PREFIX-DENOISING CONSISTENCY: TEST-TIME VERIFICATION FOR DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion Language Models (DLMs) have recently become increasingly competitive with autoregressive (AR) models, and even outperform them on certain tasks. Unlike AR models, DLMs produce output through iterative denoising without a left-to-right order. To further improve the performance of DLMs, we introduce PDC (*Prefix-Denoising Consistency*), a test-time self-verification method for DLMs. PDC exploits a distinctive test-time signal in DLMs under prefix conditioned regeneration, correct trajectories are more stable and reproducible than incorrect ones. Concretely, given an initially generated sample, PDC splits the sentence at an intermediate position and regenerates the remaining tokens conditioned on the fixed prefix. Across mathematical reasoning and commonsense reasoning benchmarks, PDC consistently improves upon the initial sample, outperforms independent generations under a computational constrained comparison, and is robust to different unmasking strategies and parameter settings. These results highlight prefix-conditioned regeneration as an effective DLM-specific primitive for test-time verification.

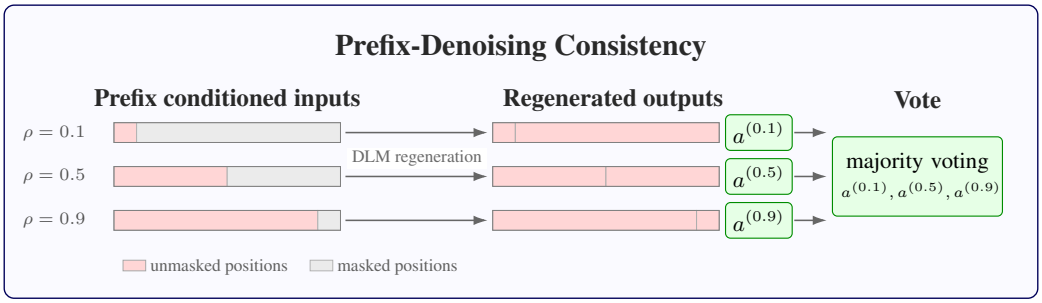


Figure 1: Overview of Prefix-Denoising Consistency (PDC). Given an initial Diffusion Language Model output, PDC keeps exact prefixes at rates $\rho \in \{0.1, 0.5, 0.9\}$, remasks the remaining positions within the same length-L output, and regenerates them to obtain candidate answers. The final answer is selected by majority vote over the regenerated answers only; the initial answer is recorded for analysis but excluded from the vote.

1 INTRODUCTION

Diffusion Language Models (DLMs) (Nie et al., 2025; Zhu et al., 2025; Ye et al., 2025) have recently emerged as a compelling alternative to autoregressive (AR) Language Models (OpenAI, 2025; Qwen Team, 2025; NVIDIA, 2025). Unlike AR models, which generate tokens strictly from left to right, DLMs refine outputs through iterative denoising, enabling more global revision and potentially more parallel and efficient inference. Recent DLMs have shown competitive performance across a range of language and reasoning tasks (Gong et al., 2025; Fu et al., 2026), suggesting that diffusion-based generation can offer a promising new scaling direction beyond the standard AR paradigm.

Compared to AR models, state-of-the-art DLMs are decoded at low temperature to obtain strong pass@1 accuracy. *However, this also creates a challenge for test-time verification: due to the low temperature, repeated samples often have limited diversity, because the DLM can follow similar*

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.

denoising trajectories and return the same answer. This makes a naive transfer of self-consistency from AR models less direct. In AR reasoning, self-consistency improves accuracy by sampling multiple reasoning paths and selecting the most frequent answer (Wang et al., 2023). For low-temperature DLM decoding, however, repeated full generations can result in near-identical denoising trajectories, so agreement among samples may overstate reliability rather than provide an independent check. In recent work, TIF (Wang et al., 2026a) addresses this issue by using temporal information inside the unmasking process. TIF votes over answers extracted from the information of intermediate denoising steps, showing that the denoising trajectory contains a useful reasoning signal and can improve accuracy. However, TIF exploits consistency within a single denoising trajectory: it reuses intermediate states from one run rather than generating multiple alternative reasoning trajectories. By construction, it therefore cannot directly leverage the multi-trajectory self-consistency mechanism that makes majority voting effective in AR reasoning. TIF can exploit temporal fluctuations along that run, but it cannot branch into a new trajectory once the run becomes locked into an erroneous reasoning path.

To address this issue, we propose PDC (*Prefix-Denoising Consistency*). Starting from a completed output, it keeps an exact prefix, remasks the remaining positions, and re-denoises them at the same temperature. Repeating this intervention at several keep rates (i.e., amount of the prefix kept) produces structured alternative trajectories, whose extracted answers are aggregated by majority vote. This allows the model to revise errors in the original generation while preserving useful context from the initial solution.

Our contributions are:

- **Prefix-conditioned regeneration.** PDC is a test-time scaling method that holds exact prefixes inside a fixed output window, remasks the unkept positions, and votes only over regenerated answers. The initial answer is excluded to isolate the effect of prefix-conditioned denoising.
- **Initially correct answers are more reproducible.** Figure 2 shows that initially correct answers are more reproducible than initially wrong answers. This observation leads to our algorithm.
- **Improved performance over the initial sample.** In the full diffusion setting across math and commonsense benchmarks, PDC consistently improves over the initial sample and TIF.
- **Compute efficiency and robustness analysis.** We compare PDC to standard self-consistency (i.e., majority voting). We show that PDC outperforms majority voting with a smaller total denoising-step budget. We also conduct validation under different unmasking schemes and settings and observe improvements.

2 PREFIX-DENOISING CONSISTENCY

2.1 STANDARD DLM INFERENCE TIME

DLMs generate outputs by iteratively denoising a fixed-length sequence initialized with mask tokens. Given a prompt x, a maximum output length L, and a total number of denoising steps T, let $\mathcal{V}^{\leq L}$ denote the set of token sequences over the vocabulary $\mathcal{V}$ with length at most L.

The denoising process begins with an output window consisting entirely of mask tokens:

$$z^{(0)} = \left[x,\ \underbrace{\texttt{[MASK]}, \ldots, \texttt{[MASK]}}_{L} \right], \tag{1}$$

At each denoising step $t = 1, \ldots, T$, the DLM predicts tokens for all currently masked output positions in parallel, conditioned on the partially denoised sequence $z^{(t-1)}$. It then selects a subset of these positions to unmask according to a denoising schedule, often based on the model's confidence (Nie et al., 2025; Zhu et al., 2025; Ye et al., 2025). The remaining positions stay masked, producing an updated sequence $z^{(t)}$, for example,

$$z^{(t)} = \left[x,\ \underbrace{\texttt{[MASK]}, \texttt{[TOKEN]}, \ldots, \texttt{[TOKEN]}, \texttt{[MASK]}, \ldots}_{L} \right].$$

Algorithm 1: **Prefix-Denoising Consistency (PDC)**

Require: Prompt x, generation budget L, denoising budget T, keep rates $\mathcal{R} = \{0.1, 0.5, 0.9\}$, answer extractor $\mathrm{Extract}$

Ensure: Regeneration-vote answer $\hat{a}$, or no-vote

$y \leftarrow \mathrm{DLM}(x; L, T)$; $\ell_y \leftarrow |y|$
Record $a^{(0)} \leftarrow \mathrm{Extract}(y)$ for diagnostics only
for each keep rate $\rho \in \mathcal{R}$ **do**
  if $\ell_y = 0$ **then**
    $L_\rho^{\text{prefix}} \leftarrow 0$
  else
    $L_\rho^{\text{prefix}} \leftarrow \max(1, \lfloor \rho \ell_y \rfloor)$
  end if
  $L_\rho \leftarrow L - L_\rho^{\text{prefix}}$
  $T_\rho \leftarrow L_\rho$
  Run the constrained DLM with output positions $1{:}L_\rho^{\text{prefix}}$ fixed and L_ρ suffix positions masked: $y^{(\rho)} \leftarrow \mathrm{DLM}\left(x, y_{1:L_\rho^{\text{prefix}}}; L_\rho, T_\rho\right)$
  $a^{(\rho)} \leftarrow \mathrm{Extract}(y^{(\rho)})$
end for
$\mathcal{B} \leftarrow \{a^{(\rho)} : \rho \in \mathcal{R},\ a^{(\rho)} \neq \varnothing\}$
return $\mathrm{Maj}(\mathcal{B})$, breaking ties by the fixed order $\rho = 0.1, 0.5, 0.9$

Thus, at an intermediate denoising step, the output window contains both positions that remain masked and tokens filled in during earlier steps. The positions are not necessarily unmasked in left-to-right order; instead, they may be resolved in an arbitrary order determined by the denoising schedule. After T denoising steps, DLM obtains a fully unmasked output window, output sequence $z^{(T)} = [x, y]$, length of y is at most the output length L.

Our method does not rely on the internal form of the DLM update rule. We therefore treat the DLM as a black-box generator that, given an input x, an output budget L, and a number of denoising steps T, returns a sequence

$$y = \mathrm{DLM}(x; L, T), \qquad y \in \mathcal{V}^{\leq L}.$$

The output length and initial extracted answer are:

$$\ell_y = |y|, \qquad a^{(0)} = \mathrm{Extract}(y),$$

where $\mathrm{Extract}$ is a task-specific canonical answer extractor and $\mathrm{Extract}(y) = \varnothing$ denotes a null extraction that represents a failure to parse the answer.

2.2 Prefix-Denoising Consistency (PDC)

Our proposed method, PDC (Figure 1 and Algorithm 1) holds a prefix of the initial generated output y at a keep rate $\rho \in \mathcal{R}$, masks the remaining output positions, regenerates those positions using the same DLM, and then aggregates the regenerated answers by majority vote.

