Test-Time Generation

Zefan Cai[1] **Hao Tan**[3] **Yuchen Zhu**[2] **Tianyuan Zhang**[4] **Sai Bi**[3] **Haoyi Qiu**[5]
Yicong Hong[3] **Jason Kuen**[3] **Haoliang Wang**[3] **Haozhe Zhao**[6] **Ruiyi Zhang**[7]
Ryan A. Rossi[3] **Wanrong Zhu**[8] **Molei Tao**[2] **Wen Xiao**[9] **Rajiv Jain**[3]
Junjie Hu[1] **Jiuxiang Gu**[3]

[1]University of Wisconsin–Madison [2]Georgia Institute of Technology [3]Adobe
[4]Massachusetts Institute of Technology [5]UCLA Computer Science Department
[6]University of Illinois at Urbana-Champaign [7]Apple [8]Meta [9]Microsoft

Abstract

Some long-context architectures compress sequences into fixed-size fast-weight memories with deterministic regression readouts. At its population optimum, squared-error regression returns the conditional mean and can average values compatible with the same query. Test-Time Generation (`TTG`) formulates fast-weight recall as conditional generation without enlarging the fast-weight state. Online updates train the state on noisy key–value bindings; each inference read starts from fresh noise and follows a query-conditioned denoising trajectory with memory fixed. A simplified analysis suggests that the trajectory uses key similarity and value-space evidence. In a controlled two-values-per-key probe, stochastic reads recover both bindings at low and moderate load, unlike the tested deterministic controls. Fixing readout noise reduces two-value coverage, and higher load further degrades retrieval, consistent with separate read- and write-side limitations. Across the reported long-context runs from 124M to 3B parameters, `TTG` shows higher mean RULER accuracy than the corresponding deterministic reference. A single readout step retains this language-model improvement; extra steps do not improve the model trained with one readout step in our depth control. In novel view synthesis, `TTG` achieves higher PSNR than the tested fast-weight baselines. These results motivate studying readout design separately from fast-weight state size.

1 Introduction

Long contexts are increasingly central to agents, language models, and generative visual systems, yet retaining the full history makes memory and access costs grow with sequence length. This motivates recurrent, state-space, fast-weight, and test-time-training architectures that compress a stream into a fixed-size online state, in contrast to attention's growing key–value cache (Schlag et al., 2021; Yang et al., 2024a; Gu & Dao, 2024; Sun et al., 2025; Behrouz et al., 2024; Zhang et al., 2025). Because one fixed-size state must serve many accumulated bindings, retrieval depends not only on what is written but also on how it is read. Existing work largely focuses on the state and its update rule. A common readout in this setting is deterministic regression, whose squared-loss population optimum is the conditional mean (Bishop, 1994; Sohn et al., 2015). When a cue is compatible with several stored values, a point prediction can merge them even when the state encodes distinct alternatives. Preserving information and expressing it are separate requirements.

Test-time training (TTT) turns a model's hidden state into a learner: as a sequence arrives, a self-supervised objective updates fast weights that later tokens can query (Sun et al., 2025). This makes online learning itself a memory mechanism, but frequent parameter updates become costly for long sequences. Subsequent work improves online update rules and memory organization (Behrouz et al., 2024; 2025a;b); Cartridges trains a compact key–value cache offline on a corpus for reuse across queries (Eyuboglu et al., 2025). LaCT, for example,

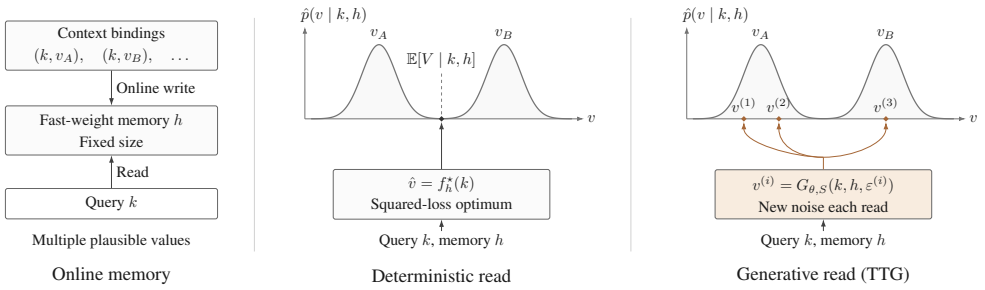


Figure 1: Readouts at matched fast-weight capacity. (left) Online writes form a fixed-size state h. (middle) The squared-loss optimum $f_h^\star(k)$ returns the conditional mean. (right) $G_{\theta,S}$ samples with fresh noise and frozen h. Curves smoothly illustrate the discrete binding conditional. Mode recovery depicts a multi-step regime; at the Bayes optimum, $S = 1$ returns the mean.

groups tokens into large chunks and applies one fast-weight update per chunk, improving hardware efficiency while retaining a nonlinear, fixed-size online state (Zhang et al., 2025). Despite these advances in how context is written and maintained, the LaCT readout remains a deterministic endpoint. When the state retains evidence for several cue-compatible values, this interface can express only one prediction. Our motivation is therefore to improve how the learned state is read, rather than simply enlarge it or change the update schedule.

Inspired by constructive, cue-conditioned episodic recall (Schacter et al., 1998; Hassabis et al., 2007; Yassa & Stark, 2011; Nader et al., 2000), we introduce Test-Time Generation (`TTG`). As shown in Fig. 1, `TTG` retains LaCT's memory architecture and chunked update operator, but trains on noisy bindings and replaces endpoint regression with a key-conditioned denoising trajectory. At inference, each read starts from fresh noise and follows this trajectory while the updated memory remains fixed; repeated reads can represent different plausible values without expanding fast-weight capacity. The key distinction from endpoint regression is how evidence evolves during retrieval, not memory size.

Our analysis makes this distinction precise. Key similarity assigns prior weight to query-compatible bindings, while the evolving sample contributes value-space evidence that separates their candidate values along the trajectory (Tong et al., 2024; Bertrand et al., 2025). Unlike associative recall from weights fixed after training, this readout queries a fast-weight state written online from the current context. We test this account with a controlled gated-MLP probe that writes two values per key. At low and moderate loads, stochastic reads recover both alternatives, whereas deterministic readouts do not, even with enlarged fast-weight capacity or a matched denoiser architecture; fixing the readout noise likewise eliminates coverage. At the highest tested load, all readouts fail. These controls link two-value recall to stochastic reading while exposing write-side interference.

