Looped Diffusion Language Models

Selective recurrence in masked diffusion
Research group discussion

Paper authors: Sanghyun Lee, Chunsan Hong, Seungryong Kim
Jonghyun Lee, Jongho Park, Dongmin Park

arXiv:2605.26106v1 (May 25, 2026)
Reading: October 13, 2026

Reuse two early layers to improve quality at fixed training FLOPs.

arxiv.org/abs/2605.26106v1. Presentation prepared for paper discussion; not an author presentation.

A second iteration axis inside each denoising step

Research question

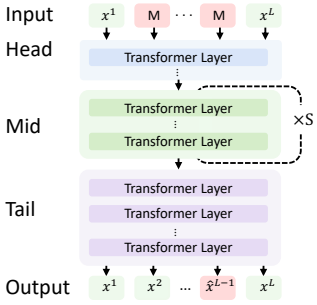
With fixed parameters and total training FLOPs, should an MDM spend more computation on each example through shared layers?

Resource	What is repeated?	State being updated
Denoising steps T	Entire denoising network	Discrete sequence x_t
Internal loops S	A shared subset of layers	Hidden states $H_t^{(k)}$

- Unresolved positions can exchange information before token commitment.
- More work per example means fewer training tokens at fixed FLOPs.

Paper, Secs. 1–3, PDF pp. 1–4.

Loop only a small block of the denoising network



Selective Looping

$$H_t^{(0)} = \text{head}(x_t),$$
$$H_t^{(k)} = \text{mid}(H_t^{(k-1)}),$$
$$p_\theta^S(x_0 \mid x_t) = \text{tail}(H_t^{(S)}).$$

- $H_t^{(k)} \in \mathbb{R}^{L \times d}$: all token positions.
- The mid-block shares parameters across S applications.
- Head and tail run once; tail includes the vocabulary projection.

$$D_{\text{eff}} = n_h + Sn_m + n_o$$

Paper, Fig. 1 left panel, PDF p. 2; Eq. (4), PDF p. 3.

Random loop counts train one operator at many depths

$$S \sim \mathcal{U}\{1, \ldots, S_{\max}\}, \qquad t \sim \mathcal{U}[0, 1]$$

$$\mathcal{L} = \mathbb{E}_{t,S,x_t}\left[\sum_{i=1}^{L} \mathbb{I}[x_t^i = \mathrm{m}] \, \frac{\alpha_t'}{1-\alpha_t} \log \langle p_\theta^{S,i}(x_t, t), x_0^i \rangle\right]$$

- Apply the usual masked diffusion loss to the final loop output.
- α_t: decreasing token retention probability.
- x_0^i: one-hot target; the inner product selects its probability.
- Both α_t' and log probability are nonpositive: this is a weighted negative log-likelihood.

Observed extrapolation

Train with $S_{\max} = 12$, evaluate up to $S = 24$ in language modeling.

Paper, Eq. (5), PDF p. 4; Fig. 2, PDF p. 5.

Shared parameters still require repeated computation

$$F_{\text{loop}} = F_{\text{base}} + (\mathbb{E}[S] - 1) n_m F_{\text{layer}}, \qquad \mathbb{E}[S] = \frac{S_{\max} + 1}{2}$$

12-layer language model	Loops	Layer executions
Non-looped MDM	—	12
LoopMDM: training average	6.5	23
LoopMDM: inference	12	34
LoopMDM: inference	24	58

Match total training FLOPs by reducing LoopMDM training steps. The models use the same corpora but see different token totals.

Paper, Eq. (6), PDF p. 4. Layer counts are a teaching calculation, excluding embedding/output costs.