Throughout the paper, we use a fixed set of three rates $\mathcal{R} = \{0.1, 0.5, 0.9\}$, this set covers weak, intermediate, and strong prefix conditioning while requiring only three regenerations. Given an initial output y of length ℓ_y, the number of held tokens for rate ρ is:

$$L_\rho^{\text{prefix}} = \max(1, \lfloor \rho \ell_y \rfloor), \qquad \ell_y > 0.$$

The remaining regeneration length and denoising step are:

$$L_\rho = L - L_\rho^{\text{prefix}}, \qquad T_\rho = L_\rho.$$

Different from Eq. 1, the regeneration input is the following equation:

$$z_\rho^{(0)} = \left[x,\ y_{1:L_\rho^{\text{prefix}}},\ \underbrace{\texttt{[MASK]}, \ldots, \texttt{[MASK]}}_{L_\rho} \right],$$

where $y_{1:L_\rho^{\text{prefix}}}$ is kept output tokens with a rate of ρ and L_ρ is the number of masked suffix positions to denoise. Only the remaining L_ρ output positions are denoised:

$$y^{(\rho)} = \mathrm{DLM}\left(x, y_{1:L_\rho^{\text{prefix}}}; L_\rho, T_\rho\right), \qquad |y^{(\rho)}| \le L.$$

The regenerated answer is:

$$a^{(\rho)} = \mathrm{Extract}(y^{(\rho)}).$$

Thus, the final length of $y^{(\rho)}$ may differ from the initial output length ℓ_y. However, regeneration does not continue from the end of the initial output. Instead, it only refills the remasked positions within the same output window of length L.

Motivated by analogous observations for AR models (Iwase et al., 2026), we find that the correct reasoning paths are more reproducible under regeneration than incorrect ones in DLMs (Figure 2). The gold answer is denoted by $a^\star$. Reproduction of the correct answer measures whether regeneration preserves a correct initial answer, and is defined as:

$$R_\rho^+ = \Pr\left[a^{(\rho)} = a^\star \mid a^{(0)} = a^\star\right].$$

The same wrong preservation measures whether regeneration repeats the same wrong answer when the initial answer is wrong, are defined as:

$$R_\rho^- = \Pr\left[a^{(\rho)} = a^{(0)} \mid a^{(0)} \neq \varnothing,\ a^{(0)} \neq a^\star\right].$$

Observation 2.1 (Reproduction rates)**.** Across most evaluated settings, initially correct answers are more likely to be reproduced than initially incorrect answers are to repeat the same wrong answer:

$$R_\rho^+ \ge R_\rho^-.$$

Using Observation 2.1, PDC assesses the reliability of the initial answer. PDC votes only over regenerated answers, initial answer $a^{(0)}$ is excluded from the final vote. This isolates the effect of prefix-conditioned regeneration: the DLMs can improve over the initial answer only if the regenerated candidates support a better answer. The multiset of non-null regenerated answers is:

$$\mathcal{B} = \left\{a^{(\rho)} : \rho \in \mathcal{R},\ a^{(\rho)} \neq \varnothing\right\}$$

If $\mathcal{B} = \{\}$, the example is marked as no-vote. Otherwise, PDC returns the most frequent regenerated answer, denoted by $\mathrm{Maj}(\mathcal{B})$:

$$\hat{a} = \mathrm{Maj}(\mathcal{B}).$$

The three keep rates probe different neighborhoods around the same initial output. The keep rate of 0.1 imposes only weak conditioning on the original generation trajectory and allows most of the reasoning path to change. The keep rate of 0.9 largely preserves the answer while perturbing only the final portion of the output. The keep rate of 0.5 provides an intermediate between the two. Together, these rates evaluate answer stability under varying strengths of prefix conditioning.

When does PDC improve accuracy? To demonstrate the effectiveness of PDC, we consider a stylized setting with only two possible answers: the correct answer $a^\star$ and one incorrect answer b. The purpose of this analysis is to identify when taking a majority vote over three regenerated answers improves upon the initial answer.

First, consider three independent Bernoulli variables (i.e., variables that take either one or zero) whose probabilities of being one are p_1, p_2, p_3, respectively. The probability that at least two of them are one is

$$\begin{aligned} V_3(\boldsymbol{p}) &:= \Pr[\text{at least two of the three variables are one}] \\ &= p_1p_2(1-p_3) + p_1(1-p_2)p_3 + (1-p_1)p_2p_3 + p_1p_2p_3 \\ &= p_1p_2 + p_1p_3 + p_2p_3 - 2p_1p_2p_3, \end{aligned} \tag{2}$$

where $\boldsymbol{p} = (p_1, p_2, p_3)$.

When all three probabilities are equal to p, we write

$$V_3(p) := V_3(p, p, p) = 3p^2 - 2p^3.$$

Thus, V_3 is the probability that the majority outcome is one.

For regeneration j, define

$$R^+_{\rho_j} := \Pr\left[a^{(\rho_j)} = a^{(0)} \mid a^{(0)} = a^\star\right], \quad R^-_{\rho_j} := \Pr\left[a^{(\rho_j)} = a^{(0)} \mid a^{(0)} = b\right].$$

In words, $R^+_{\rho_j}$ is the correct-answer retention probability, while $R^-_{\rho_j}$ is the same-wrong-answer persistence probability.

Theorem 2.2 (Benefit of the reproduction gap)**.** *Assume a binary answer space $\mathcal{A} = \{a^\star, b\}$ with no null answers, and let*

$$\pi := \Pr[a^{(0)} = a^\star].$$

Suppose that, conditional on $a^{(0)}$, the three regenerated answers are independent. Define

$$\boldsymbol{R}^\pm := \left(R^\pm_{\rho_1}, R^\pm_{\rho_2}, R^\pm_{\rho_3}\right).$$

Then

$$\mathrm{Acc}(\mathrm{PDC}) = \pi V_3(\boldsymbol{R}^+) + (1-\pi)\left[1 - V_3(\boldsymbol{R}^-)\right]. \tag{3}$$

In the non-degenerate case, PDC *improves upon* INIT *if and only if*

$$\pi < \frac{1 - V_3(\boldsymbol{R}^-)}{2 - V_3(\boldsymbol{R}^-) - V_3(\boldsymbol{R}^+)}. \tag{4}$$

In particular, if $\pi = 1/2$, then

$$\mathrm{Acc}(\mathrm{PDC}) - \mathrm{Acc}(\textsc{Init}) = \frac{1}{2}\left[V_3(\boldsymbol{R}^+) - V_3(\boldsymbol{R}^-)\right]. \tag{5}$$

Consequently,

$$V_3(\boldsymbol{R}^+) > V_3(\boldsymbol{R}^-) \iff \mathrm{Acc}(\mathrm{PDC}) > \mathrm{Acc}(\textsc{Init}).$$

Moreover, under the same assumption $\pi = 1/2$, if all coordinates of R^+ and R^- lie in $(0, 1)$, then the coordinate-wise condition

$$R^+_{\rho_j} \geq R^-_{\rho_j}, \qquad j = 1, 2, 3,$$

with strict inequality for at least one j, implies $V_3(R^+) > V_3(R^-)$, and hence PDC strictly improves over the initial prediction.

Proof. We separate the analysis into two cases.

Case 1: the initial answer is correct. Suppose $a^{(0)} = a^\star$. Regeneration j reproduces the initial answer, which is also the correct answer, with probability $R^+_{\rho_j}$. By conditional independence, the probability that at least two of the three regenerated answers are correct is therefore

$$\Pr[\hat{a} = a^\star \mid a^{(0)} = a^\star] = V_3(\boldsymbol{R}^+).$$

Case 2: the initial answer is wrong. Suppose $a^{(0)} = b$. Because there are only two possible answers, a regeneration is correct exactly when it does not reproduce the initial answer.

The final majority vote is wrong exactly when at least two regenerations reproduce b. The probability of this event is $V_3(\boldsymbol{R}^-)$. Hence,

$$\Pr[\hat{a} = a^\star \mid a^{(0)} = b] = 1 - V_3(\boldsymbol{R}^-).$$

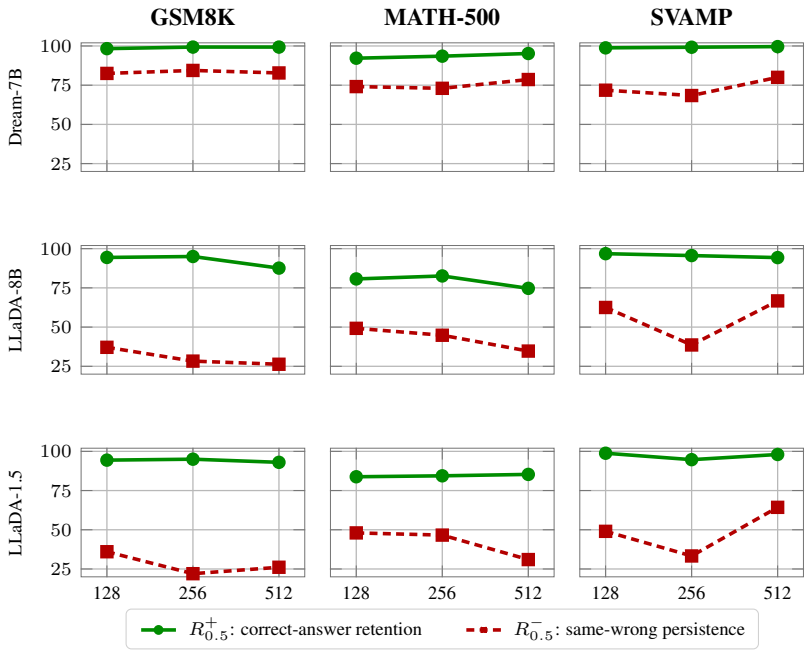


Figure 2: Reproduction rates under $\rho = 0.5$ regeneration. Green denotes correct-answer preservation $R^+_{0.5}$, while red denotes same-wrong-answer preservation $R^-_{0.5}$, across generation lengths $L = 128, 256, 512$ on three math benchmarks. The gap $R^+_{0.5} - R^-_{0.5} > 0$ shows that correct answers are more reproducible than wrong answers. Full results of reproduction rates, including CSQA and SQA, are reported in Appendix Table 4.

The initial answer is correct with probability π and wrong with probability $1 - \pi$. Combining the two cases gives

$$\mathrm{Acc}(\mathrm{PDC}) = \pi V_3(\boldsymbol{R}^+) + (1-\pi)\left[1 - V_3(\boldsymbol{R}^-)\right],$$

which proves Eq. 3.

