Clock Diffusion

Efficient Semi-Autoregressive Continuous Diffusion Language Models

Paper discussion: Schiff et al. (NVIDIA / Cornell)

arXiv:2610.00894v1 · 19 October 2026

Stagger denoising across tokens, then cache the finished prefix.

Reading presentation; not authored by the paper's authors. arxiv.org/abs/2610.00894v1

A continuous LM needs a prefix that stops changing

Full-sequence diffusion

Repeatedly update all token latents.

Dynamic future context prevents permanent prefix KV reuse.

Clock Diffusion

Give positions different noise schedules.

Completed tokens become fixed context; an active window keeps denoising.

- Block clocks finish a group together; sliding clocks finish tokens in order.
- The contribution combines a probability model, training masks, and sampling.

Paper, Secs. 1–3, PDF pp. 2–6; relationship to Rolling Diffusion: Sec. 5, p. 9.

Local clocks determine which positions need an update

$$u^{\ell}(t) = \mathrm{clip}_{[0,1]}\big(rt - \varphi(\ell)\big), \qquad r = 1 + \max_{\ell'} \varphi(\ell').$$

- Global time runs from 1 to 0 during generation.
- A larger phase $\varphi(\ell)$ makes a token finish earlier.
- For $s < t$, only $\mathcal{A}_{s|t} = \{\ell : u^{\ell}(s) < u^{\ell}(t)\}$ changes.

A fixed clock means a fixed latent

$u^{\ell}(s) = u^{\ell}(t)$ implies a Dirac transition: $q(\mathbf{z}_s^{\ell} \mid \mathbf{z}_t^{\ell}, \mathbf{x}^{\ell}) = \delta(\mathbf{z}_s^{\ell} - \mathbf{z}_t^{\ell})$.

Paper, Sec. 3.1, Eqs. (2)–(9), PDF pp. 3–4.

Sliding clocks turn denoising into a pipeline

$$\varphi_{\mathrm{SW}}(\ell) = \frac{L-\ell}{S}, \qquad u^{\ell}(t) = \mathrm{clip}_{[0,1]}\left(\frac{(L+S-1)t+\ell-L}{S}\right).$$

Sliding clocks: L = 8, S = N = 4

Teaching illustration from Eq. (11). Columns show the state after each denoising call.

	0	1	2	3	4	5	6	7	8	9	10	11
Token 1	prior	0.75	0.50	0.25	done	done	done	done	done	done	done	done
Token 2	prior	prior	0.75	0.50	0.25	done	done	done	done	done	done	done
Token 3	prior	prior	prior	0.75	0.50	0.25	done	done	done	done	done	done
Token 4	prior	prior	prior	prior	0.75	0.50	0.25	done	done	done	done	done
Token 5	prior	prior	prior	prior	prior	0.75	0.50	0.25	done	done	done	done
Token 6	prior	prior	prior	prior	prior	prior	0.75	0.50	0.25	done	done	done
Token 7	prior	prior	prior	prior	prior	prior	prior	0.75	0.50	0.25	done	done
Token 8	prior	prior	prior	prior	prior	prior	prior	prior	0.75	0.50	0.25	done

Model calls --> (global diffusion time t runs from 1 to 0)

Local time: 1 = not started; 0 < u < 1 = denoising; 0 = committed. Total: L + S - 1 = 11 calls.

The newly finished token is cached in the next fused forward; prompt/setup overhead is excluded here.

New teaching illustration from Eq. (11), PDF p. 5, and Appendix D, p. 25. L: length; S: window; N: updates/token.