Our contributions are: (i) we formulate fast-weight readout as conditional generation and introduce `TTG`, replacing endpoint regression with a key-conditioned denoising trajectory over a fixed-size online state; (ii) we decompose retrieval margins into key-space compatibility and value-space evidence, and design probes distinguishing stochastic readout from added capacity, noisy training, and write-side interference; and (iii) we instantiate `TTG` in long-context language modeling and novel view synthesis (NVS), where it improves retrieval and reconstruction over corresponding deterministic references while characterizing write and readout trade-offs.

2 Related Work

Transformers use self-attention for direct token interactions (Vaswani et al., 2017), with quadratic full-sequence computation and a linearly growing autoregressive key–value cache. These costs motivate bounded-state alternatives. Linear attention accumulates the past in a fixed-size summary interpretable as fast weights storing key–value associations

(Katharopoulos et al., 2020; Schlag et al., 2021). Related state-space and gated recurrent models use input-dependent dynamics, decay, and targeted overwrite (Gu & Dao, 2024; Yang et al., 2023; 2024a). Our comparisons include YOCO (Sun et al., 2024), DeltaNet (Yang et al., 2024b), Mamba-2 (Dao & Gu, 2024), and KDA (Kimi Team et al., 2025).

The bounded state can itself become an expressivity bottleneck. Test-time training (TTT) addresses this by representing recurrent state as the parameters of an online learner, with self-supervised updates writing incoming observations into fast weights for later queries (Sun et al., 2025). Canonical TTT performs frequent tokenwise or small-minibatch updates, which are memory-bound; LaCT improves this write-side efficiency by aggregating losses over large chunks and updating once per chunk (Sun et al., 2025; Zhang et al., 2025). `TTG` targets an orthogonal, read-side limitation: a deterministic fast-weight readout returns one prediction for each query and state, whose squared-loss population optimum is the conditional mean (Bishop, 1994; Sohn et al., 2015). It instead formulates retrieval as query-conditioned generative refinement from a fixed memory. Although iterative, this readout differs from both the online write recursion of TTT and recurrent-depth architectures, which repeatedly transform evolving sequence representations (Dehghani et al., 2019; Giannou et al., 2023; Geiping et al., 2025; Zhu et al., 2025; Bae et al., 2025): here the query and memory remain fixed while only the candidate value evolves. Additional refinement therefore allocates read-time computation to a value-space sample without changing the write schedule or memory capacity.

Among continuous generative formulations, our denoising readout is closest to linear conditional flow matching with a rectified-flow-style straight path (Liu et al., 2022; Lipman et al., 2023). It interpolates noise and memory values, then integrates with Euler steps. Related work studies conditional transport couplings (Cheng & Schwing, 2025) and weighted flow matching (Calvo-Ordonez et al., 2025). Predicting the clean value rather than velocity yields an equivalent pointwise vector field under the linear bridge, while changing how errors at different noise levels are weighted. The trajectory is conditioned on a query and an online-written fast-weight state that remains fixed throughout reading. The formulation is therefore more naturally described as straight-path flow matching than as a diffusion probabilistic or score-based model: it learns neither reverse transition kernels nor a score field, although these families share Gaussian denoising ingredients (Ho et al., 2020; Song et al., 2021). Prior work interprets denoising as associative retrieval from fixed weights; energy-based models offer another route to iterative inference (Ambrogioni, 2024; Hoover et al., 2023; LeCun et al., 2006; Du & Mordatch, 2019).

3 Method

Preliminaries. Test-time training (TTT) represents recurrent state h as an online learner's fast weights (Sun et al., 2025). Offline training learns slow parameters θ and initialization h_0 from next-token or image-reconstruction loss, differentiating through the implemented reads and updates. At test time, θ and h_0 remain fixed: each independent sequence or scene starts with a copy of h_0 and fresh optimizer buffers, retaining its updated state between groups. Slow projections supply write bindings (k_j, v_j, η_j). At the objective level, a group C_b updates the fast weights as

$$\mathcal{L}_b^{\text{write}}(h_b) = \sum_{j \in C_b} \eta_j \ell_{\text{write}}(h_b; k_j, v_j), \qquad h_{b+1} = \text{Opt}\Big(h_b, \nabla_{h_b} \mathcal{L}_b^{\text{write}}(h_b)\Big). \tag{1}$$

Here Opt includes the online optimizer's state and transformations (Zhang et al., 2025); the notation η_j omits head and matrix indices. The gradient notation describes the write objective; implemented kernels use explicit residual-based approximations to this direction, and outer gradients follow those operations. We abstract the resulting state as effective bindings $\mathcal{D}_h = \{(k_i, v_i)\}_{i=1}^M$, with association weights $\pi_i(k; h) := P(i \mid k, h)$ for query k and empirical conditional $\hat{p}(v \mid k, h) = \sum_i \pi_i(k; h)\delta(v - v_i)$. This is a conceptual view of a fixed-size state, not an explicit table (Geva et al., 2021; Ramsauer et al., 2021; Ambrogioni, 2024; Hoover et al., 2023). In Fig. 1, conventional TTT (middle) returns a deterministic value $\hat{v} = f_h(k)$, whereas `TTG` (right) uses the conditional generative readout $\hat{v} = \mathcal{R}(k, h, \varepsilon)$ with $\varepsilon \sim \mathcal{N}(0, I)$. Denoising and energy-based sampling instantiate this readout.

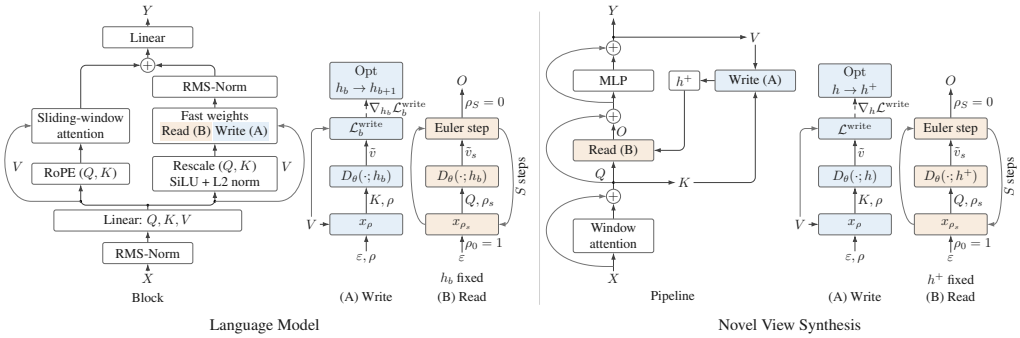


Figure 2: Language-model (left) and novel-view-synthesis (right) interfaces. Blue (A) accumulates weighted denoising losses into $\mathcal{L}_b^{\text{write}}$; dashed arrows supply its gradient to Opt (Eq. 1). Orange (B) reads with frozen state: prefix h_b before the language-model chunk update, or updated scene state h^+ for target-camera queries. Scene-group indices are suppressed. Write repeats r are omitted; training uses d read steps and evaluation uses S.