Since the accuracy of the initial answer is $\mathrm{Acc}(\textsc{Init}) = \pi$, the change in accuracy is

$$\mathrm{Acc}(\mathrm{PDC}) - \mathrm{Acc}(\textsc{Init}) = (1-\pi)\left[1 - V_3(\boldsymbol{R}^-)\right] - \pi\left[1 - V_3(\boldsymbol{R}^+)\right]. \tag{6}$$

This expression has a direct interpretation:

$$\underbrace{(1-\pi)\left[1 - V_3(\boldsymbol{R}^-)\right]}_{\text{wrong initial answers that are corrected}} - \underbrace{\pi\left[1 - V_3(\boldsymbol{R}^+)\right]}_{\text{correct initial answers that are spoiled}}.$$

Thus, the method improves accuracy precisely when the benefit from correcting initially wrong answers exceeds the loss from spoiling initially correct answers. Rearranging $\mathrm{Acc}(\mathrm{PDC}) - \mathrm{Acc}(\textsc{Init}) > 0$ gives Eq. 4. Finally, suppose that $\pi = 1/2$. Then

$$\mathrm{Acc}(\mathrm{PDC}) - \mathrm{Acc}(\textsc{Init}) = \frac{1}{2}\left[V_3(\boldsymbol{R}^+) - V_3(\boldsymbol{R}^-)\right],$$

which gives Eq. 5 and the stated equivalence. It remains to establish the sufficient condition. From Eq. 2,

$$\frac{\partial V_3}{\partial p_1} = p_2 + p_3 - 2p_2p_3 = p_2(1-p_3) + p_3(1-p_2) > 0$$

for $p_2, p_3 \in (0,1)$. The same argument applies to the other two coordinates, so V_3 is strictly increasing in each coordinate on $(0,1)^3$. Therefore, if $R^+_{\rho_j} \geq R^-_{\rho_j}$ for every j, with strict inequality for at least one j, then

$$V_3(\boldsymbol{R}^+) > V_3(\boldsymbol{R}^-),$$

□

Table 1: For each pair, we select the generation length that yields the highest INIT accuracy; this selection is based only on INIT and is not optimized for PDC. PDC attains the best or tied-best accuracy in all 15 settings and improves over the selected INIT baseline in 14 of them, with gains of up to $+5.68$ accuracy points. The full results are reported in Appendix C. Signed green/red values denote the advantage of the method relative to the selected INIT baseline. Bold indicates the best accuracy in each row, and an underline indicates the second best accuracy, with ties marked together. For all benchmarks, examples for which the initial generation yields an empty output are excluded from evaluation. INIT accuracy is therefore computed over the remaining non-null examples.

Model	Dataset	Length	INIT	TIF Fixed	TIF Linear	TIF Exp.	PDC	Δ TIF avg.	Δ PDC
Dream-7B	GSM8K	512	82.03	81.96	82.03	82.03	**84.08**	-0.02	+2.05
	MATH-500	512	46.00	46.00	46.00	46.20	**47.00**	+0.07	+1.00
	SVAMP	256	87.00	87.00	87.00	87.00	**88.00**	+0.00	+1.00
	CSQA	128	73.22	73.79	73.96	73.55	**74.20**	+0.55	+0.98
	SQA	128	70.45	70.74	70.74	70.74	**71.76**	+0.29	+1.31
LLaDA-8B	GSM8K	256	60.73	60.12	60.80	60.80	**65.88**	-0.15	+5.15
	MATH-500	256	26.45	26.25	26.25	26.25	**27.86**	-0.20	+1.41
	SVAMP	128	83.67	83.67	83.67	83.67	**84.33**	+0.00	+0.66
	CSQA	128	80.51	80.51	80.51	80.51	**80.59**	+0.00	+0.08
	SQA	128	65.07	65.07	65.07	65.07	**66.67**	+0.00	+1.60
LLaDA-1.5	GSM8K	256	60.58	60.05	60.27	60.50	**66.26**	-0.30	+5.68
	MATH-500	256	25.65	25.65	25.45	25.65	**26.65**	-0.07	+1.00
	SVAMP	128	83.67	83.33	83.67	83.67	**84.67**	-0.11	+1.00
	CSQA	128	**80.10**	**80.10**	**80.10**	**80.10**	**80.10**	+0.00	+0.00
	SQA	128	66.08	66.08	66.08	66.08	**66.81**	+0.00	+0.73

To isolate the mechanism underlying the reproduction gap, we consider an idealized binary-answer setting with conditionally independent regenerations. This simplified setting yields an exact characterization of when PDC improves over INIT. In open-ended settings, R_ρ^- should be interpreted as same-wrong persistence rather than as a direct error-correction probability.

3 EXPERIMENTS

The experiments first examine the diagnostic signal behind PDC: whether prefix regeneration preserves initially correct answers more often than it repeats the same wrong answer. We then evaluate PDC against the initial sample and TIF (Wang et al., 2026a) baselines in full diffusion setting [1]. In the ablation study, we compare PDC with standard majority voting that spends more token budget than it. Furthermore, we conduct several robustness checks across alternative unmasking strategies.

Models and datasets. We study Dream-7B (Ye et al., 2025) (`Dream-org/Dream-v0-Instruct-7B`) and LLaDA-family models, LLaDA-8B-Instruct (Nie et al., 2025) (`GSAI-ML/LLaDA-8B-Instruct`) and LLaDA-1.5 (Zhu et al., 2025) (`GSAI-ML/LLaDA-1.5`). The main results use the full-diffusion setting. For LLaDA-family models, we set the block length equal to the generation length. All models are decoded at temperature 0.2. We evaluate on GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), SVAMP (Patel et al., 2021), CSQA (CommonsenseQA) (Talmor et al., 2019), and SQA (StrategyQA) (Geva et al., 2021).

3.1 CORRECT ANSWERS ARE MORE REPRODUCIBLE THAN WRONG ANSWERS

We first examine the reproduction rates that motivate PDC. Figure 2 shows reproduction rates at keep rate 0.5; the results for keep rates 0.1 or 0.9 across all datasets are reported in Appendix Table 4. The figure shows a significant gap between correct-answer retention and the same wrong answer

[1] In the LLaDA family, the block length controls the generation. When the block length is 1, the model reduces to an AR model; when the block length is L, it corresponds to the full diffusion setting; and when the block length is between 1 and L, it yields a semi-AR setting.

Table 2: We compare PDC with Self-Consistency under different total denoising-step budgets using the same generation length for each model and dataset pair as in Table 1. For both INIT and TIF, $\times 4$ denotes four independent initial denoising trajectories. INIT x4 votes over their final answers with budget $4T$, while PDC uses one initial generation followed by three regenerations at keep rates 0.1, 0.5, and 0.9. Among the 15 settings, PDC achieves the best or tied-best accuracy in 14 and improves over INIT x4 in 13.

Dataset	Model	INIT	INIT x4	TIF Fixed x4	TIF Linear x4	TIF Exp. x4	PDC	Δ vs. INIT x4
GSM8K	Dream-7B	82.03	82.03	81.96	82.03	82.03	**84.08**	+2.05
	LLaDA-1.5	60.58	60.73	60.27	60.50	60.65	**66.26**	+5.53
	LLaDA-8B	60.73	60.73	60.27	60.80	60.73	**65.88**	+5.15
MATH-500	Dream-7B	46.00	46.00	46.00	46.00	46.20	**47.00**	+1.00
	LLaDA-1.5	25.65	25.45	25.45	25.25	25.45	**26.65**	+1.20
	LLaDA-8B	26.45	26.45	26.25	26.45	26.45	**27.86**	+1.41
SVAMP	Dream-7B	87.00	87.00	87.00	87.00	87.00	**88.00**	+1.00
	LLaDA-1.5	83.67	83.67	83.33	83.67	83.67	**84.67**	+1.00
	LLaDA-8B	83.67	**84.33**	**84.33**	84.00	**84.33**	**84.33**	+0.00
CSQA	Dream-7B	73.22	73.22	73.79	73.96	73.55	**74.20**	+0.98
	LLaDA-1.5	80.10	80.51	**80.67**	**80.67**	**80.67**	80.10	-0.41
	LLaDA-8B	80.51	80.34	**80.59**	**80.59**	**80.59**	**80.59**	+0.25
SQA	Dream-7B	70.45	70.45	70.74	70.74	70.74	**71.76**	+1.31
	LLaDA-1.5	66.08	65.94	65.94	65.79	65.79	**66.81**	+0.87
	LLaDA-8B	65.07	64.63	65.07	65.07	65.07	**66.67**	+2.04

persistence: initially correct answers are typically reproduced under prefix-conditioned regeneration, whereas initially, wrong answers are less likely to be regenerated as the same wrong answer.

3.2 PDC IMPROVES THE INITIAL SAMPLE

We next examine whether PDC actually improves the accuracy of the answer. This is non-trivial: regenerations can move away from an initial wrong answer, but the new answers may still be wrong, and the final vote may still result in an incorrect answer.

Baselines. INIT indicates the initial answer y generated by the standard denoising process. PDC votes over 0.1, 0.5, and 0.9 keep-rate regenerations. Ties are broken by the fixed keep-rate order $0.1, 0.5, 0.9$. TIF, the state-of-the-art method for improving DLM accuracy, extracts answers from intermediate steps. To aggregate answers on these steps, TIF Linear and TIF Exp use linearly and exponentially increasing temporal weights, respectively. The exponential setting uses $\alpha = 5$, following the settings of temporal voting in DLMs (Wang et al., 2026a).

Results. Table 1 compares the accuracy of the methods. For each model and dataset pair, we optimize the generation length L to achieve the highest INIT accuracy. This compares PDC against the strongest available initial sample baseline for that pair, rather than against a favorable length chosen post hoc for PDC. Results across several values of L are reported in Appendix C.

On Dream-7B, PDC consistently improves over INIT on all five datasets. The advantage of PDC over INIT ranges from +0.98 to +2.05 points, with an unweighted mean gain of +1.27 points. On LLaDA-family models, PDC improves over INIT in nine of ten model–dataset pairs and ties in the remaining pair, with an unweighted mean gain of +1.73 points. Appendix D reports additional LLaDA-family runs with block length 32, a semi-AR setting. It shows that the same prefix-regeneration signal remains useful.