ClockDLM predicts tokens and denoises their embeddings

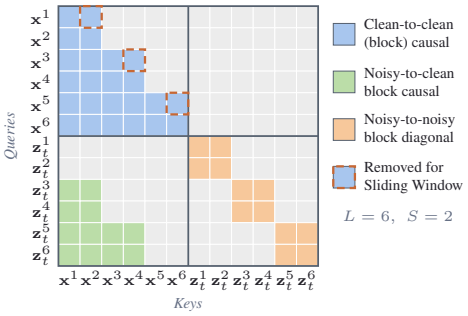
$$\mathbf{e}^{\ell} = E^{\top}\mathbf{x}^{\ell}, \qquad \mathbf{z}_t^{\ell} \sim \mathcal{N}\Big(\alpha(u^{\ell}(t))\mathbf{e}^{\ell}, \sigma(u^{\ell}(t))^2 I_d\Big)$$

$$\mathbf{x}_\theta^{\ell} \in \Delta^V \quad \longrightarrow \quad \mathbf{e}_\theta^{\ell} = E^{\top}\mathbf{x}_\theta^{\ell} \quad \longrightarrow \quad \text{Gaussian reverse update}$$

- Learned embedding matrix $E \in \mathbb{R}^{V \times d}$; $d = 16$, unit-norm rows.
- The denoiser sees the active latents, local noise levels, and clean prefix.
- RePlaid components: learned schedule, self-conditioning, logit prior.

Paper, Sec. 3.5 and Appendix B.1–B.4, PDF pp. 6, 18–21. Conditional-task sampling uses top-1 embeddings.

Causal clean states make permanent KV reuse valid

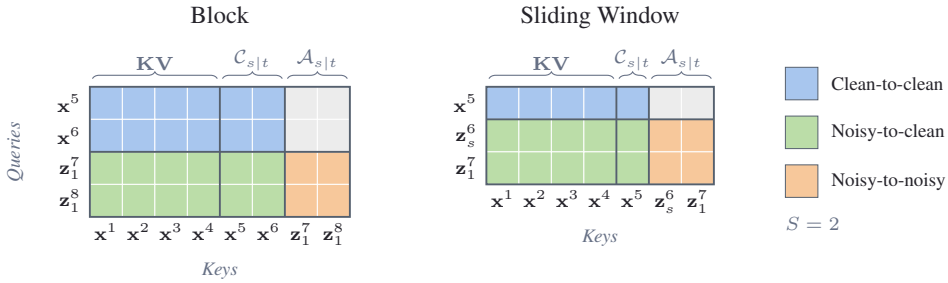


$$\mathcal{M} = \begin{pmatrix} \mathcal{M}_{\mathrm{CC}} & 0 \\ \mathcal{M}_{\mathrm{NC}} & \mathcal{M}_{\mathrm{NN}} \end{pmatrix}$$

- Training layout: clean L tokens followed by noisy L tokens.
- Noisy groups see their clean prefix and their own noisy group.
- Sliding windows remove the outlined clean-to-clean edges: token causality.

Paper, Fig. 2 and Sec. 3.3, PDF p. 5. Rows are queries; columns are keys.

One forward can write clean KV and denoise the window



- Concatenate newly committed tokens with the next active noisy window.
- Clean queries never see noisy keys: their cached states remain valid.
- Fuse caching with the next denoising pass to avoid an extra forward.

Paper, Appendix D and Fig. 7, PDF p. 26. Existing KV forms the leftmost key group.

The likelihood bound charges only moving positions

$$-\log p_\theta(\mathbf{x}^{1:L}) \le \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{recons}} + \mathcal{L}_{\text{diffusion}}.$$

- Prior: terminal latent versus the Gaussian prior.
- Reconstruction: read out each token when it completes.
- Diffusion: reverse-transition KL over awake positions only.

For Gaussian embedding diffusion

$$\mathrm{KL}_\ell = \frac{1}{2}\big[\mathrm{SNR}(u^\ell(s)) - \mathrm{SNR}(u^\ell(t))\big]\,\|\mathbf{e}^\ell - \mathbf{e}_\theta^\ell\|_2^2.$$

Paper, Eq. (10), Appendix A and Eq. (23), PDF pp. 4, 17–19. Frozen positions have identical Dirac transitions.

Training proxies and likelihood evaluation use different clocks

Efficient training

- Draw independent times, then sort within each S-token group.
- Random gaps support variable-stride Cache Grab sampling.
- Reconstruction proxy: all noisy positions at $u = 0$.