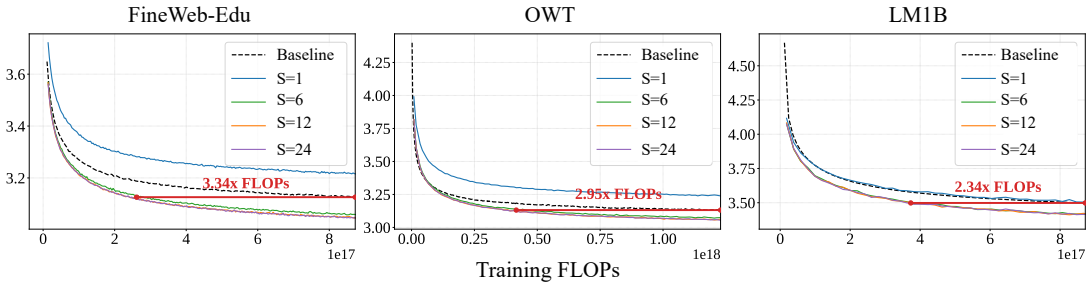
Three experiments probe different claims

Setting	Model and data	Comparison / evaluation
Language	170M, 12 layers, $d = 768$; LM1B, OWT, FineWeb-Edu	Same-parameter MDM; matched total training FLOPs
Math	125M, 14 layers, $d = 512$; TinyGSM with Python solutions	14/18/21-layer MDMs; GSM8K, Top-2 and Top-3 unmasking
Sudoku	1 layer, $d = 384$; 100k train / 1,000 test	Fixed left-to-right order; exact-solve evaluation

Main LM configuration: loop zero-based layers 1–2 (the second and third layers), $S_{\max} = 12$. Math uses $S_{\max} = 8$; Sudoku uses 6.

Paper, Secs. 4.1–4.3, PDF pp. 4, 7–8; Appendix C, pp. 18–20.

Selective looping reaches the target NLL sooner



- Solid: LoopMDM at different inference S; dashed: MDM.
- $S \geq 6$ improves NLL; $S = 1$ underperforms; $S = 24$ still helps.
- Figure annotations: FineWeb-Edu 3.34×, OWT 2.95×, LM1B 2.34× less training compute to the target.

Paper, Fig. 2, PDF p. 5. Version note: Sec. 4.1, p. 4 reverses the FineWeb-Edu/LM1B ratio assignments. The figure's mapping is shown here.

Generation quality improves at fixed denoising steps

Steps T	MDM	LoopMDM $S = 6$	LoopMDM $S = 12$
64	116.71 ± 2.72	105.59 ± 3.83	99.06 ± 5.91
256	79.43 ± 2.60	64.08 ± 1.39	56.45 ± 0.50
1024	42.56 ± 1.09	39.23 ± 1.31	38.53 ± 1.23

- Generative perplexity: lower is better; five random seeds.
- Both external denoising steps and internal loops spend inference compute.
- This table fixes T, rather than total inference FLOPs.

Paper, Table 1, PDF p. 5; selected rows with the original uncertainty values. The scoring model is not specified in the described protocol.

Transfer PPL improves, with task-specific saturation

Corpus	MDM	$S = 6$	$S = 12$
PTB	108.5	100.2	90.2
WikiText	35.1	34.2	32.4
LM1B	68.1	68.0	67.8
Lambada	49.6	44.6	45.8
AG News	69.3	59.2	56.6
PubMed	43.0	37.2	36.5
Arxiv	37.8	33.1	32.7

Paper, Table 2, PDF p. 5. Means over five random seeds; uncertainty omitted here.

- All seven corpora improve over MDM at $S = 12$.
- Largest change: PTB, -18.3 PPL units.
- Lambada is better at 6 loops than at 12.
- At $S = 1$, all seven are worse than MDM.

Average downstream gains are uneven across tasks

Task	MDM (%)	$S = 12$ (%)	Change (pp)
BoolQ	46.6	57.4	+10.8
RACE	28.5	31.7	+3.2
HellaSwag	31.9	32.7	+0.8
ARC-c	25.5	25.3	−0.2
OBQA	30.8	28.0	−2.8
All nine: average	40.0 ± 0.3	41.4 ± 0.6	+1.4

Six tasks improve, one ties, two decline. BoolQ contributes much more than most tasks.