Generative fast-weight readout. TTG writes observed bindings into a gated fast-weight MLP and, with the resulting state fixed, generates each queried value from fresh noise. For linear flow matching, let $x_t = \alpha_t \varepsilon + \beta_t v$, where $\alpha_t = 1 - t$ and $\beta_t = t$ (Lipman et al., 2023; Liu et al., 2022; Albergo et al., 2025). For the idealized binding mixture, Bayes' rule gives posterior weights and mean

$$w_i(x, t, k; h) = \frac{\pi_i(k; h)\, \mathcal{N}(x; \beta_t v_i, \alpha_t^2 I)}{\sum_j \pi_j(k; h)\, \mathcal{N}(x; \beta_t v_j, \alpha_t^2 I)}, \qquad m_t(x, k; h) = \sum_i w_i v_i. \tag{2}$$

Under squared loss, m_t is optimal. For distinct values and positive priors, the ideal bridge posterior concentrates on its generating component as $t \uparrow 1$ (Appendix Observation C.3). With $\Delta_{ij} := \log(w_i / w_j)$, the expected margin of i over j is

$$\mathbb{E}[\Delta_{ij} \mid i, k, h] = \log \frac{\pi_i(k; h)}{\pi_j(k; h)} + \frac{\beta_t^2}{2\alpha_t^2} \|v_i - v_j\|^2. \tag{3}$$

The two terms measure key-space compatibility and value-space evidence, respectively, but only for distinctions retained in h (Appendix Proposition C.4).

With $\rho = 1 - t$, let $x_\rho = \rho\varepsilon + (1 - \rho)v$ and let $D_\theta(x, \rho, k; h)$ approximate this posterior denoiser. Each query draws fresh noise and holds h fixed, starting at $x_{\rho_0} = \varepsilon$ on the uniform schedule $\rho_s = 1 - s/S$. Implementations normalize Gaussian draws to their value-space convention (L_2 for the language model, RMS for NVS); the Gaussian mixture above is an analytical idealization. At step $s = 0, \ldots, S - 1$, predict $\tilde{v}_s = D_\theta(x_{\rho_s}, \rho_s, k; h)$, set $\lambda_s = (\rho_s - \rho_{s+1})/\rho_s$, and update $x_{\rho_{s+1}} = (1 - \lambda_s)x_{\rho_s} + \lambda_s \tilde{v}_s$. These Euler steps define $G_{\theta,S}(k, h, \varepsilon) := x_{\rho_S}$. More steps add fast-MLP evaluations without changing capacity. In the ideal mixture, the Bayes-optimal $S{=}1$ read returns $m_0(\varepsilon, k; h) = \sum_i \pi_i(k; h) v_i$; mode separation requires $S > 1$. A finite network trained jointly with the task need not equal this optimum: its single-step gains do not establish multimodal recovery.

Each head stores $h = (W_0, W_1, W_2)$ and computes $F_h(z) = [\text{SiLU}(zW_0) \odot (zW_2)]W_1$. Adaptive layer normalization (AdaLN) maps the key plus sinusoidal time embedding to input shift, scale, and output gate: the modulated noisy value enters F_h, whose gated output is normalized. The illustrated NVS variant also adds the key to this input; additive conditioning instead sums value, key, and time embedding. Slow feature projections and conditioning belong to θ, outside h.

Write objective, repeats, and trained depth. For each observed (k, v), sample noise and $\rho \sim \mathcal{U}[0, 1]$, form x_ρ, and define

$$\ell_{\text{write}}(h; k, v) = \mathbb{E}_{\rho,\varepsilon}\left[\left\| D_\theta(x_\rho, \rho, k; h) - v \right\|_2^2\right]. \tag{4}$$

The inner update changes only fast weights, using tokenwise rates and residuals accumulated over C_b. For $\rho > 0$, $v - x_\rho = \rho\,(v - \varepsilon)$, so $u_\theta = (D_\theta(x_\rho, \rho, k; h) - x_\rho)/\rho$ gives $\|u_\theta - (v - \varepsilon)\|_2^2 = \rho^{-2}\|D_\theta(x_\rho, \rho, k; h) - v\|_2^2$. Thus v-prediction and velocity prediction share the pointwise optimum, but differ by weight ρ^{-2} when averaged over noise levels. *Stochastic repeats* r supply independent (ρ, ε) draws for each binding; residuals are averaged over the repeated group before one optimizer update. *Trained readout depth* d sets the number of read steps in the offline forward pass, through which the task loss backpropagates. The inner write still uses one denoiser prediction per corrupted binding, not a d-step endpoint loss. Evaluation depth S can differ from d; neither depth improves results monotonically (Section 4.2; Fig. 4).

Discrete-time variants sample ρ uniformly from $\{1 - j/d\}_{j=0}^{d-1}$. With $d = 1$, this fixes $\rho = 1$, the pure-noise endpoint. The experimental label $t \equiv 1$ uses the implementation's noise-level variable, equal to our ρ, rather than the clean-endpoint bridge time $t = 1$.

Shared online execution. Chunks use local window attention and read prefix state h_b with the sampler above. Slow feature projections supply $\{(k_j, v_j, \eta_j)\}_{j \in C_b}$ for the single boundary update in Eq. (1). All S readout evaluations reuse the cached low-precision fast weights. A *shared-conditioning* variant reuses the timestep and AdaLN parameters across blocks (Peebles & Xie, 2023; Dhariwal & Nichol, 2021). Matched deterministic and matched-noise controls test conditioning capacity and write/read stochasticity, respectively (Section 4.4).