PPL evaluation

- Randomly shift the group partition and attention mask.
- Restore time gaps of $1/S$.
- Estimate reconstruction and diffusion separately and without bias.

The cleaner-context reconstruction proxy is not itself a valid likelihood bound.

Paper, Appendix C.2, Algorithms 2–3 and Eqs. (49)–(52), PDF pp. 24–25.

Confidence commits the longest eligible prefix

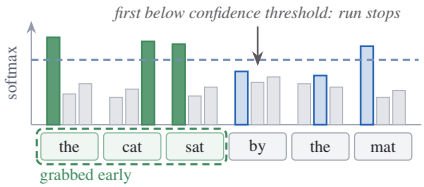
A token is eligible when its budget expires **or** its top-token probability reaches a threshold $\eta < 1$.

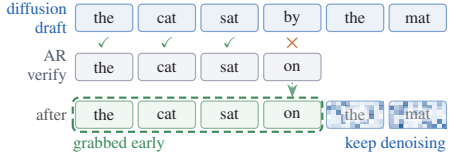
Window position	1	2	3	4
Top probability	0.98	0.96	0.70	0.99
Commit at $\eta = 0.95$	yes	yes	no	no

- The example assumes that no token has exhausted its budget.
- Keep the remaining latents and ages; refill on the right.
- No retraining required. A block model must wait for the whole block.

Teaching example of Algorithms 4–5, Appendix E.1, PDF pp. 26–27. $\eta = 1$ disables confidence commits.

SSD verifies drafts and keeps unfinished continuous states





- Retrain with joint AR + diffusion loss; share model parameters.
- Two calls per iteration: diffusion draft, then causal AR verification.
- Accept the matching prefix plus the AR correction at the first mismatch.
- Retain the uncommitted tail; reaching $u = 0$ does not force a commit.

Paper, Fig. 3 (right), Appendix E.2 and F, PDF pp. 6, 27–28. Top-1 output matches this joint model's AR decoding.

A sliding window does not automatically reduce call count

$$T_{\text{block}} = \frac{L}{S}N, \qquad T_{\text{SW}} = \frac{L+S-1}{S}N.$$

- At $N = S$, the standard sliding sampler takes $L + S - 1$ denoising calls.
- It commits one token per call after the pipeline fills.
- Lower budgets use cohorts; Cache Grab skips work by early commits.

How to interpret efficiency

NFE counts all forwards, including setup and SSD verification.
It does not measure latency: the tokens and attention work per call differ.

Paper, Appendix D, PDF pp. 25–26; NFE accounting in Appendix F, p. 28. Fixed-length formulas exclude setup overhead.

Three tasks test different consequences of the design

Training data	Evaluation	Updates	Main metric
OpenWebText	100k held-out docs	1M	PPL bound ↓
TinyGSM	GSM8K, 1,319 tasks	250k	pass@1 ↑
CNN/DailyMail	11,490 test articles	up to 500k	ROUGE ↑

- OWT / math: 12-layer DiT, width 768, 12 attention heads.
- Summarization: Qwen3-style architecture, trained from scratch.
- GSM8K is evaluated by executing generated Python programs.
- Generation: seed 1, batch 1 per GPU; NFE SD is across examples.

Paper, Appendix F and Tables 4–6, PDF pp. 28–32. Tokenizer names do not imply pretrained model weights.

OWT: 19.5 is the best reported diffusion PPL bound

Model	S	PPL $\downarrow$
AR	—	17.5
RePlaid	full	21.3
BD3LM	16	22.3
BD3LM	4	20.7
B-ClockDLM	4	20.2
ClockDLM	16	20.6
ClockDLM	4	**19.5**

Paper, Table 1, PDF p. 7. Some baseline values are taken from prior work; RePlaid is retrained.

The "38%" claim

AR gap: $3.2 \rightarrow 2.0$.

$(3.2 - 2.0)/3.2 = 37.5\%$.

The PPL reduction versus BD3LM itself is about 5.8%. Diffusion values are variational upper bounds; AR is ordinary PPL.