3.3 ABLATION STUDY

3.3.1 PDC OUTPERFORMS SELF-CONSISTENCY AT LOWER INFERENCE COST

We compare PDC with Self-Consistency using the same generation lengths as Table 1, selected for each model and dataset pair by the highest INIT accuracy. The INIT x4 baseline votes over four

Table 3: Robustness to alternative unmasking rules on MATH-500 across models and generation lengths. Dream uses origin unmasking, while the LLaDA-family models use random remasking. All rows use the full-diffusion setting with block length equal to generation length. We show answer accuracy for the initial generation, PDC with keep rates $\rho \in \{0.1, 0.5, 0.9\}$, and TIF variants. PDC achieves the best accuracy in all models and length settings.

Model	Length	Accuracy (%)					Reproduction rates (%)					
				TIF variants			$\rho = 0.1$		$\rho = 0.5$		$\rho = 0.9$	
		INIT	PDC	Fixed	Linear	Exp.	R_ρ^+	R_ρ^-	R_ρ^+	R_ρ^-	R_ρ^+	R_ρ^-
Dream-7B	128	17.60	**23.40**	17.60	17.80	17.60	51.14	12.66	64.77	17.53	95.45	64.61
	256	14.20	**20.40**	14.80	14.80	15.00	60.56	10.74	56.34	17.18	78.87	57.67
	512	14.80	**20.40**	15.20	15.20	15.00	54.05	14.78	54.05	19.81	83.78	58.18
LLaDA-1.5	128	27.20	**28.60**	27.80	28.20	27.20	68.38	21.25	73.53	36.26	94.85	81.30
	256	31.46	**33.07**	31.66	31.86	32.06	70.06	18.34	77.07	30.77	92.99	80.47
	512	31.45	**36.29**	31.25	32.66	32.86	71.79	11.68	73.08	23.65	90.38	73.35
LLaDA-8B	128	25.20	**28.00**	25.00	25.40	25.40	64.29	18.31	77.78	34.37	93.65	76.34
	256	31.06	**33.27**	29.46	30.86	31.66	69.03	22.71	76.77	33.92	94.19	79.65
	512	30.85	**34.68**	31.65	31.05	31.45	65.36	14.08	80.39	23.17	95.42	65.98

independent full generations and has a total denoising-step budget of $4T$. In comparison, PDC's total denoising-step budget is approximately $2.5T$ when the initial output fills the length-L window, can exceed $2.5T$ for shorter outputs, and remains below $4T$ for every nonempty initial output. Thus, INIT x4 is a conservative higher-budget baseline rather than an exactly cost-matched baseline. Table 2 presents the result. Among the 15 model and dataset pairs with complete comparison results, PDC improves over INIT x4 in 13 pairs, ties in one pair, and underperforms in one pair. These results support the effectiveness of allocating inference steps to prefix-conditioned regeneration. The fixed-$L = 128$ comparison is reported in Appendix E.

3.3.2 ROBUSTNESS

We now conduct robustness checks. First, we tested several variants of the unmasking strategy. Up to this point, the results have used entropy unmasking for Dream and low-confidence unmasking for LLaDA-family models; Table 3 repeats the MATH-500 evaluation with alternative unmasking rules, including random and origin unmasking. Table 3 shows that larger keep rates also retain more initially correct answers, while smaller keep rates more often break exact repetition of the initial wrong answer. Accuracy remains comparable to or above the corresponding initial generation in these settings.

In addition, we conduct two other robustness checks. Appendix D reports LLaDA-family experiments with a block length of 32, corresponding to semi-AR inference. The results show that PDC remains effective under this semi-AR setting. Moreover, we test whether the advantage of PDC is specific to the low-temperature decoding regime by repeating the $L = 128$ comparison under more stochastic decoding at $\tau = 1.0$. At this temperature, PDC attains the highest accuracy in 12 of the 15 model and dataset settings and exceeds INIT $\times 4$ by 4.98 accuracy points on average (see Appendix H). These runs are not used for the main full-diffusion claims, but they check whether the prefix-conditioning signal also appears under a different inference time schedule.

4 CONCLUSION

In this work, we presented PDC (*Prefix-Denoising Consistency*), a test-time self-verification method. PDC exploits a distinctive consistency signal from prefix-conditioned regeneration, correct reasoning trajectories tend to be more stable and reproducible than incorrect ones. Across math and commonsense benchmarks, PDC consistently improved over the initial sample, compared favorably to independent generations under a compute-constrained setting, and remained robust across unmasking strategies and hyperparameter choices. Moreover, PDC is agnostic to the specific unmasking strategy, allowing it to be readily applied to future DLMs that adopt newly developed unmasking methods.

AI USE STATEMENT

We used generative AI tools to polish the sentences and to code during the preparation of this work. AI-assisted text was reviewed and edited by the authors, and AI-assisted code was manually inspected and tested for correctness before being used in our experiments. The authors take full responsibility for the experimental design, analysis, claims, and final content of the paper.

REPRODUCIBILITY STATEMENT

We provide the experimental settings, model and dataset details, decoding configurations, and the full specification of Prefix-Denoising Consistency in the main paper. The appendix reports additional results across generation lengths, full reproduction-rate measurements, semi-autoregressive settings, alternative unmasking strategies, and other robustness checks. These materials are intended to provide the information necessary to reproduce the main empirical findings of this work.

ETHICS STATEMENT

This work studies a test-time inference method for diffusion language models using existing language models and established reasoning benchmarks. We do not introduce new datasets or collect human-subject data. Our method is intended to improve the reliability of model outputs through self-verification; however, improved answer consistency does not guarantee factual correctness or safety in downstream applications.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's Sample Step by Step: Adaptive-Consistency for Efficient Reasoning and Coding with LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12375–12396, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.761. URL https://aclanthology.org/2023.emnlp-main.761/.

Marianne Arriola, Subham Sekhar Sahoo, Aaron Gokaslan, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Justin T Chiu, and Volodymyr Kuleshov. Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tyEyYT267x.

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

Jinbin Bai, Yixuan Li, Yuchen Zhu, Yi Xin, Qingyu Shi, Aosong Feng, Xiaohong Liu, Molei Tao, Jianru Xue, Xiangtai Li, and Ming-Hsuan Yang. PRISM: Efficient Test-Time Scaling via Hierarchical Search and Self-Verification for Discrete Diffusion Language Models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QrC8OgQyOI.

Wenrui Bao, Zhiben Chen, Dan Xu, and Yuzhang Shang. Learning to Parallel: Accelerating Diffusion Large Language Models via Learnable Parallel Decoding. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=bFJ8Sdr224.

Ahsan Bilal, Muhammad Ahmed Mohsin, Muhammad Umer, Asad Aali, Muhammad Usman Khanzada, Muhammad Usman Rafique, Zihao He, Emily Fox, and Dean F Hougen. S^3: Stratified Scaling Search for Test-Time in Diffusion Language Models. *arXiv preprint arXiv:2604.06260*, 2026.

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.

Yonggan Fu, Lexington Whalen, Abhinav Garg, Chengyue Wu, Maksim Khadkevich, Nicolai Oswald, Enze Xie, Daniel Egert, Sharath Turuvekere Sreenivas, Shizhe Diao, Chenhan Yu, Ye Yu, Weijia Chen, Sajad Norouzi, Jingyu Liu, Shiyi Lan, Ligeng Zhu, Jin Wang, Jindong Jiang, Morteza Mardani, Mehran Maghoumi, Song Han, Ante Jukic, Nima Tajbakhsh, Jan Kautz, and Pavlo Molchanov. Nemotron-Labs-Diffusion: A Tri-Mode Language Model Unifying Autoregressive, Diffusion, and Self-Speculation Decoding. Technical report, NVIDIA, 2026. Technical report.

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.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling Diffusion Language Models via Adaptation from Autoregressive Models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=j1tSLYKwg8.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. SSD-LM: Semi-autoregressive Simplex-based Diffusion Language Model for Text Generation and Modular Control. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11575–11596, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.647. URL https://aclanthology.org/2023.acl-long.647/.

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. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Naoto Iwase, Yuki Ichihara, Mohammad Atif Quamar, and Junpei Komiyama. Reliable Chain-of-Thought via Prefix Consistency. *arXiv preprint arXiv:2605.07654*, 2026.

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. *arXiv preprint arXiv:2512.09106*, 2025.

Ishan Jindal, Sai Prashanth Akuthota, Jayant Taneja, and SACHIN DEV SHARMA. THE PATH OF LEAST RESISTANCE: GUIDING LLM REASONING TRAJECTORIES WITH PREFIX CONSENSUS. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hrnSqERgPn.

Junpei Komiyama, Daisuke Oba, and Masafumi Oyamada. Best-of-∞: Asymptotic Performance of Test-Time LLM Ensembling. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3qiCnLf3jf.

Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Yi Liang, Soroush Vosoughi, and Shiwei Liu. Diffusion Language Model Knows the Answer Before It Decodes. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=g88nt4ieTG.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape Sky-high Cost: Early-stopping Self-Consistency for Multi-step Reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ndR8Ytrzhh.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution. In *International Conference on Machine Learning*, pp. 32819–32848. PMLR, 2024.

Lizhuo Luo, Zhuoran Shi, Jiajun Luo, Zhi Wang, Shen Ren, Wenya Wang, and Tianwei Zhang. DAWN: Dependency-Aware Fast Inference for Diffusion LLMs. *arXiv preprint arXiv:2602.06953*, 2026.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large Language Diffusion Models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KnqiC0znVF.

NVIDIA. Nemotron 3 Nano: Open, Efficient Mixture-of-Experts Hybrid Mamba-Transformer Model for Agentic Reasoning. *arXiv preprint arXiv:2512.20848*, 2025. URL https://arxiv.org/abs/2512.20848.