Paper, Table 3, PDF p. 6; selected task means and original average uncertainty. The text claims all nine match/improve, but ARC-c and OBQA decline in the table.

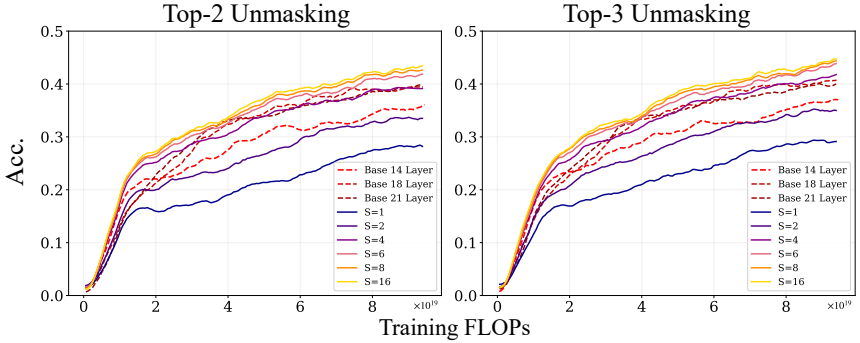
Where and how much to loop are critical

Factor	Selected configuration	Test NLL ↓
Non-looped baseline	12 layers	3.798
Position	Start at 1 or 2 (zero-based)	3.729
Position	Start at 10	3.777
Looped layers	$n_m = 2$	3.729
Looped layers	$n_m = 12$	3.933
Training max loops	$S_{\max} = 12$	3.729
Training max loops	$S_{\max} = 24$	3.800

More repeated work per example leaves fewer training steps. Two early-middle layers capture most of the benefit.

Paper, Table 4, PDF p. 6. LM1B, 170M, matched training FLOPs. Training with 24 loops differs from evaluating a 12-loop-trained model at 24.

Math gains persist against deeper training baselines



- At $S = 16$: +8.5 pp (Top-2), +8.1 pp (Top-3) vs. 14 layers.
- Matches the 14-layer endpoint with 43% of its training FLOPs.
- Also exceeds the 21-layer model: reported +2.7 pp at $S = 8$.

Paper, Sec. 4.2 and Fig. 3, PDF p. 7. TinyGSM-trained 125M model; three random seeds; $S_{\max} = 8$.

Training and inference have different effective depths

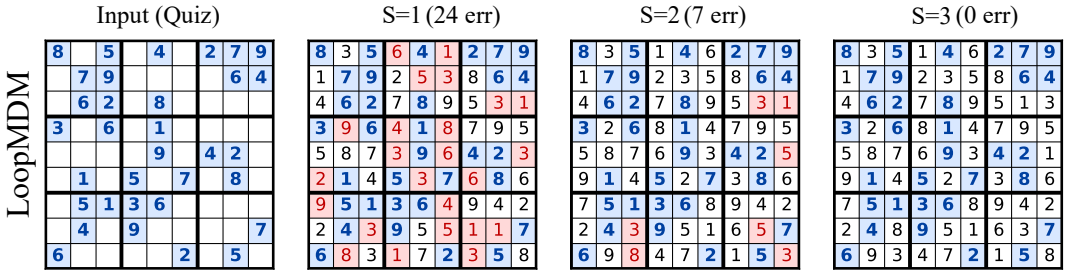
$$D_{\text{eff}}(S) = 14 + 2(S-1) = 12 + 2S$$

Model / operating point	Loop count	Layer executions
Deeper non-looped baseline	—	21
LoopMDM training average	$\mathbb{E}[S] = 4.5$	21
LoopMDM inference	$S = 8$	28
LoopMDM inference	$S = 16$	44

- Reusing computation improves the training-budget trade-off.
- An equal-inference-FLOPs accuracy frontier remains to be measured.

Teaching calculation from the architecture in Sec. 4.2, PDF p. 7. Layer counts approximate compute and exclude embedding/output overhead.