GSM8K: quality and call savings come in distinct stages

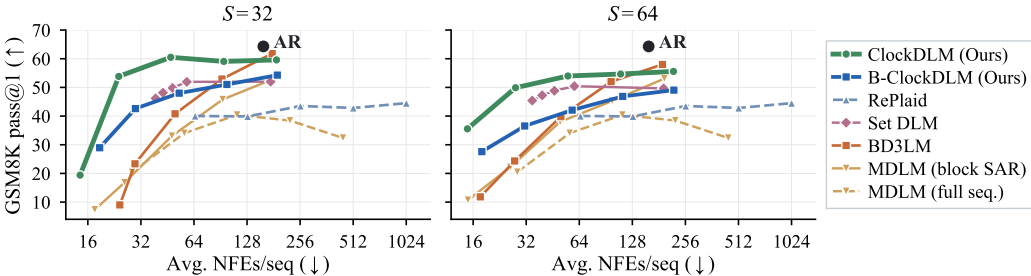
Model / sampler	Mean NFE ↓	pass@1 ↑
AR baseline	158.4	64.3%
RePlaid	1025.0	44.5%
BD3LM, $S = 32$	177.7	61.8%
B-ClockDLM, $S = 32$	190.1	54.3%
ClockDLM, $S = 32$	188.1	59.6%
+ Confidence, $\eta = 0.7$	59.5	60.0%
+ SSD, joint training	**35.5**	**67.6%**

Confidence: 3.16× fewer calls than standard ClockDLM.

SSD: 4.46× fewer calls and +3.3 accuracy points versus the AR baseline.

Paper, Tables 2 and 7, PDF pp. 7, 33. Clock models use $N = S$; SSD requires a separately trained joint model.

The sliding model improves the math budget frontier



- Compare points toward the upper left: higher accuracy, fewer calls.
- Standard $S = 32$ ClockDLM is below BD3LM at the main-table budget; several smaller budgets favor the sliding model.

Paper, Fig. 4, PDF p. 8. Log-scale NFE axis. Clock models sweep N; SW-SetDLM sweeps a confidence threshold.

Confidence saves calls, with task-dependent quality costs

Model	S	Standard: NFE / Acc.	$\eta = 0.7$: NFE / Acc.
ClockDLM	32	188.1 / 59.6%	59.5 / 60.0%
ClockDLM	64	219.1 / 55.6%	78.4 / 53.6%
B-ClockDLM	32	190.1 / 54.3%	71.1 / 53.4%
B-ClockDLM	64	220.1 / 49.1%	103.6 / 48.4%

- A sliding model can commit a prefix; a block model waits for all positions.
- Lower thresholds reduce calls, but GSM8K accuracy is not monotonic.
- At $S = 64$, the sliding model loses 2.0 accuracy points at $\eta = 0.7$.

Paper, Appendix G.2 and Table 7, PDF pp. 32–33. $N = S$; NFE means shown.

Joint training helps even when SSD is not used

Window S	Diffusion loss only	Joint AR + diffusion	Change
4	65.7%	66.6%	+0.9 pp
8	63.8%	67.2%	+3.4 pp
16	60.0%	63.8%	+3.8 pp
32	59.6%	63.5%	+3.9 pp
64	55.6%	61.3%	+5.7 pp

Both columns use standard diffusion decoding

NFE is essentially unchanged. At $S = 32$: 188.1 versus 187.4.
The main-table SSD gain therefore includes a training effect.

Paper, Table 8, PDF p. 33. Accuracy on all 1,319 GSM8K test examples. pp = percentage points.