Task-specific interfaces. As shown in Fig. 2, the shared read/write mechanism is instantiated for language modeling and novel view synthesis. The sampler above and the write objective in Eq. (4) realize the conditional generative readout with rectified flow (Liu et al., 2022; Lipman et al., 2023); Table A4 also reports results for an energy-based readout. For language modeling, text has no natural chunk boundary, so B is a hyperparameter and reads use a shifted apply-before-update order (Fig. 2, left). Before writing $C_b = \{n_b, \ldots, n_b + B - 1\}$, $\mathcal{D}_{h_b} = \{(k_i^{\mathrm{upd}}, v_i)\}_{i < n_b}$. Each $\hat{v}_n = G_{\theta,S}(k_n^{\mathrm{read}}, h_b, \varepsilon_n)$ for $n \in C_b$ reads only this prefix, preserving causality; then $x_{\rho,n} = \rho\varepsilon_n + (1 - \rho)v_n$ trains the chunk update via Eq. (4). The recurrent TTT interface remains unchanged (Sun et al., 2025; Tandon et al., 2025; Hu et al., 2025; Zhang et al., 2025).

For novel view synthesis, an LVSM-style model patchifies posed images and target poses (Fig. 2, right) (Jin et al., 2025). Observed tokens supply write bindings $\{(k_j^{\mathrm{upd}}, v_j)\}_{j \in C_b}$, where k_j^{upd} encodes pose, patch position, and context, and v_j is the reconstruction latent. Target-camera tokens then produce $\hat{v}_q = G_{\theta,S}(k_q^{\mathrm{read}}, h_{b+1}, \varepsilon_q)$, approximating the posterior in Eq. (2). The feed-forward posed-token interface remains unchanged (Zhang et al., 2025; Jin et al., 2025). In both tasks, recovery is limited by the key-space evidence retained in $\pi_i(k; h)$, which governs binding separability at read time (Eq. (3) and Section 4.4).

Each forward pass uses one noise trajectory per query. The sampled latent returns to the block through normalization, projection, and residual addition; final features are decoded by the language-model vocabulary head or NVS RGB-patch head.

4 Experiments

We evaluate `TTG` on long-context language modeling and NVS, testing retrieval and reconstruction from a fixed-size recurrent state. We measure quality, analyze write and readout choices, and use a binding probe to separate readout effects from capacity and write interference.

4.1 Experimental Setup and Metrics

We train 124M, 760M, and 3B language-model families on Long-Data-Collections (Together AI, 2023) with the Mistral tokenizer (Jiang et al., 2023), matching data, optimizer, schedule,

and 4K-token chunks (B=4096) within each fast-weight contrast. Evaluation reports deep-context per-token loss (PTL; archived 30K–32K tail summaries) on 32K-token Book-3 (Gao et al., 2020) sequences and mean RULER (Hsieh et al., 2024) accuracy over 13 tasks at 4K, 8K, 16K, and 32K. Scale labels identify model families; conditioning changes total parameter counts. Appendix A documents the historical PTL aggregation conventions.

Object-level NVS models train on Objaverse (Deitke et al., 2023) and evaluate on GSO (Downs et al., 2022); scene-level models train and evaluate on DL3DV-10K (Ling et al., 2023). Both use an 8-layer, width-512 backbone, 12 training views, 24 evaluation views, and PSNR. As an L_2 criterion, PSNR alone cannot falsify mean collapse; retrieval and binding provide sharper tests.

4.2 Long-Context Language Modeling

Main comparison. `TTG` with r=d=1 leads bounded-state retrieval at all three scales (Table 1; Fig. 3). Relative to LaCT, mean RULER rises by 5.02, 1.88, and 3.8 points at 124M, 760M, and 3B; the 32K margins are +3.78, +5.44, and +5.6. At 124M, it exceeds full attention (17.38 vs. 16.52) and LaCT at every length (Appendix Table A1). At 760M, it has the lowest reported PTL and leads LaCT at 8K–32K, but trails at 4K (43.96 vs. 46.67). Lower PTL does not imply better retrieval: KDA-SWA and Mamba-2-SWA have lower PTL at 124M, while `TTG` improves 3B RULER despite 0.039 worse PTL than LaCT. This disagreement motivates our probe but does not establish memory-value averaging: next-token prediction and the inner squared-error objective are different losses. Section 4.4 tests competing-value recovery directly.

Table 1: Long-context summary. Deep-context PTL and mean RULER accuracy over 4K, 8K, 16K, and 32K. Bold marks the best bounded-state value at each scale. The `TTG` row uses r=d=1; † denotes an unbounded-cache reference. Fig. 3 shows loss–retrieval trajectories across model sizes; Appendix Table A1 gives exact values.

	124M		760M		3B	
Model	PTL ↓	RULER ↑	PTL ↓	RULER ↑	PTL ↓	RULER ↑
Transformer (full attention) †	3.0869	16.52	2.5978	34.16	2.3881	51.87
YOCO †	3.0661	15.14	2.5749	26.39	—	—
Transformer-SWA	3.1243	14.50	2.6325	22.27	2.4544	30.73
Gated DeltaNet-SWA	3.0994	12.32	2.6075	21.48	2.3900	31.31
DeltaNet-SWA	3.1336	11.69	2.6086	20.77	2.3974	25.81
KDA-SWA	**3.0560**	12.12	2.5705	22.81	**2.3625**	32.31
Mamba-2-SWA	3.0845	12.96	2.5862	23.54	2.3797	30.54
LaCT	3.1088	12.36	2.5722	24.88	2.4464	31.1
`TTG`	3.0985	**17.38**	**2.5630**	**26.76**	2.4855	**34.9**

The r=d=1 label specifies repeats and depth, not a complete conditioning configuration. The 124M-family main row uses timestep embedding, per-block AdaLN, and fixed ρ=1, without conditioning sharing or modulation reuse (172.7M total parameters). Consequently, model-family comparisons and conditioning ablations with explicit parameter counts answer different questions.

Training-time write and readout trade-offs. We vary write repeats r and trained readout depth d to study how additional training compute should be allocated. At 124M and 760M, the main comparison and this sweep use different implementation cohorts, each with its own LaCT reference; conclusions are within-cohort comparisons (Appendix B). Increasing either factor is not uniformly beneficial (Fig. 4). At 124M, d=4 consistently worsens PTL, whereas r=4, d=1 gives the lowest PTL and r=1, d=2 the highest RULER accuracy. Only r=2, d=1 exceeds the sweep's LaCT reference at 760M; at 3B, several configurations improve retrieval but none matches LaCT in PTL. Thus r=d=1 is a low-cost default, not a universal optimum.