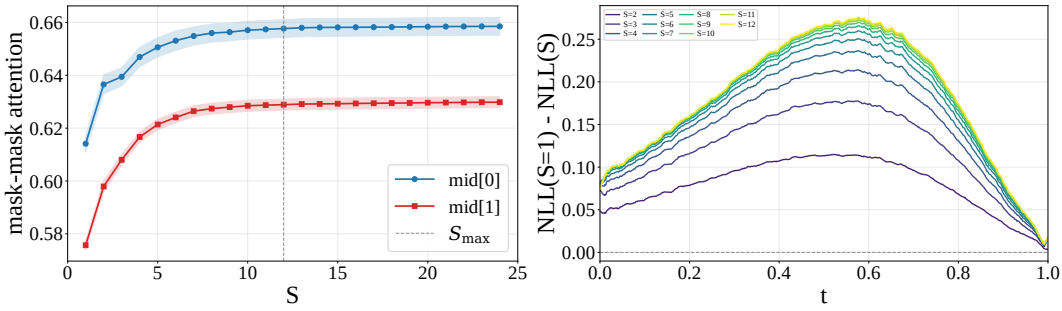
Sudoku improves even with a fixed generation order



- Fixed left-to-right unmasking removes the advantage of solving easy cells first.
- One-layer baseline: 10.9% exact solves; looped model at $S = 1$: 10.6%.
- Shown puzzle: 24 errors $\rightarrow$ 7 $\rightarrow$ 0 at $S = 1, 2, 3$.

Paper, Fig. 4 and Sec. 4.3, PDF p. 8. The illustration is one puzzle; aggregate exact-solve rates for larger S are not provided.

Loops increase interactions between unresolved positions



- Left: mask-to-mask attention at $t = 0.5$ grows and then saturates.
- Right: $\mathrm{NLL}_t(1) - \mathrm{NLL}_t(S)$ peaks at intermediate noise.
- Interpretation: useful context and unresolved positions coexist there.

Paper, Fig. 5, PDF p. 9. OWT, length 1024, $S_{\max} = 12$. Attention trends support the workspace interpretation; they are not a causal intervention.

Stop looping when hidden-state changes become small

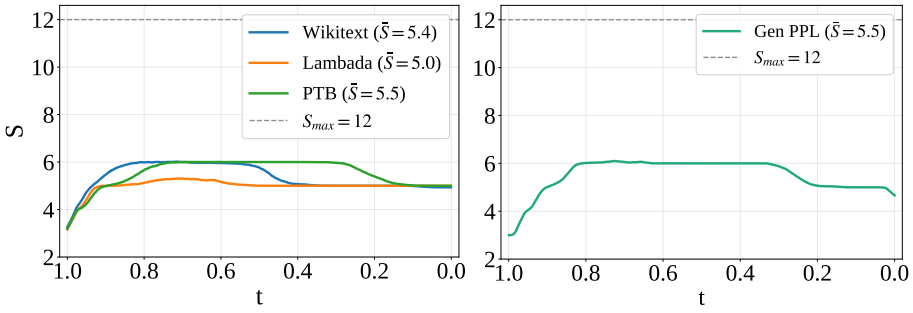
$$r_k = \frac{\|H^{(k)} - H^{(k-1)}\|}{\|H^{(k)}\|}; \qquad \text{stop if } r_k < \epsilon \text{ or the budget is reached.}$$

Metric	Fixed $S = 12$	Adaptive	Mean loops
Zero-shot PPL $\downarrow$	51.7	52.9	5.1
Generative PPL $\downarrow$	38.53	39.11	5.2
Downstream accuracy $\uparrow$	41.4%	41.3%	4.9

With $\epsilon = 0.1$, about five loops retain most of the gains. This monitors state stability rather than answer correctness.

Paper, Sec. 4.3 and Table 5, PDF p. 9. Norm type and batch aggregation are not specified in the text.