OpenAI. gpt-oss-120b & gpt-oss-20b Model Card. *arXiv preprint arXiv:2508.10925*, 2025. URL https://arxiv.org/abs/2508.10925.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP Models really able to Solve Simple Math Word Problems? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

Qwen Team. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and Effective Masked Diffusion Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

Aman Sharma and Paras Chopra. The Sequential Edge: Inverse-Entropy Voting Beats Parallel Self-Consistency at Matched Compute. *arXiv preprint arXiv:2511.02309*, 2025. URL https://arxiv.org/abs/2511.02309.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and Generalized Masked Diffusion for Discrete Data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=xcqSOfHt4g.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Wen Wang, Bozhen Fang, Chenchen Jing, Yongliang Shen, Yangyi Shen, Qiuyu Wang, Hao Ouyang, Hao Chen, and Chunhua Shen. Time Is a Feature: Exploiting Temporal Dynamics in Diffusion Language Models. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=HsB6CtagP7.

Xu Wang, Chenkai Xu, Yijie Jin, Jiachun Jin, Hao Zhang, Kai Yu, and Zhijie Deng. Diffusion LLMs Can Do Faster-Than-AR Inference via Discrete Diffusion Forcing. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=t5uLZSRjhF.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Qingyan Wei, Yaojie Zhang, Zhiyuan Liu, Puyu Zeng, Yuxuan Wang, Biqing Qi, Dongrui Liu, and Linfeng Zhang. Accelerating Diffusion Large Language Models with SlowFast Sampling: The Three Golden Principles. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Uh17FiwF4q.

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.

Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, yelong shen, Jian Jiao, Juntao Li, Zhongyu Wei, Jian Guo, Nan Duan, and Weizhu Chen. AR-Diffusion: Auto-Regressive Diffusion Model for Text Generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=0EG6qUQ4xE.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7B: Diffusion Large Language Models. *arXiv preprint arXiv:2508.15487*, 2025.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. LLaDA 1.5: Variance-Reduced Preference Optimization for Large Language Diffusion Models. *arXiv preprint arXiv:2505.19223*, 2025.

A Related Work

Diffusion language models. Diffusion models for text generation succeeded by Austin et al. (2021), advanced through the masked token framework. Recent DLM work has improved both the probabilistic formulation and the scale of masked/discrete diffusion. Score-entropy discrete diffusion, masked diffusion language modeling, and simplified masked diffusion objectives improve training and likelihood modeling for token sequences (Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024). At a larger scale, LLaDA shows that a masked diffusion model can be trained from scratch and instruction-tuned as a large language model (Nie et al., 2025), while LLaDA 1.5 studies preference optimization for such models (Zhu et al., 2025). Dream further demonstrates a strong open diffusion LLM with parallel iterative refinement and flexible generation orders (Ye et al., 2025). Our work is complementary to these models and training advances: given a base model, we study how its conditional denoising behavior can be used at inference time.

Blockwise and semi-autoregressive diffusion inference time. Several DLMs introduce left-to-right or blockwise structure to improve length flexibility and efficiency. SSD-LM and AR-Diffusion use semi-autoregressive or position-dependent denoising to combine diffusion with sequential dependencies (Han et al., 2023; Wu et al., 2023). Block diffusion interpolates between autoregressive and discrete diffusion models, enabling arbitrary-length generation and KV-cache reuse (Arriola et al., 2025). Discrete diffusion forcing similarly turns pretrained dLLMs into an AR-diffusion hybrid for faster inference (Wang et al., 2026b). These methods modify the model architecture, training objective, or inference-time schedule. In contrast, PDC is a black-box test-time procedure that probes the existing DLM by changing which parts of one completed output are held fixed.

Adaptive unmasking and early termination. A closely related acceleration direction treats DLM inference as a dynamic unmasking, token-commitment, or stopping problem. Fast-dLLM selectively unmasks tokens whose confidence exceeds a threshold, while using approximate KV caching to reduce per-step cost (Wu et al., 2025). SlowFast Sampling adapts the inference time pace using token certainty, convergence, and positional structure, alternating between exploratory and accelerated phases (Wei et al., 2026). Learning Unmasking Policies formulates masked diffusion sampling as a Markov decision process and learns token-unmasking decisions from model confidences (Jazbec et al., 2025). DAWN instead uses dependency graphs to avoid simultaneously unmasking strongly coupled uncertain tokens (Luo et al., 2026). Learn2PD trains a lightweight filter that predicts whether

each current token prediction matches the final output, and combines this with End-of-Text Prediction to terminate inference after the sequence is complete (Bao et al., 2026). Prophet observes early answer convergence and commits the remaining tokens in one step when the top-2 confidence gap indicates sufficient stability (Li et al., 2026). These methods decide when to unmask, commit, or stop in order to reduce inference cost. PDC has a different goal: it starts from a completed sample and uses fresh prefix-conditioned regenerations to test answer reproducibility and improve accuracy, rather than shortening the original denoising run.

Self-consistency method. Self-consistency (majority-voting) has been widely used as a decoding-time strategy for improving chain-of-thought reasoning. Rather than relying on a single reasoning path, Wang et al. (2023) samples multiple reasoning traces and aggregates the final answers by majority vote. Subsequent work has investigated how to reduce the sampling cost of this procedure through early termination. Adaptive Consistency (Aggarwal et al., 2023) formulates stopping as a posterior decision problem, using a Beta-binomial model over the leading answer counts and terminating once the estimated margin of the current top answer is sufficiently large. Early-Stopping Self-Consistency (Li et al., 2024) instead adopts a simpler window-based criterion, stopping when all answers within a fixed-size recent window agree. More recently, Sharma & Chopra (2025) demonstrated that sequential, entropy-aware voting can yield stronger cost-matched performance than parallel self-consistency, emphasizing the need for compute-equivalent comparisons. Prefix consistency has also been studied for autoregressive language models (Iwase et al., 2026), where a fixed prefix is followed by left-to-right continuation. Our setting differs in that a DLM retains the prefix while remasking and jointly denoising the remaining output positions. We study whether the resulting prefix-conditioned regeneration signal can be used for test-time verification in DLMs. Jindal et al. (2026) proposed an inference time method that clusters short reasoning prefixes and discards prefixes on non-dominant clusters to save computation. In a related theoretical direction, Komiyama et al. (2026) studied the asymptotic behavior of majority voting, or best-of-∞, and proposed a Bayesian nonparametric stopping rule.

Recent work has explored test-time search for diffusion language models by intervening during the denoising trajectory. Bai et al. (2026) performs hierarchical trajectory search with intermediate self-verification, pruning, and partial remasking, while Bilal et al. (2026) repeatedly expands and resamples candidate trajectories according to verifier scores. These approaches aim to allocate test-time computation toward promising trajectories during generation. In contrast, PDC operates after an initial completion has been produced and uses prefix-conditioned suffix regeneration as a post-generation consistency signal. PDC, therefore, does not require verifier-guided search over intermediate denoising states and is complementary to trajectory-search-based test-time scaling methods.

B Full reproduction rates Results

Due to space limitations, the main paper reports only the results with a keep rate of 0.5 on GSM8K, MATH-500, and SVAMP. Table 4 reports the full conditional stability diagnostics behind Figure 2.

The full table includes CSQA and SQA across keep rates $0.1, 0.5$, and 0.9. Across these additional datasets and rates, correct-answer retention is generally higher than same-wrong persistence that is consistent with the results in the main paper.

Table 4: Reproducing rate across the full model and benchmarks. We report correct-answer retention R_ρ^+, same-wrong persistence R_ρ^-, and their separation $\Delta = R_\rho^+ - R_\rho^-$. Bold marks the largest Δ within each row; negative Δ values are shown in red.

Dataset	Model	Length	R_ρ^+			R_ρ^-			Δ		
			0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
GSM8K	Dream-7B	128	97.9	98.3	99.2	82.5	82.5	93.3	15.3	**15.8**	5.8
		256	97.8	99.3	99.8	83.5	84.4	91.3	14.3	**15.0**	8.5
		512	97.5	99.3	99.5	86.8	82.8	90.7	10.7	**16.4**	8.9
	LLaDA-1.5	128	92.6	94.4	99.2	42.8	36.0	86.4	49.8	**58.4**	12.8
		256	95.2	95.0	99.6	65.9	22.0	84.7	29.4	**73.0**	14.9
		512	88.8	93.0	98.4	44.8	26.1	75.9	44.0	**66.9**	22.5
	LLaDA-8B	128	93.2	94.4	99.1	43.9	37.1	90.1	49.4	**57.3**	9.0
		256	93.0	95.0	99.6	69.5	28.3	87.9	23.5	**66.7**	11.8
		512	80.3	87.6	99.2	48.5	26.3	77.8	31.8	**61.3**	21.4
MATH-500	Dream-7B	128	90.5	92.2	97.8	78.3	74.1	91.5	12.2	**18.1**	6.3
		256	93.5	93.5	98.6	72.0	73.0	83.1	**21.5**	20.6	15.5
		512	95.7	95.2	97.8	75.2	78.6	79.5	**20.4**	16.7	18.3
	LLaDA-1.5	128	83.8	83.8	96.6	55.6	48.0	91.1	28.2	**35.7**	5.5
		256	88.3	84.4	99.2	69.3	46.6	94.1	19.0	**37.8**	5.2
		512	86.2	85.3	94.5	61.8	31.0	81.6	24.5	**54.3**	12.9
	LLaDA-8B	128	79.8	80.7	97.5	54.2	49.2	88.7	25.6	**31.5**	8.8
		256	88.6	82.6	97.7	73.0	44.8	92.6	15.7	**37.8**	5.1
		512	81.6	74.7	95.4	63.5	34.7	81.7	18.1	**40.0**	13.7
SVAMP	Dream-7B	128	98.8	98.8	99.2	74.4	71.8	79.5	24.4	**27.0**	19.7
		256	98.9	99.2	100.0	81.6	68.4	86.8	17.3	**30.8**	13.2
		512	98.8	99.6	100.0	82.9	80.0	82.9	16.0	**19.6**	17.1
	LLaDA-1.5	128	97.2	98.8	99.6	73.5	49.0	95.9	23.7	**49.8**	3.7
		256	93.4	94.7	99.1	79.2	33.3	83.3	14.2	**61.4**	15.8
		512	94.1	98.0	100.0	78.6	64.3	92.9	15.6	**33.8**	7.1
	LLaDA-8B	128	98.0	96.8	100.0	75.0	62.5	95.8	23.0	**34.3**	4.2
		256	89.4	95.6	99.4	75.0	38.6	95.5	14.4	**57.0**	3.9
		512	96.2	94.3	100.0	100.0	66.7	66.7	-3.8	27.7	**33.3**
CSQA	Dream-7B	128	98.9	96.2	97.5	96.8	87.5	91.4	2.1	**8.7**	6.2
		256	98.1	95.9	97.0	95.9	87.6	93.0	2.2	**8.3**	4.0
		512	98.8	95.4	98.5	95.1	87.5	93.1	3.7	**7.9**	5.5
	LLaDA-1.5	128	97.9	96.9	99.3	86.0	89.7	97.1	**11.8**	7.2	2.2
		256	96.6	98.3	99.8	88.1	89.0	99.6	8.5	**9.3**	0.2
		512	97.5	97.7	99.9	88.5	88.9	100.0	**9.1**	8.8	-0.1
	LLaDA-8B	128	97.6	98.6	99.7	88.7	90.8	97.5	**8.9**	7.8	2.2
		256	95.2	97.7	99.8	85.5	88.0	100.0	**9.7**	9.7	-0.2
		512	98.1	97.5	100.0	81.6	85.1	100.0	**16.6**	12.4	0.0
SQA	Dream-7B	128	98.6	95.7	96.9	92.4	91.9	97.8	**6.1**	3.8	-0.9
		256	97.4	93.8	96.0	91.7	85.4	94.3	5.8	**8.4**	1.7
		512	98.1	92.5	96.6	92.7	87.5	97.4	**5.4**	5.0	-0.8
	LLaDA-1.5	128	93.6	97.1	99.3	88.0	92.7	98.7	**5.6**	4.4	0.6
		256	94.3	96.4	98.5	87.1	89.1	95.0	7.3	**7.4**	3.5
		512	96.8	99.1	100.0	86.2	91.9	97.6	**10.6**	7.2	2.4
	LLaDA-8B	128	94.6	96.4	99.6	87.1	86.3	97.5	7.5	**10.2**	2.1
		256	87.5	96.7	99.2	90.8	84.2	93.4	-3.3	**12.5**	5.8
		512	93.8	100.0	100.0	84.6	92.3	100.0	**9.1**	7.7	0.0