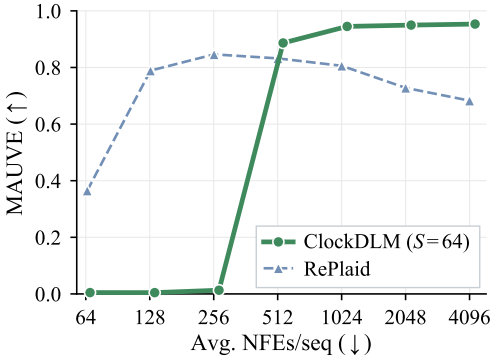
Summarization still trails discrete diffusion

Model	NFE	R-1	R-2	R-L
AR	73.1	35.9	14.8	25.0
BD3LM, $S = 32$	110.7	39.4	17.5	27.0
SW-SetDLM, $S \leq 64$	72.2	39.7	17.9	27.8
RePlaid, $T = 1024$	1025.0	32.2	12.7	22.1
ClockDLM, $S = 32$	120.3	36.0	14.1	23.8

- Versus RePlaid: all three ROUGE scores improve; 8.52× fewer calls.
- Discrete models offer better quality at similar or lower call counts.
- ROUGE does not directly measure factual accuracy.

Paper, Table 3, PDF p. 8; protocol in Appendix F.3, pp. 30–32. All models trained from scratch; NFE means shown.

Open-ended generation exposes a low-budget failure region



- ClockDLM is stronger around 512+ calls.
- At roughly 64–256 calls, its MAUVE is near zero.
- At a total budget near 256: about 16 updates/token for ClockDLM, versus 256 for full-sequence RePlaid.

Paper, Appendix G.1 and Fig. 8, PDF pp. 31–32. $S = 64$; 5,120 sequences; each model uses its best tested filtering settings.

The main lesson is to co-design clocks and dependencies

1. **Representation:** continuous embedding diffusion can support variable-length generation and a stable prefix KV cache.
2. **Computation:** prefix commits save calls; Confidence preserves the existing diffusion weights, while SSD adds joint AR training and verification.
3. **Evidence:** OWT likelihood and math improve, while summarization and low-budget open generation remain concrete weaknesses.

Next tests: closer training/commit-time alignment, larger models, and measured serving latency.

Synthesis from Sec. 6, Tables 1–3, Appendix C.2 and Fig. 8. Proposed tests are discussion points.

Paper reference and discussion

Clock Diffusion: Efficient Semi-Autoregressive Continuous Diffusion Language Models

Yair Schiff, Omer Belhasin, Roy Uziel, Matan Rusanovsky, Ran Zilberstein, Marianne Arriola, Gilad Turok, Guanghan Wang, Volodymyr Kuleshov, Michael Elad.

NVIDIA and Cornell University. arXiv:2610.00894v1, 1 October 2026.
https://arxiv.org/abs/2610.00894v1

Discussion question

At a fixed latency budget, how should a sampler allocate window width, per-token denoising depth, and early-commit confidence?

Main talk ends here (approximately 25 minutes). Backup slides contain derivations and additional protocol details.

Backup: completion is a boundary with noisy future context

$$\tau_\ell = \frac{L-\ell}{L+S-1}, \qquad \mathcal{A}_{\tau_\ell^+} = \{\ell, \dots, \min(\ell+S-1, L)\}.$$

$$u^j(\tau_\ell) = \frac{j-\ell}{S}, \qquad \mathcal{C}_{\tau_\ell^+} = \{1, \dots, \ell-1\}.$$

$$\mathcal{L}_{\text{recons}}^{\infty,\text{SW}} = \sum_{\ell=1}^{L} \mathbb{E}_q \left[-\log p_\theta \left(\mathbf{x}^\ell \mid \mathbf{z}_{\tau_\ell}^{\mathcal{A}_{\tau_\ell^+}}, \mathbf{x}^{<\ell} \right) \right].$$

- At $S = 4$, the completion ladder is $[0, 1/4, 1/2, 3/4]$.
- Each reconstruction target has a different window and clean prefix.

Paper, Appendix B.6, Eqs. (41)–(44), PDF pp. 22–23.

Backup: unbiased validation uses shifted windows

Choose $\rho \sim \mathrm{Uniform}\{0, \ldots, S-1\}$; shift groups and mask. Let $j(\ell) = (\ell - \rho) \bmod S$.