Adaptive loops concentrate on intermediate timesteps



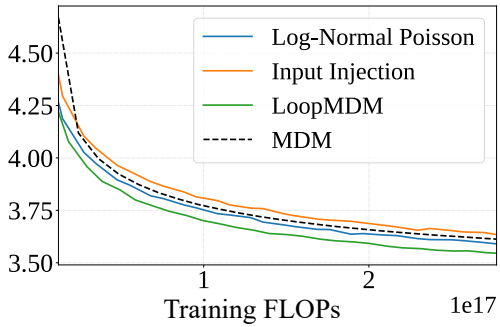
$$\frac{\text{adaptive layer executions}}{\text{fixed-12 layer executions}} \approx \frac{10+2(5.2)}{10+2(12)} = 0.60$$

About 40% fewer layer executions in this teaching estimate. The loop-count ratio alone excludes head and tail computation.

Paper, Fig. 6, PDF p. 21; OWT, $\epsilon = 0.1$, 100 sequences. Time runs from 1 to 0. Cost estimate uses Table 5's generative mean, not measured latency.

Simple uniform loop sampling beats the tested variants

- Injecting Gaussian noise into mask embeddings worsens NLL relative to MDM.
- Log-normal Poisson loop sampling gives only a modest benefit.
- Uniform sampling performs best in this comparison.



The tested input injection is specifically a mask-noise variant.

Paper, Appendix D.2 and Fig. 7, PDF pp. 20–21.

More loops can move a correct prediction away

A failure trajectory from OpenWebText

Ground-truth token: `with`

Loop count S	1	6	8	24
Prediction	`with`	`who`	`whether`	`whether`
Target NLL	0.75	2.38	5.31	6.69

- Other examples improve copying, semantic choice and syntax.
- Aggregate gains coexist with local prediction drift.
- NLL here scores the true token, not confidence in the wrong token.

Paper, Fig. 8(A–E), PDF p. 22. Selected values from panel E.

The theory isolates workspace as an extra resource

Construction

For fixed k, add n^k masked slots, one for each candidate k-tuple in an n-vertex graph.

Each slot checks its edges; an answer slot aggregates the results. Constant internal depth can evaluate candidates in parallel.

Conditional separation

Assume the unpadded architecture has an AC^0 simulation of size $O(N^C)$, with $N = \Theta(n^2)$.

For $k > 8C$, the invoked Clique circuit lower bound rules out that unpadded model.

Extra slots increase computation. Equal constant loop budgets do not establish equal FLOPs.

Paper, Example 1 and Appendix B, PDF pp. 8, 16–18. The circuit simulation is an explicit assumption.

The result is a training-budget trade-off worth testing

1. **Architecture:** a small shared block can be more useful than broadly adding repeated depth.
2. **Mechanism:** unresolved positions offer space for iterative communication before discrete commitment.
3. **Deployment:** choose S, denoising steps and stopping rules jointly; high-loop quality has a compute cost.

Discussion

At fixed inference FLOPs, how should compute be split between loops and denoising steps? Does the result persist at billion-parameter scale?

Paper, Sec. 5 and Appendix E, PDF pp. 9–10, 22–23. Main experiments: 125M–170M; manually selected loop block and stopping rule.

Paper reference and discussion guide

Sanghyun Lee, Chunsan Hong, Seungryong Kim, Jonghyun Lee, Jongho Park, Dongmin Park.

Looped Diffusion Language Models.
arXiv:2605.26106v1, submitted May 25, 2026.

`https://arxiv.org/abs/2605.26106v1`

- Method: Sec. 3; quality and efficiency: Secs. 4.1–4.2.
- Mechanism and adaptive inference: Sec. 4.3.
- Proof assumptions and extra results: Appendices B and D.

All experimental values and reused figures in this deck come from this version. Arithmetic estimates and interpretation are identified on their slides.

