_{t,j} - \bar{y}_t\|_2^2\big)^{1/2}$, which is at most $\sqrt{|\mathcal{M}_t|} \max_j \|y_{t,j} - \bar{y}_t\|_2$. The first inequality becomes an equality only for functionals and inputs that attain the Lipschitz bound, while the second is tight when all deviations have equal norm.

The bound is tight when the masked positions' embeddings coincide, in which case the mean is a lossless summary of the context, and it grows with their dispersion. Two consequences follow. Layer normalization precedes the pooling because it controls the scale of the per-token representations the bound depends on, which makes it structural rather than stylistic. And the dispersion term grows with the size and heterogeneity of the masked set, so the mean field carries the most information when the sequence is long enough for that dispersion to accumulate. The ablation in Table 2 is consistent with this reading: removing the mean-field embedding costs 0.9 points on average at $L = 128$ and 4.6 at $L = 256$, whereas removing the skip layer does not produce a cost that grows with sequence length in the same way.

E Dataset Introduction

- **GSM8K Cobbe et al. (2021):** A widely used benchmark of grade-school math word problems that require multi-step arithmetic reasoning. Each problem is paired with a natural-language solution, making it a standard testbed for evaluating chain-of-thought reasoning.
- **Math Hendrycks et al. (2021):** A challenging collection of competition-level mathematics problems covering topics such as algebra, geometry, and number theory. For efficient evaluation, we adopt the Math-500 subset as the test set.
- **Countdown Pan et al. (2025):** An arithmetic planning task in which the model must combine a given set of numbers with basic operations to reach a target value. Following Kim et al. (2026b), we adopt the same 1-shot prompting strategy to ensure valid outputs and fair comparison.
- **StrategyQA Geva et al. (2021):** A logical reasoning benchmark of open-domain yes/no questions that require implicit multi-hop reasoning, where the necessary reasoning steps are not explicitly stated and must be inferred by the model.

F Expanded Experiment Results

The experimental results for Dream are presented in Table 3 and lead to conclusions consistent with our analysis in Section 4.2. Furthermore, the results for LLaDA on Math under different inference budgets are shown in Fig. 4, where our proposed method consistently outperforms others across all settings.

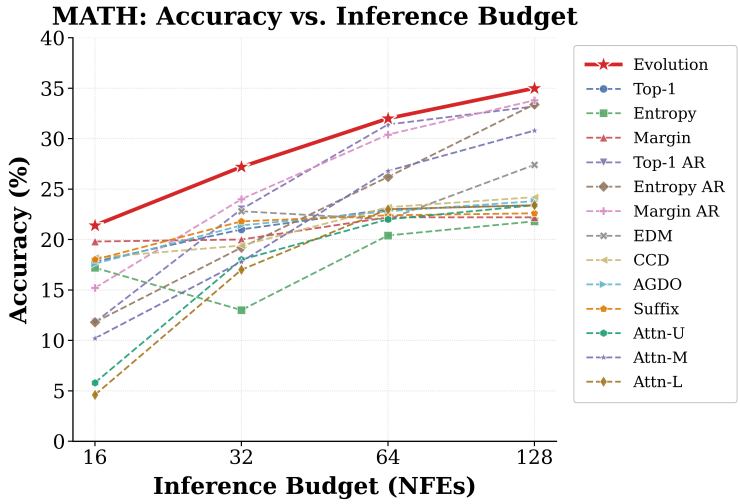


Figure 4: Performance comparison across inference budgets: our method consistently outperforms baselines on Math500 using LLaDA ($L = 128$).

Table 3: Model Performance on Dream-7B-Instruct Ye et al. (2025b). For Dream, which is based on an AR LLM backbone, the attention scores are shifted by one position to align with the token prediction order. Regarding EDM, the original paper does not report results for Countdown or StrategyQA on Dream.

Type	Scheduler	GSM8K				Math			
		L=128		L=256		L=128		L=256	
		T=16	T=32	T=16	T=32	T=16	T=32	T=16	T=32
Heuristics	Top-1 Prob	37.4	42.8	22.0	45.8	15.8	19.4	10.8	18.2
	Entropy	26.2	38.9	16.1	39.7	13.4	19.0	5.8	17.8
	Prob Margin	38.3	45.4	26.8	44.6	14.6	16.8	11.8	17.6
Block-AR (B=32)	Top-1 Prob	23.0	56.6	2.1	16.5	7.8	23.8	1.0	5.0
	Entropy	16.8	50.9	2.2	10.1	8.2	19.4	1.8	6.8
	Prob Margin	28.8	<u>58.8</u>	2.0	20.8	11.6	<u>24.4</u>	2.2	6.8
Recent Works	EDM (5M params)	–	–	–	<u>52.4</u>	–	–	–	<u>21.4</u>
	CCD	32.3	42.8	18.6	38.0	13.6	21.4	6.8	17.4
	AGDO Scheduler	36.5	41.6	20.2	39.9	16.0	19.8	11.0	17.6
	Suffix Anchor	<u>42.3</u>	46.4	<u>28.4</u>	37.0	<u>19.4</u>	22.2	**17.0**	20.2
Proposed	Attn (u)	16.7	48.1	1.6	12.6	4.6	19.4	2.0	6.8
	Attn (m)	37.4	42.8	30.7	41.3	11.7	15.4	12.0	14.8
	Attn (l)	9.5	46.0	2.0	5.1	2.4	14.4	2.0	2.0
	Evolution (393 params)	**43.5**	**58.9**	**37.5**	**60.3**	**27.2**	**27.0**	<u>15.2</u>	**22.6**

Type	Scheduler	Countdown				StrategyQA			
		L=128		L=256		L=128		L=256	
		T=16	T=32	T=16	T=32	T=16	T=32	T=16	T=32
Heuristics	Top-1 Prob	31.3	36.7	16.0	<u>31.6</u>	46.9	67.4	33.6	45.3
	Entropy	22.7	34.0	<u>18.4</u>	29.7	44.0	60.3	26.2	42.6
	Prob Margin	<u>35.2</u>	37.1	12.5	29.7	51.1	67.8	40.6	51.7
Block-AR (B=32)	Top-1 Prob	30.5	29.3	0.0	17.2	49.1	68.5	14.4	43.5
	Entropy	18.4	36.7	0.0	8.2	44.8	58.2	25.9	44.1
	Prob Margin	27.0	30.5	0.0	21.9	51.1	**69.1**	11.1	44.5
Recent Works	CCD	25.4	34.4	12.1	25.6	42.1	62.0	29.0	42.4
	AGDO Scheduler	29.3	<u>40.2</u>	<u>18.4</u>	28.5	45.9	58.7	34.2	46.0
	Suffix Anchor	14.5	33.6	2.7	10.9	<u>61.7</u>	64.5	53.7	61.3
Proposed	Attn (u)	5.5	17.2	0.4	2.34	50.9	58.9	29.5	53.9
	Attn (m)	10.9	15.2	0.0	4.3	58.1	60.1	<u>61.4</u>	<u>62.2</u>
	Attn (l)	3.9	34.0	5.5	8.7	22.7	22.9	23.6	26.6
	Evolution (393 params)	**47.3**	**48.4**	**19.9**	**33.6**	**63.8**	<u>68.6</u>	**65.6**	**65.1**