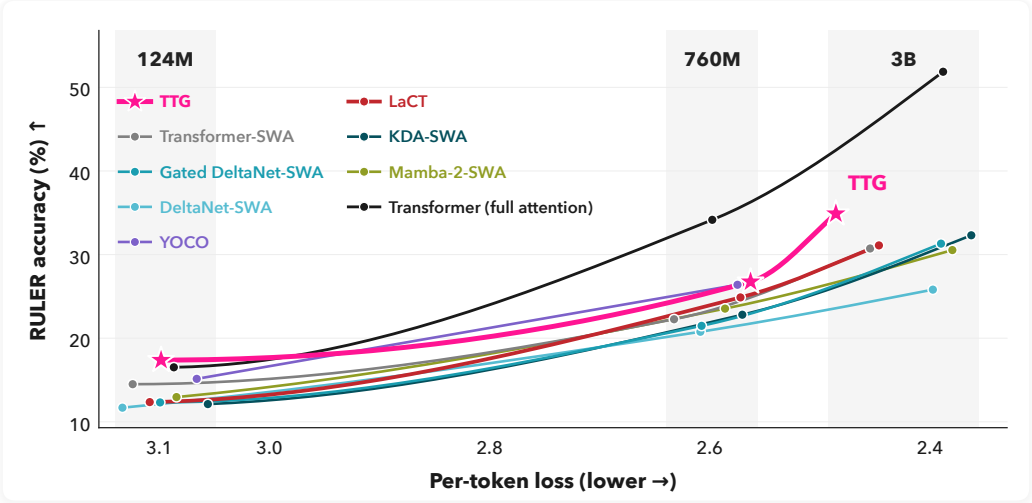


Figure 3: Pareto trade-offs across model scales. Colors distinguish model families, and markers show their available 124M, 760M, and 3B results, with pink stars denoting TTG. Smooth curves interpolate measurements within each family; shaded bands indicate the PTL range at each size. The upper right corresponds to lower loss and higher mean RULER accuracy. TTG achieves the highest mean RULER among bounded-state models at all three sizes, while loss and retrieval rank models differently. YOCO has results only at 124M and 760M.

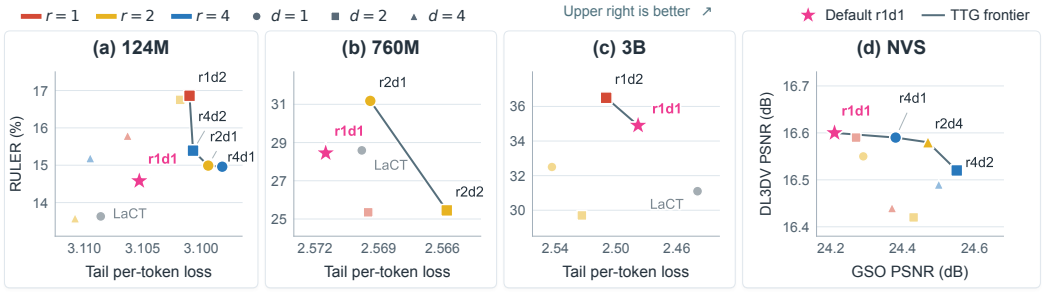


Figure 4: Pareto trade-offs across write repeats and trained readout depth. (a–c) Tail per-token loss versus mean RULER accuracy at 124M, 760M, and 3B. (d) GSO versus DL3DV PSNR. Colors encode write repeats r, and marker shapes encode trained readout depth d. Pink stars mark the default r1d1 configuration; gray points in (a–c) denote LaCT references. Lines connect nondominated configurations within each TTG sweep, while lighter points show the remaining results. Loss axes are reversed, so the upper right is better in every panel. Connecting segments serve as visual guides; the NVS frontier uses scores rounded to two decimal places.

Evaluation-time depth. We vary S for one step-10,240 checkpoint trained with $r{=}d{=}1$, using a separate Book-3 evaluation harness. Absolute PTL values are therefore not directly comparable with the main table or training sweep. $S{=}1$ improves PTL over its LaCT reference (3.1282 vs. 3.1313), whereas $S{=}2$ and $S{=}4$ worsen it to 3.1316 and 3.1395. A model trained with $r{=}1, d{=}2$ performs best at $S{=}2$ (Appendix E), suggesting that evaluation depth should match training depth. The $S{=}1$ gain may arise from the learned denoising readout rather than extra iterations; this full-model result does not establish recovery of multiple valid values.

Conditioning ablations. Per-block AdaLN improves all four metrics over additive conditioning across the four combinations of timestep embedding (TE) on/off and continuous/fixed noise level (Fig. 5). With TE enabled, shared AdaLN improves both language-model metrics over additive conditioning; per-block AdaLN improves them further. Sharing saves 32.5M language-model parameters and 9.2M NVS parameters. The 124M-family ablation spans 137.3M–172.7M total parameters (Table A2). Sharing ties conditioning parameters; reuse additionally shares the computed per-token modulation across blocks. NVS PSNR remains close after sharing, while reuse has mixed effects across datasets (Table 2(a)).

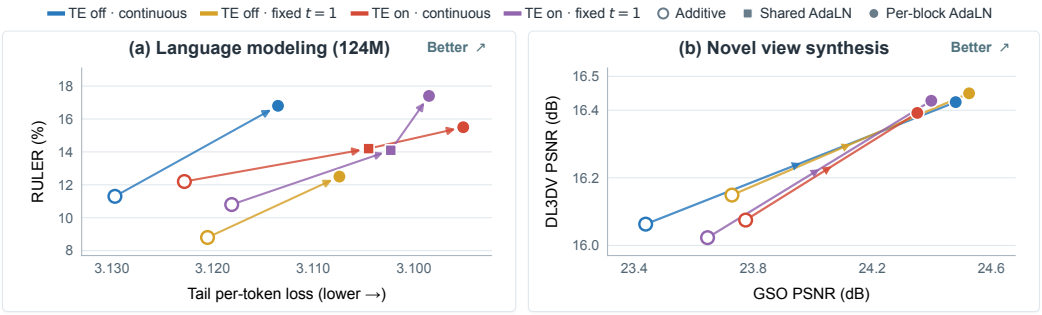


Figure 5: Conditioning ablations at $r{=}d{=}1$. (a) Tail per-token loss and RULER accuracy; (b) GSO and DL3DV PSNR. Colors encode timestep embedding (TE) and time settings (plot t denotes noise level ρ). Arrows connect additive conditioning (hollow circles) to per-block AdaLN (filled circles); the TE-on language-model paths include shared AdaLN (squares). Loss is reversed, so upper right is better.