C FULL-DIFFUSION ACCURACY SWEEP

As shown in Table 5, the proposed method consistently improves INIT accuracy across the completed generation lengths $L = 128, 256, 512$. This indicates that the gains reported in Table 5 are not an artifact of selecting a favorable block length, but rather reflect a robust improvement over the baseline. These results demonstrate the effectiveness of the proposed approach for improving initial-generation quality in full diffusion settings.

Table 5: Full accuracy results for Dream and LLaDA-family diffusion language models. PDC denotes majority voting over the three suffix-regenerated completions obtained with 0.1, 0.5, and 0.9 keep rates. Signed green/red values report accuracy-point changes relative to the INIT baseline. For TIF, the reported change is averaged over Fixed, Linear, and Exp. $\alpha = 5$ variants. The table includes all completed lengths in the current full-diffusion sweep.

Method / Length	GSM8K			MATH-500			SVAMP			CSQA			SQA		
	128	256	512	128	256	512	128	256	512	128	256	512	128	256	512
Dream-7B															
INIT baseline	64.06	80.06	82.03	35.80	43.20	46.00	83.00	87.00	86.33	73.22	71.74	72.40	70.45	68.56	67.98
+ TIF Fixed	62.40	79.61	81.96	34.20	43.00	46.00	84.00	87.00	86.33	73.79	71.66	72.48	70.74	69.29	68.56
+ TIF Linear	64.52	80.14	82.03	34.80	43.40	46.00	83.67	87.00	86.33	73.96	71.66	72.48	70.74	69.29	68.56
+ TIF Exp. $\alpha = 5$	65.81	80.36	82.03	35.60	44.00	46.20	84.00	87.00	86.33	73.55	71.74	72.48	70.74	69.29	68.56
Δ TIF avg. vs. INIT	+0.18	-0.02	-0.02	-0.93	+0.27	+0.07	+0.89	+0.00	+0.00	+0.55	-0.05	+0.08	+0.29	+0.73	+0.58
+ PDC	**66.03**	**81.05**	**84.08**	**37.20**	**44.60**	**47.00**	**84.33**	**88.00**	**88.00**	**74.20**	**72.24**	**73.30**	**71.76**	**71.62**	**70.60**
Δ PDC vs. INIT	+1.97	+0.99	+2.05	+1.40	+1.40	+1.00	+1.33	+1.00	+1.67	+0.98	+0.50	+0.90	+1.31	+3.06	+2.62
LLaDA-8B															
INIT baseline	58.30	60.73	18.88	23.80	26.45	17.54	83.67	53.33	17.67	80.51	42.51	48.32	65.07	17.47	4.66
+ TIF Fixed	57.77	60.12	19.26	24.00	26.25	18.15	83.67	56.33	18.00	80.51	45.29	**51.76**	65.07	**28.38**	9.46
+ TIF Linear	58.00	60.80	19.48	23.80	26.25	17.94	83.67	56.33	18.33	80.51	45.21	51.60	65.07	**28.38**	9.46
+ TIF Exp. $\alpha = 5$	58.30	60.80	19.41	24.00	26.25	18.15	83.67	56.33	18.33	80.51	45.21	51.60	65.07	**28.38**	9.46
Δ TIF avg. vs. INIT	-0.28	-0.15	+0.50	+0.13	-0.20	+0.54	+0.00	+3.00	+0.55	+0.00	+2.73	+3.33	+0.00	+10.91	+4.80
+ PDC	**65.88**	**65.88**	**33.36**	**25.40**	**27.86**	**22.38**	**84.33**	**62.67**	**31.00**	**80.59**	**45.54**	50.78	**66.67**	25.62	**9.61**
Δ PDC vs. INIT	+7.58	+5.15	+14.48	+1.60	+1.41	+4.84	+0.66	+9.34	+13.33	+0.08	+3.03	+2.46	+1.60	+8.15	+4.95
LLaDA-1.5															
INIT baseline	56.48	60.58	39.12	23.40	25.65	21.98	83.67	75.67	51.00	**80.10**	73.22	75.51	66.08	51.53	32.02
+ TIF Fixed	56.41	60.05	39.80	23.40	25.65	21.98	83.33	75.00	51.33	**80.10**	**74.37**	75.84	66.08	56.77	33.33
+ TIF Linear	56.56	60.27	39.95	23.40	25.45	22.38	83.67	75.33	51.33	**80.10**	74.28	75.76	66.08	**57.06**	33.19
+ TIF Exp. $\alpha = 5$	56.56	60.50	40.03	23.40	25.65	22.38	83.67	75.67	51.33	**80.10**	74.28	75.76	66.08	**57.06**	33.19
Δ TIF avg. vs. INIT	+0.03	-0.30	+0.81	+0.00	-0.07	+0.27	-0.11	-0.34	+0.33	+0.00	+1.09	+0.28	+0.00	+5.43	+1.22
+ PDC	**65.88**	**66.26**	**56.79**	**26.40**	**26.65**	**26.01**	**84.67**	**79.00**	**62.67**	**80.10**	74.28	**76.09**	**66.81**	52.98	**40.32**
Δ PDC vs. INIT	+9.40	+5.68	+17.67	+3.00	+1.00	+4.03	+1.00	+3.33	+11.67	+0.00	+1.06	+0.58	+0.73	+1.45	+8.30

D SEMI-AR LLADA-FAMILY RESULTS

Table 6 and Table 7 report the LLaDA-family results with block length 32.[2] When the block length is smaller than the generation length, the sequence is generated block by block, with diffusion performed independently within each block, referred to as the semi-autoregressive blockwise setting. In the semi-AR setting, the prefix cutoff is applied at the token level and is not rounded to a block boundary. If the cutoff falls inside a block, the retained prefix positions in that block remain fixed, while only the remaining masked positions are denoised before generation proceeds to subsequent blocks.

Overall, the results in Table 6 and Table 7 show that the proposed method remains effective even in this semi-autoregressive blockwise setting. Across generation lengths, the proposed method all-in-all improves INIT accuracy for the LLaDA-family models, indicating that its benefits are not limited to the full-diffusion setting used for the main claims.

[2]Note that all results reported in the main paper correspond to the setting in which the block length equals the generation length, referred to as the full diffusion setting.

Table 6: Semi-AR LLaDA-family prefix-denoising consistency results. These runs use block length 32 with generation lengths 128, 256, and 512. Values are percentages. Bold marks the largest Δ within each row; negative Δ values are shown in red.