Reconstruction

$$u^\ell = \frac{j(\ell)}{S}, \qquad \widehat{\mathcal{L}}_{\mathrm{recons}} = S \sum_\ell \mathbf{1}[j(\ell) = 0] \cdot \mathrm{CE}_\ell.$$

CE is token cross-entropy; each token is a head with probability $1/S$.

Diffusion

$$u^\ell = \frac{j(\ell) + U}{S}, \qquad U \sim \mathrm{Uniform}[0, 1] \text{ per group.}$$

Random position offsets make each token's local time uniform on $[0, 1]$.

Paper, Appendix C.2, Eqs. (50)–(52), PDF p. 25. Validation: four forwards/batch with self-conditioning.

Backup: the continuous-time loss includes a clock derivative

$$\mathcal{L}^{\infty}_{\text{diffusion}} = \int_0^1 \mathbb{E}_q \left[\sum_{\ell \in \mathcal{A}_{t+}} -\frac{1}{2} \mathrm{SNR}'(u^\ell(t)) \dot{u}^\ell(t) \|\mathbf{e}^\ell - \mathbf{e}^\ell_\theta\|_2^2 \right] dt.$$

- $\mathrm{SNR}(u) = \alpha(u)^2/\sigma(u)^2$ decreases as forward noise grows.
- For sliding clocks, $\dot{u}^\ell = (L + S - 1)/S$ while a position is awake.
- Under $du = \dot{u}^\ell(t)dt$, the clock derivative cancels the measure change.

Assumptions: absolutely continuous clocks and a continuously differentiable base SNR schedule (derivatives almost everywhere).

Paper, Eq. (27), PDF p. 19; Eqs. (45)–(48), p. 23. Denoiser conditioning is suppressed in the display.

Backup: a learned residual corrects an analytic token prior

With unit-norm embedding rows, the Gaussian log-likelihood gives

$$\pi_t^\ell = \frac{\alpha(u^\ell(t))}{\sigma(u^\ell(t))^2} E z_t^\ell.$$

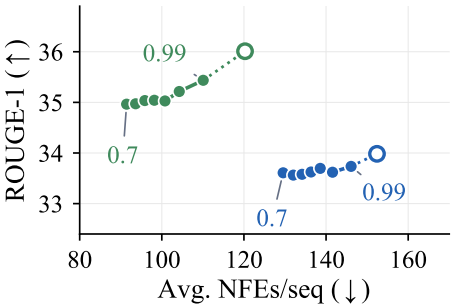
The final vocabulary probabilities are

$$\mathbf{x}_\theta^\ell = \mathrm{softmax}\left(30 \tanh\left(\frac{h_\theta^\ell}{30}\right) + \pi_t^\ell\right).$$

- h_θ^ℓ is the learned vocabulary-logit vector.
- Soft-capping applies to the learned logits, not to the analytic prior.
- Input rescaling and self-conditioning stabilize the 16-dimensional latents.

Paper, Appendix B.4, Eqs. (31)–(32), PDF p. 21. Noise schedules and optimizer settings vary by task.

Backup: summarization confidence gains are modest



- Hollow points: standard decoding.
- Lower threshold: fewer calls, generally lower ROUGE-1.
- From $\eta = 0.99$ to 0.7: about 1.1–1.2× fewer calls; at most 1.3% relative ROUGE-1 reduction.
- The threshold comparison starts at 0.99, not at the hollow points.

Paper, Sec. 4.3 and Fig. 6, PDF pp. 8–9. Green: $S = 32$; blue: $S = 64$.

Backup: conditional-task protocols differ in meaningful ways

TinyGSM $\rightarrow$ GSM8K

- Code continuation, context 512.
- Padding contributes to diffusion-model loss.
- Execute the generated program; tolerance 10^{-3}.
- Fixed 250k-step EMA checkpoint.

CNN/DailyMail

- Article 512 tokens; target 180; canvas 768.
- Padding excluded from loss and attention.
- Minimum 50 generated non-EOS tokens; length/repetition penalties.
- Best validation-loss checkpoint.

Paper, Appendix F.2–F.3 and Tables 5–6, PDF pp. 29–32. Both use DDPM with top-1 clean-token estimates.