Backup: masking and the sign of the NELBO

$$q(x_t^i \mid x_0^i) = \alpha_t \delta_{x_0^i}(x_t^i) + (1 - \alpha_t)\delta_{\mathrm{m}}(x_t^i)$$

$$w(t) = -\frac{\alpha_t'}{1 - \alpha_t} \geq 0$$

$$\mathcal{L} = \mathbb{E}_{t,S,x_t}\left[\sum_{i:x_t^i=\mathrm{m}} w(t)(-\log p_\theta^{S,i}(x_0^i \mid x_t, t))\right]$$

- $t = 0$: nearly clean; $t = 1$: nearly fully masked.
- Only masked positions contribute to the denoising loss.
- Predictions share contextual hidden states even when reverse sampling factorizes across positions.

Paper, Eqs. (1), (3), (5), PDF pp. 3–4. Last equation is an algebraic rewriting for explanation.

Backup: all nine downstream tasks

Accuracy (%)	MDM	$S = 1$	$S = 6$	$S = 12$
ARC-c	25.5	23.6	25.7	25.3
ARC-e	47.7	45.2	47.7	47.7
BoolQ	46.6	53.8	53.8	57.4
HellaSwag	31.9	31.1	32.5	32.7
OBQA	30.8	28.2	28.0	28.0
PIQA	60.6	58.5	58.9	61.1
RACE	28.5	27.2	31.3	31.7
SIQA	36.8	35.9	37.4	37.3
WinoGrande	51.3	50.9	51.4	51.5
Average	40.0	39.4	40.7	41.4

Paper, Table 3, PDF p. 6. Means; original uncertainty omitted. Five evaluation seeds do not by themselves establish five independent pretraining runs.

Backup: the three architecture sweeps

Start position

Position	NLL
0	3.744
1	3.729
2	3.729
5	3.734
8	3.747
10	3.777

Block size

n_m	NLL
1	3.743
2	3.729
3	3.734
6	3.784
7	3.857
12	3.933

Training loop cap

$S_{\max}$	NLL
4	3.741
8	3.733
12	3.729
16	3.740
20	3.763
24	3.800

Baseline NLL: 3.798. The best reported configuration uses a small block and a moderate training cap.

Paper, Table 4, PDF p. 6. Selected rows; matched training FLOPs. Position is zero-based.

Backup: threshold ablation has a different baseline

Setting	Zero-shot PPL	$\bar{S}$	Gen. PPL	$\bar{S}$
Fixed $S = 12$	55.58	12.00	62.44	12.00
$\epsilon = 0.02$	55.57	11.73	62.66	10.73
$\epsilon = 0.10$	56.68	6.20	64.39	5.45
$\epsilon = 0.20$	59.45	3.74	66.03	3.41

- Larger thresholds generally trade quality for fewer iterations.
- Table 6's fixed baselines and loop means differ from Table 5. The paper does not fully reconcile the protocols.
- Its Compute percentages track $\bar{S}/12$ numerically, excluding the non-looped network portion.

Paper, Table 6, PDF p. 20; compare Table 5, PDF p. 9. These tables are kept separate rather than merged into one trade-off curve.

Backup: the conditional Clique separation

$$c_{\mathbf{v}} = \left(\bigwedge_{r<s} A_{v_r, v_s} \right) \wedge \left(\bigwedge_{r<s} [v_r \neq v_s] \right), \qquad \text{answer} = \bigvee_{\mathbf{v} \in [n]^k} c_{\mathbf{v}}.$$

- One workspace position per candidate tuple: n^k positions.
- Constant k gives constant local resources and $O(k \log n)$ bits per tuple index.
- Assume the fixed unpadded architecture has an AC^0 simulation of size $O(N^C)$.

$$N = \Theta(n^2), \quad k > 8C \quad \Longrightarrow \quad O(N^C) = O(n^{2C}) = o(n^{k/4}).$$

This contradicts the invoked $\omega(n^{k/4})$ Clique circuit lower bound. Extra workspace is the distinguishing computational resource.

Paper, Appendix B.2–B.3, PDF pp. 16–18; the paper cites Rossman (2008) for the circuit lower bound.