Dataset	Model	Length	R_ρ^+			R_ρ^-			Δ		
			0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
GSM8K	LLaDA-1.5	128	95.6	99.5	99.0	78.0	91.5	89.7	**17.6**	8.0	9.2
		256	96.1	98.7	99.7	72.8	90.6	96.7	**23.3**	8.2	3.0
		512	97.7	99.2	99.9	67.8	86.8	98.7	**29.9**	12.4	1.2
	LLaDA-8B	128	95.7	98.3	99.4	74.9	87.2	91.7	**20.8**	11.2	7.6
		256	97.3	99.4	99.6	70.1	87.8	95.6	**27.2**	11.6	4.0
		512	98.1	99.5	99.9	67.1	82.3	95.4	**31.0**	17.3	4.6
MATH-500	LLaDA-1.5	128	89.4	95.9	97.6	63.0	85.3	90.9	**26.3**	10.6	6.7
		256	88.6	95.1	99.5	54.3	80.7	94.2	**34.3**	14.4	5.3
		512	88.5	93.8	99.5	45.6	66.6	93.4	**42.8**	27.2	6.1
	LLaDA-8B	128	88.8	97.5	97.5	59.4	84.0	89.9	**29.4**	13.5	7.7
		256	90.3	96.6	99.4	55.0	77.7	94.2	**35.3**	18.9	5.3
		512	82.7	91.1	98.6	43.1	70.7	93.8	**39.6**	20.5	4.8
SVAMP	LLaDA-1.5	128	98.9	99.6	99.6	68.6	88.6	100.0	**30.3**	11.1	-0.4
		256	98.5	99.6	100.0	70.0	87.5	97.5	**28.5**	12.1	2.5
		512	98.1	99.3	100.0	74.3	91.4	97.1	**23.8**	7.8	2.9
	LLaDA-8B	128	98.5	99.6	99.2	74.3	88.6	97.1	**24.2**	11.0	2.1
		256	98.9	100.0	100.0	60.5	84.2	94.7	**38.3**	15.8	5.3
		512	98.5	99.3	100.0	73.5	91.2	97.1	**25.0**	8.1	2.9
CSQA	LLaDA-1.5	128	95.7	98.3	99.3	74.5	90.7	95.7	**21.3**	7.6	3.6
		256	93.8	97.8	99.4	71.6	89.4	98.6	**22.2**	8.4	0.8
		512	94.3	97.6	99.6	69.0	84.5	97.5	**25.3**	13.1	2.0
	LLaDA-8B	128	94.8	98.7	98.8	74.1	90.7	96.4	**20.7**	8.1	2.4
		256	92.3	97.3	99.2	70.6	88.6	97.8	**21.8**	8.7	1.4
		512	93.9	96.8	99.2	65.0	84.3	96.7	**29.0**	12.5	2.4

Table 7: Semi-AR LLaDA-family main accuracy results. These runs use block length 32 with generation lengths 128, 256, and 512.

Method / Length	GSM8K			MATH-500			SVAMP			CSQA		
	128	256	512	128	256	512	128	256	512	128	256	512
LLaDA-8B												
INIT baseline	72.25	74.91	81.80	32.20	35.27	**43.15**	86.00	87.00	**88.67**	76.58	76.99	77.07
+ TIF Fixed	71.87	69.37	80.14	29.60	31.06	39.31	84.67	86.67	88.33	**78.79**	77.07	**77.40**
+ TIF Linear	73.54	74.75	81.65	30.80	34.67	41.53	87.67	86.33	88.33	77.31	77.07	**77.40**
+ TIF Exp. $\alpha = 5$	73.62	**76.27**	81.96	31.80	35.27	42.54	**88.00**	86.67	88.33	76.99	**77.31**	**77.40**
Δ TIF avg. vs. INIT	+0.76	-1.45	-0.55	-1.47	-1.60	-2.02	+0.78	-0.44	-0.34	+1.12	+0.16	+0.33
+ PDC	**73.77**	76.04	**82.49**	**33.20**	**36.47**	42.34	87.00	**88.00**	88.33	77.31	**77.31**	**77.40**
Δ PDC vs. INIT	+1.52	+1.13	+0.69	+1.00	+1.20	-0.81	+1.00	+1.00	-0.34	+0.73	+0.32	+0.33
LLaDA-1.5												
INIT baseline	72.33	78.01	82.71	34.00	36.87	**41.94**	87.67	86.67	88.33	76.66	76.58	75.92
+ TIF Fixed	71.95	72.48	80.74	32.80	33.27	38.91	84.33	86.33	87.33	**78.30**	77.23	75.92
+ TIF Linear	73.31	77.56	82.56	33.40	36.67	41.13	87.67	86.33	87.67	77.31	77.15	76.09
+ TIF Exp. $\alpha = 5$	73.46	**79.00**	**83.02**	33.80	37.07	**41.94**	**88.33**	86.33	88.33	76.82	76.58	76.17
Δ TIF avg. vs. INIT	+0.58	-1.66	-0.60	-0.67	-1.20	-1.28	-0.89	-0.34	-0.55	+0.82	+0.41	+0.14
+ PDC	**73.92**	78.47	82.71	**34.41**	**37.27**	41.53	88.00	**87.67**	**89.00**	77.23	**77.31**	**77.64**
Δ PDC vs. INIT	+1.59	+0.46	+0.00	+0.41	+0.40	-0.41	+0.33	+1.00	+0.67	+0.57	+0.73	+0.72

E SELF-CONSISTENCY RESULT AT LENGTH $L = 128$

We complement the selected-length comparison in Table 2 by fixing the generation length to $L = 128$ for all 15 model and dataset pairs. All runs use the full-diffusion setting at temperature 0.2. The INIT x4 baseline votes over four independent full generations with a total denoising-step budget of $4T$, while PDC uses one initial generation followed by three prefix-conditioned regenerations at keep rates 0.1, 0.5, and 0.9, requiring fewer total denoising steps.

As shown in Table 8, PDC improves over INIT x4 in 13 pairs, ties in one, and underperforms in one. The largest gain is $+9.25$ accuracy points on GSM8K with LLaDA-1.5. These results show that the advantage of prefix-conditioned regeneration is also observed when all pairs use a common generation length.

Table 8: We compare PDC with Self-Consistency under different total denoising-step budgets at generation length $L = 128$. For both INIT and TIF, $\times 4$ denotes four independent initial denoising trajectories. INIT x4 votes over their final answers with budget $4T$, while PDC uses one initial generation followed by three regenerations at keep rates 0.1, 0.5, and 0.9. PDC achieves the best or tied-best accuracy in 14 of 15 model and dataset settings and improves over INIT x4 in 13 of them, with gains of up to $+9.25$ accuracy points.

Dataset	**Model**	INIT	INIT x4	TIF Fixed x4	TIF Linear x4	TIF Exp. x4	PDC	Δ vs. INIT x4
GSM8K	Dream-7B	64.06	64.22	62.40	64.52	65.81	**66.03**	+1.81
	LLaDA-1.5	56.48	56.63	56.41	56.56	56.71	**65.88**	+9.25
	LLaDA-8B	58.30	58.91	58.38	58.61	58.91	**65.88**	+6.97
MATH-500	Dream-7B	35.80	36.00	34.20	34.80	35.60	**37.20**	+1.20
	LLaDA-1.5	23.40	23.20	23.40	23.40	23.00	**26.40**	+3.20
	LLaDA-8B	23.80	23.60	23.80	23.80	23.60	**25.40**	+1.80
SVAMP	Dream-7B	83.00	83.00	84.00	83.67	84.00	**84.33**	+1.33
	LLaDA-1.5	83.67	83.67	83.33	83.67	83.67	**84.67**	+1.00
	LLaDA-8B	83.67	**84.33**	**84.33**	84.00	**84.33**	**84.33**	+0.00
CSQA	Dream-7B	73.22	73.22	73.79	73.96	73.55	**74.20**	+0.98
	LLaDA-1.5	80.10	80.51	**80.67**	**80.67**	**80.67**	80.10	-0.41
	LLaDA-8B	80.51	80.34	**80.59**	**80.59**	**80.59**	**80.59**	+0.25
SQA	Dream-7B	70.45	70.45	70.74	70.74	70.74	**71.76**	+1.31
	LLaDA-1.5	66.08	65.94	65.94	65.79	65.79	**66.81**	+0.87
	LLaDA-8B	65.07	64.63	65.07	65.07	65.07	**66.67**	+2.04

F CASE STUDY: HOW PREFIX REGENERATION REPAIRS ERRORS

Section 2.2 shows that prefix-conditioned regeneration has high correct-answer preservation (R^+) and low same-wrong preservation (R^-). In words, correct initial answers tend to remain correct after regeneration, while wrong initial answers often do not reproduce the same wrong answer. This is a useful regime for PDC. High R^+ means regeneration is unlikely to disturb a correct trajectory, while a low R^- means an incorrect trajectory often has a chance to escape its original mistake. Escaping the same wrong answer does not by itself guarantee correctness; the regenerated answer may still be wrong. However, when the kept prefix contains the right setup and the erroneous intermediate step lies in the regenerated suffix, regeneration can cut away the local mistake and replace it with a corrected continuation. Table 9 shows two MATH-500 with Dream-v0-Instruct-7B examples of this behavior.

In Problem 97, the initial sample follows the correct modulo-reduction strategy but makes a local arithmetic error, treating the remainder of 2250 degrees modulo 360 degrees as 150 degrees and concluding *south*. The regenerated sample recomputes the suffix as $2250 - 6 \cdot 360 = 90$ degrees and recovers *east*. Similarly, in Problem 144, the initial sample substitutes the correct value $t = 2$, but evaluates the resulting expression as -27, whereas regeneration expands the same expression as $-12 - 12 - 5 = -29$ and recovers the correct point.

Problem 97

Problem. *A figure skater is facing north when she begins to spin to her right. She spins 2250 degrees. Which direction (north, south, east or west) is she facing when she finishes her spin?*

Initial generation

The figure skater is facing north when she begins to spin to her right. She spins 2250 degrees.

... We can do this by subtracting multiples of 360 degrees from 2250 degrees until we get an angle between 0 and 360 degrees.

2250 degrees divided by 360 degrees is 6 with a remainder of 150 degrees.

Therefore, she is facing $\boxed{\text{south}}$.

Prefix-conditioned regeneration

The figure skater is facing north when she begins to spin to her right. She spins 2250 degrees.

... We can do this by subtracting multiples of 360 degrees from 2250 degrees.

2250 - 6*360 = 2250 - 2160 = 90 degrees.

So, when she finishes her spin, she is facing $\boxed{\text{east}}$.

Problem 144

Problem. *A curve is parameterized by*

$$(x, y) = (t^3 + 7, -3t^2 - 6t - 5).$$

Find the point the curve passes through at $t = 2$.

Initial generation

When $t = 2$, we have

$$\begin{aligned}(x, y) &= (2^3 + 7, -3(2)^2 - 6(2) - 5) \\ &= (15, -27).\end{aligned}$$

So the curve passes through the point $\boxed{(15, -27)}$.