4.3 Novel View Synthesis

Evaluation protocol. We evaluate all 1,019 GSO objects and 140 DL3DV scenes. At position $k{=}1, \ldots, 23$, the first k sampled views predict view $k{+}1$: the target changes along the curve. PSNR averages scenes and, for headline scores, all 23 positions, using one generated sample. `TTG`, DeltaNet, and Gated DeltaNet share seed-0 view sequences; archived LaCT and Transformer results use seed 9595. All cover the full test sets, but the latter comparisons are not paired by view sampling. The GSO/DL3DV `TTG` checkpoints use four/one fast-weight heads, batches 128/64, and 24K/18K training steps, respectively; Appendix D specifies their conditioning and selection.

Reconstruction quality. Figure 6 compares `TTG` with LaCT, Transformer, DeltaNet, and Gated DeltaNet. With $r{=}d{=}1$, `TTG` reaches mean PSNR of 24.93 dB on GSO and 16.59 dB on DL3DV, exceeding LaCT (23.99/16.14), DeltaNet (23.95/15.83), and Gated DeltaNet (24.33/16.05) on both datasets. The growing-cache Transformer reference remains higher at 25.36/17.41 dB. A historical sampled-online GSO sweep yields median PSNRs of 23.559, 24.785, and 24.879 dB for LaCT, energy-based, and generative readouts (Table A4). All runs with valid 24K evaluations count, including failed-but-complete runs; configurations and counts differ ($n{=}20, 22, 85$), so the comparison is descriptive. Readout effects are not isolated.

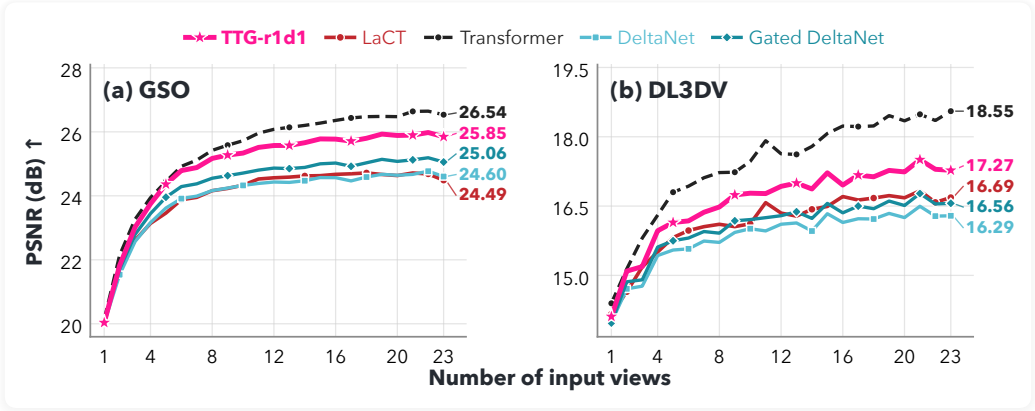


Figure 6: Per-view reconstruction quality on (a) GSO and (b) DL3DV. Curves show PSNR as input views accumulate for TTG (r=d=1), LaCT, Transformer, DeltaNet, and Gated DeltaNet. Pink stars denote TTG; endpoint labels report the 24th view predicted from 23 inputs, rather than the mean across all prediction positions.

Qualitative reconstruction. Figure 7 shows two selected GSO objects reconstructed after writing all 24 views, including observed views. This differs from sequential next-view prediction.

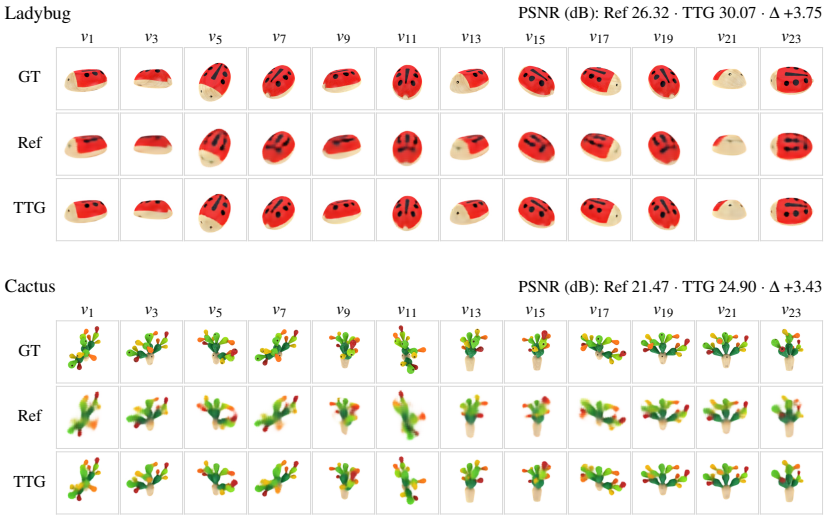


Figure 7: Full-set GSO reconstruction after writing all 24 views. Rows show ground truth (GT), deterministic reference (Ref), and TTG; columns select every other view index. Ladybug and cactus are selected high-gain examples highlighting markings and multi-part geometry, not benchmark averages. Gains of +3.75/+3.43 dB average all 24 targets, including observed views.

4.4 Controlled Two-Value Recall

To test whether the readout can recover competing bindings, we write two values per key into the same gated-MLP fast-weight memory. Values lie on the unit sphere with separation 1.732; decoded reads are likewise L_2-normalized. We compare TTG, LaCT, 4×-capacity LaCT, and a matched-noise deterministic control; N keys contain $2N$ bindings. The matched-noise control preserves TTG's squared-loss form, conditioning, widths, and denoising loop, replacing noise in both writes and reads with a learned constant. The capacity control quadruples

LaCT's fast weights at matched slow-parameter count. Coverage is the fraction of keys for which 32 reads include a sample at distance strictly below half the value separation from each stored value. TTG reaches 65.9% at N=4, S=4 and 34.4% at N=16, S=8 (Fig. 8). All deterministic controls have zero two-value coverage by construction.