Prefix-conditioned regeneration

When $t = 2$, we have

$$\begin{aligned}(x, y) &= (2^3 + 7, -3(2)^2 - 6(2) - 5) \\ &= (8 + 7, -12 - 12 - 5) \\ &= (15, -29).\end{aligned}$$

Therefore, the point the curve passes through at $t = 2$ is $\boxed{(15, -29)}$. The answer is: (15,-29)

Table 9: Two examples in which the wrong answer is corrected by regeneration. Red text highlights the incorrect continuation in the initial sample; green text highlights the corrected continuation produced by regeneration. When the wrong intermediate step is remasked, prefix-conditioned regeneration avoids the same mistake.

Table 10: Language models used in the experiments. Dream-7B is exclusively for full diffusion, while LLaDA-family models support both full diffusion and semi-autoregressive diffusion. The main paper exclusively considers the full diffusion setting. The semi-autoregressive runs are reported separately in Appendix D.

Model	Checkpoint	Family	Main setting
Dream-7B	`Dream-org/Dream-v0-Instruct-7B`	Masked Diffusion LLM	Full-window diffusion inference time
LLaDA-1.5	`GSAI-ML/LLaDA-1.5`	Masked Diffusion LLM with Block Length	Block length matched to generation length
LLaDA-8B	`GSAI-ML/LLaDA-8B-Instruct`	Masked Diffusion LLM with Block Length	Block length matched to generation length

G Model and Dataset Summary

This appendix summarizes the language models and datasets used in the experiments. Detailed inference time hyperparameters, prompt templates, answer extraction, and voting conventions are given in Appendix I.

Table 11: Datasets used in the experiments. Math-style datasets are evaluated by canonical answer extraction and symbolic/numeric equivalence when applicable. Multiple-choice and binary reasoning datasets are evaluated by canonicalized option extraction.

Dataset	Name	Task type	Primary answer format
GSM8K	Grade-school math word problems (Cobbe et al., 2021)	Arithmetic reasoning	Boxed final answer
MATH-500	MATH subset / verification benchmark (Hendrycks et al., 2021; Lightman et al., 2024)	Mathematical reasoning	Boxed final answer
SVAMP	Arithmetic word-problem challenge (Patel et al., 2021)	Arithmetic reasoning	Boxed final answer
CSQA	CommonsenseQA (Talmor et al., 2019)	Multiple-choice commonsense reasoning	Boxed option letter
SQA	StrategyQA (Geva et al., 2021)	Binary commonsense reasoning	Boxed option letter

Table 12: Accuracy at generation length $L = 128$ under low- and high-temperature decoding. We compare the main setting ($\tau = 0.2$) with more stochastic decoding ($\tau = 1.0$). INIT $\times 4$ denotes majority voting over four independent full generations. Boldface marks the best result in each model–temperature row. Signed green/red values report the accuracy-point change of PDC relative to INIT $\times 4$ at the same temperature.

Dataset	Model	τ	INIT	INIT $\times 4$	TIF Fixed $\times 4$	TIF Linear $\times 4$	TIF Exp. $\times 4$	PDC	Δ vs. INIT $\times 4$
GSM8K	Dream-7B	0.2	64.06	64.22	62.40	64.52	65.81	**66.03**	+1.81
		1.0	41.24	50.19	45.11	47.08	49.13	**65.96**	+15.77
	LLaDA-1.5	0.2	56.48	56.63	56.41	56.56	56.71	**65.88**	+9.25
		1.0	56.94	58.83	58.15	58.15	58.23	**68.69**	+9.86
	LLaDA-8B	0.2	58.30	58.91	58.38	58.61	58.91	**65.88**	+6.97
		1.0	57.54	60.73	59.14	59.74	60.05	**69.45**	+8.72
MATH-500	Dream-7B	0.2	35.80	36.00	34.20	34.80	35.60	**37.20**	+1.20
		1.0	22.40	25.20	24.20	24.60	25.00	**37.60**	+12.40
	LLaDA-1.5	0.2	23.40	23.20	23.40	23.40	23.00	**26.40**	+3.20
		1.0	24.00	23.20	23.60	23.60	23.40	**30.20**	+7.00
	LLaDA-8B	0.2	23.80	23.60	23.80	23.80	23.60	**25.40**	+1.80
		1.0	24.45	25.05	23.85	23.85	24.25	**28.06**	+3.01
SVAMP	Dream-7B	0.2	83.00	83.00	84.00	83.67	84.00	**84.33**	+1.33
		1.0	70.33	75.67	75.00	75.67	76.00	**82.67**	+7.00
	LLaDA-1.5	0.2	83.67	83.67	83.33	83.67	83.67	**84.67**	+1.00
		1.0	85.00	84.67	85.00	84.67	84.67	**89.00**	+4.33
	LLaDA-8B	0.2	83.67	**84.33**	**84.33**	84.00	**84.33**	**84.33**	+0.00
		1.0	84.00	84.00	85.00	84.33	84.33	**86.33**	+2.33
CSQA	Dream-7B	0.2	73.22	73.22	73.79	73.96	73.55	**74.20**	+0.98
		1.0	64.78	76.49	**78.05**	77.89	77.81	74.37	-2.12
	LLaDA-1.5	0.2	80.10	80.51	**80.67**	**80.67**	**80.67**	80.10	-0.41
		1.0	80.10	**80.84**	80.67	80.75	80.75	79.44	-1.40
	LLaDA-8B	0.2	80.51	80.34	**80.59**	**80.59**	**80.59**	**80.59**	+0.25
		1.0	79.69	80.59	80.75	**80.92**	80.67	79.85	-0.74
SQA	Dream-7B	0.2	70.45	70.45	70.74	70.74	70.74	**71.76**	+1.31
		1.0	58.66	63.17	64.48	64.77	64.77	**67.10**	+3.93
	LLaDA-1.5	0.2	66.08	65.94	65.94	65.79	65.79	**66.81**	+0.87
		1.0	65.07	64.77	64.92	65.21	64.92	**66.96**	+2.19
	LLaDA-8B	0.2	65.07	64.63	65.07	65.07	65.07	**66.67**	+2.04
		1.0	64.77	64.19	64.63	64.63	64.92	**66.67**	+2.48

H RESULTS AT DIFFERENT SAMPLING TEMPERATURES

To examine whether PDC remains effective as sampling becomes more stochastic, we repeat the $L = 128$ evaluation at token-sampling temperature $\tau = 1.0$.

Results. Table 12 shows that, at $\tau = 1.0$, PDC achieves the highest numerical accuracy in 12 of the 15 model and dataset settings and outperforms INIT $\times 4$ in the same 12 settings. Averaged uniformly over the 15 settings, it exceeds INIT $\times 4$ by 4.98 accuracy points; the largest dataset-level mean gains occur on GSM8K and MATH-500, at 11.45 and 7.47 points, respectively. Across the two tested temperatures, the unweighted mean accuracy of PDC changes from 65.35 at $\tau = 0.2$ to 66.16 at $\tau = 1.0$ ($+0.81$), whereas INIT and INIT $\times 4$ decrease by 4.58 and 2.07 points, respectively. Consequently, the mean margin of PDC over INIT $\times 4$ increases from 2.11 to 4.98 points. This larger

relative margin partly reflects weaker full-generation baselines at the higher temperature and is not uniform across tasks: on all three CSQA settings, PDC trails INIT $\times 4$ by 0.74–2.12 points. The results therefore show that PDC remains effective in the tested higher-temperature setting, particularly on the mathematical reasoning tasks.

I REPRODUCIBILITY DETAILS

Models and inference time hyperparameters. We evaluate Dream-v0-Instruct-7B, LLaDA-1.5, and LLaDA-8B-Instruct. We use entropy-based and low confidence unmasking, token temperature 0.2. The generation length L is set equal to the denoising step budget $T \in \{128, 256, 512\}$.

The main tables use the block-length-matched setting, with block length set equal to the generation length. Appendix D reports semi-AR LLaDA-family runs with block length 32, as follows the TIF settings. TIF predictions are captured from the same initial denoising run with a stride of 1.

Regeneration denoising budget. During regeneration, the effective denoising budget is reduced according to the kept prefix length. In the reported full-diffusion setting, $T = L$, so the effective regeneration step budget equals the remasked suffix length:

$$T_\rho = L_\rho = L - L_\rho^{\text{prefix}}.$$

The experiment scripts also support a general nominal step budget T, using the proportional rule:

$$T_\rho = \max\{1, \text{round}(TL_\rho/L)\}.$$

This reduces to $T_\rho = L_\rho$ when $T = L$. Dream regenerates the suffix with a reduced number of denoising steps using the floor version of the same proportional rule:

$$T_\rho = \max\{1, \lfloor TL_\rho/L \rfloor\}.$$

This differs from the rounded rule by at most one denoising step, and all reported tables are computed from the saved records for the corresponding run.

Prompt template. For math-style tasks, all prompts use the same instruction suffix:

```
Please reason step by step, and put your final answer within
\boxed{}.
```

For multiple-choice tasks, we explicitly require a chain-of-thought style reasoning before the final choice. The prompt lists the answer options and includes the following mandatory format instruction:

```
Mandatory format (do not skip):
1) Write your reasoning across several sentences (facts,
definitions, or everyday knowledge you use).
2) Only after that reasoning, end with exactly one line
that contains only one capital option letter (A--E) inside
\boxed{}, e.g.  \boxed{C}.
```

Thus, the multiple-choice setting is not a direct-answer-only setting; the model is instructed to produce reasoning and then end with a single boxed option letter.

Answer extraction and equivalence. For math datasets, we extract the last boxed expression when present. If no boxed expression is found, we applies the final answer fallback patterns such as "The answer is", "answer:", and "final answer is" in the last lines of the output. Extracted answers are normalized using Hendrycks-MATH-style string normalization, including removal of superficial LaTeX wrappers, unit text, spacing, percent signs, and simple fraction normalization. If normalized strings do not match exactly, we use a SymPy fallback: both expressions are parsed with the LaTeX parser when possible, otherwise with sympify, and are judged equivalent when the symbolic simplification of their difference is zero. Numeric decimal strings are compared with tolerance 10^{-6}.

For multiple-choice tasks, the extracted answer is canonicalized to a choice letter. Exact option-text matches are mapped to their corresponding letter before loose letter-form matching, to avoid treating option text such as formulas as a choice letter. Common letter forms such as `C`, `(C)`, `C.`, and `\text{C}` are accepted.