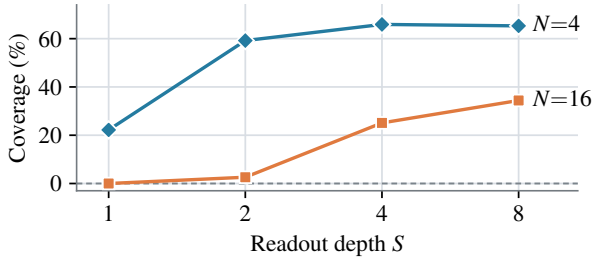


Figure 8: Two-value recall versus readout depth. Blue and orange show TTG with N=4 and N=16 keys, respectively (2N bindings). Coverage is the percentage of keys for which 32 reads recover both stored values. The dashed line marks zero coverage for deterministic controls and TTG at N=64.

Table 2: (a) NVS conditioning with TE enabled, a continuous noise schedule, and r=d=1. Params counts all model parameters; PSNR is in dB. Reuse applies per-token modulation computed in the first block to subsequent blocks. Bold marks the best PSNR per column. (b) TTG probe mixture negative log-likelihood (NLL; lower is better) at readout depths S=1 and S=8, with N keys and two values per key.

(a) NVS conditioning

Conditioning	Params (M)	GSO ↑	DL3DV ↑
Per-block AdaLN	59.3	24.351	16.392
Shared AdaLN	50.1	24.338	**16.474**
Shared + Reuse	50.1	**24.538**	16.431

(b) Probe mixture NLL

Keys N	S=1	S=8
4	57.3	10.8
16	70.8	69.9

Mixture NLL scores decoded reads under a reference two-value mixture and averages over draws and keys; it is not a likelihood assigned by a normalized TTG density. At N=4, increasing S from 1 to 8 substantially lowers NLL; at N=16, NLL changes little despite coverage rising from 0 to 34.4% (Table 2(b)). Coverage requires only one hit near each value among 32 draws, so occasional successful reads can improve coverage without improving average sample fidelity.

In the N=64, S=4 capacity check, quadrupling LaCT's fast weights moves its output closer to the normalized two-value mean (distance 0.52 → 0.42) without recovering both values. A one-value-per-key control succeeds in both families, checking that ordinary recall remains possible. Freezing noise on trained TTG weights removes all two-value coverage, as expected when repeated reads return the same output. The substantive evidence is that stochastic reads reach both valid values, beyond merely producing different outputs; these controls do not isolate denoising training as the only possible cause. At N=64, every arm has zero coverage, and at S=4, TTG's mixture NLL is about 3.5× LaCT's: generative reading does not overcome write-side interference.

5 Conclusion

We introduce TTG, a framework for reading adaptive fast-weight memory through conditional generation. Our motivation is that a fixed-size state may preserve multiple plausible values, while a deterministic regression readout can collapse them into a single prediction. TTG addresses this limitation by combining online fast-weight updates with a stochastic, query-conditioned readout. Controlled probes and experiments in long-context language modeling and novel view synthesis provide evidence for improved two-value recall, long-context retrieval, and reconstruction in the tested settings, while also revealing compute and capacity limits. Within the TTT setting, TTG pairs conditional generation with a bounded fast-weight state that updates online while slow parameters remain fixed. We hope this perspective inspires future work on long-context modeling and other settings that require test-time adaptation, including embodied agents and video world models.

References

Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions. *Journal of Machine Learning Research*, 26 (209):1–80, 2025. URL https://jmlr.org/papers/v26/23-1605.html.

Luca Ambrogioni. In Search of Dispersed Memories: Generative Diffusion Models Are Associative Memory Networks. *Entropy*, 26(5):381, 2024. doi: 10.3390/e26050381. URL https://doi.org/10.3390/e26050381.

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, and Se-Young Yun. Mixture-of-Recursions: Learning Dynamic Recursive Depths for Adaptive Token-Level Computation, July 2025. URL https://arxiv.org/abs/2507.10524.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to Memorize at Test Time, December 2024. URL https://arxiv.org/abs/2501.00663.

Ali Behrouz, Zeman Li, Praneeth Kacham, Majid Daliri, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. ATLAS: Learning to Optimally Memorize the Context at Test Time, May 2025a. URL https://arxiv.org/abs/2505.23735.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. Nested Learning: The Illusion of Deep Learning Architectures. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b. doi: 10.48550/arXiv.2512.24695. URL https://arxiv.org/abs/2512.24695.

Quentin Bertrand, Anne Gagneux, Mathurin Massias, and Rémi Emonet. On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity, 2025. URL https://arxiv.org/abs/2506.03719.

Christopher M. Bishop. Mixture density networks. Technical Report 4288, Aston University, 1994. URL https://research.aston.ac.uk/en/publications/mixture-density-networks.

Sergio Calvo-Ordonez, Matthieu Meunier, Alvaro Cartea, Christoph Reisinger, Yarin Gal, and Jose Miguel Hernandez-Lobato. Weighted Conditional Flow Matching, 2025. URL https://arxiv.org/abs/2507.22270.

Ho Kei Cheng and Alexander Schwing. The Curse of Conditions: Analyzing and Improving Optimal Transport for Conditional Flow-Based Generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. doi: 10.48550/arXiv.2503.10636. URL https://arxiv.org/abs/2503.10636.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 10041–10071, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal Transformers. In *International Conference on Learning Representations (ICLR)*, 2019. doi: 10.48550/arXiv.1807.03819. URL https://arxiv.org/abs/1807.03819.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3D objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153, 2023. doi: 10.1109/CVPR52729.2023.01263. URL https://arxiv.org/abs/2212.08051.

Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google Scanned Objects: A high-quality dataset of 3D scanned household items. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2022. doi: 10.1109/ICRA46639.2022.9811809. URL https://arxiv.org/abs/2204.11918.

Yilun Du and Igor Mordatch. Implicit Generation and Modeling with Energy Based Models, 2019. URL https://arxiv.org/abs/1903.08689.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Will Tennien, Atri Rudra, James Zou, Azalia Mirhoseini, and Christopher Re. Cartridges: Lightweight and general-purpose long context representations via self-study, June 2025. URL https://arxiv.org/abs/2506.06266.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling, 2020. URL https://arxiv.org/abs/2101.00027.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach, February 2025. URL https://arxiv.org/abs/2502.05171.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer Feed-Forward Layers Are Key-Value Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. URL https://aclanthology.org/2021.emnlp-main.446/.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped Transformers as Programmable Computers, January 2023. URL https://arxiv.org/abs/2301.13196.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, May 2024. URL https://arxiv.org/abs/2312.00752.

Demis Hassabis, Dharshan Kumaran, and Eleanor A. Maguire. Using Imagination to Understand the Neural Basis of Episodic Memory. *Journal of Neuroscience*, 27(52):14365–14374, 2007. doi: 10.1523/JNEUROSCI.4549-07.2007. URL https://www.jneurosci.org/content/27/52/14365.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, December 2020. URL https://arxiv.org/abs/2006.11239.

Benjamin Hoover, Hendrik Strobelt, Dmitry Krotov, Judy Hoffman, Zsolt Kira, and Duen Horng Chau. Memory in Plain Sight: Surveying the Uncanny Resemblances of Associative Memories and Diffusion Models, 2023. URL https://arxiv.org/abs/2309.16750.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the Real Context Size of Your Long-Context Language Models?, 2024. URL https://arxiv.org/abs/2404.06654.

Jinwu Hu, Zhitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-Time Learning for Large Language Models. In *International Conference on Machine Learning (ICML)*, 2025. doi: 10.48550/arXiv.2505.20633. URL https://arxiv.org/abs/2505.20633.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A Large View Synthesis Model with Minimal 3D Inductive Bias, April 2025. URL https://arxiv.org/abs/2410.17242.

Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models, 2025. URL https://arxiv.org/abs/2412.20292.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the Design Space of Diffusion-Based Generative Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. doi: 10.48550/arXiv.2206.00364. URL https://arxiv.org/abs/2206.00364.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Kimi Team, Yu Zhang, Zongyu Lin, Xingcheng Yao, Jiaxi Hu, Fanqing Meng, Chengyin Liu, Xin Men, Songlin Yang, Zhiyuan Li, Wentao Li, Enzhe Lu, Weizhou Liu, Yanru Chen, Weixin Xu, Longhui Yu, Yejie Wang, Yu Fan, Longguang Zhong, Enming Yuan, Dehao Zhang, Yizhi Zhang, T. Y. Liu, Haiming Wang, Shengjun Fang, Weiran He, Shaowei Liu, Yiwei Li, Jianlin Su, Jiezhong Qiu, Bo Pang, Junjie Yan, Zhejun Jiang, Weixiao Huang, Bohong Yin, Jiacheng You, Chu Wei, Zhengtao Wang, Chao Hong, Yutian Chen, Guanduo Chen, Yucheng Wang, Huabin Zheng, Feng Wang, Yibo Liu, Mengnan Dong, Zheng Zhang, Siyuan Pan, Wenhao Wu, Yuhao Wu, Longyu Guan, Jiawen Tao, Guohong Fu, Xinran Xu, Yuzhi Wang, Guokun Lai, Yuxin Wu, Xinyu Zhou, Zhilin Yang, and Yulun Du. Kimi Linear: An Expressive, Efficient Attention Architecture, 2025. URL https://arxiv.org/abs/2510.26692.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A Tutorial on Energy-Based Learning. In *Predicting Structured Data*. MIT Press, 2006.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision, 2023. URL https://arxiv.org/abs/2312.16256.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, February 2023. URL https://arxiv.org/abs/2210.02747.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow, 2022. URL https://arxiv.org/abs/2209.03003.

Karim Nader, Glenn E. Schafe, and Joseph E. Le Doux. Fear memories require protein synthesis in the amygdala for reconsolidation after retrieval. *Nature*, 406(6797):722–726, 2000. doi: 10.1038/35021052. URL https://www.nature.com/articles/35021052.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield Networks is All You Need, 2021. URL https://arxiv.org/abs/2008.02217.

Daniel L. Schacter, Kenneth A. Norman, and Wilma Koutstaal. The Cognitive Neuroscience of Constructive Memory. *Annual Review of Psychology*, 49:289–318, 1998. doi: 10.1146/annurev.psych.49.1.289. URL https://www.annualreviews.org/doi/10.1146/annurev.psych.49.1.289.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear Transformers Are Secretly Fast Weight Programmers. In *International Conference on Machine Learning (ICML)*, June 2021. doi: 10.48550/arXiv.2102.11174. URL https://arxiv.org/abs/2102.11174.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems 28*, 2015.

Yang Song, Jascha Sohl-Dickstein, Durk P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations (ICLR)*, 2021. doi: 10.48550/arXiv.2011.13456. URL https://arxiv.org/abs/2011.13456.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (Learn at Test Time): RNNs with Expressive Hidden States, August 2025. URL https://arxiv.org/abs/2407.04620.

Yutao Sun, Li Dong, Yi Zhu, Shaohan Huang, Wenhui Wang, Shuming Ma, Quanlu Zhang, Jianyong Wang, and Furu Wei. You Only Cache Once: Decoder-Decoder Architectures for Language Models, 2024. URL https://arxiv.org/abs/2405.05254.

Arnuv Tandon, Karan Dalal, Xinhao Li, Daniel Koceja, Marcel Rød, Sam Buchanan, Xiaolong Wang, Jure Leskovec, Sanmi Koyejo, Tatsunori Hashimoto, Carlos Guestrin, Jed McCaleb, Yejin Choi, and Yu Sun. End-to-End Test-Time Training for Long Context, December 2025. URL https://arxiv.org/abs/2512.23675.

Together AI. Long-Data-Collections. https://huggingface.co/datasets/togethercomputer/Long-Data-Collections, 2023. Hugging Face dataset.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport. *Transactions on Machine Learning Research*, 2024. doi: 10.48550/arXiv.2302.00482. URL https://arxiv.org/abs/2302.00482.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1706.03762.

Pascal Vincent. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7):1661–1674, July 2011. ISSN 0899-7667, 1530-888X. doi: 10.1162/NECO_a_00142. URL https://direct.mit.edu/neco/article/23/7/1661-1674/7677.

Zhengchao Wan, Qingsong Wang, Gal Mishne, and Yusu Wang. Elucidating Flow Matching ODE Dynamics with Respect to Data Geometries and Denoisers. In *International Conference on Machine Learning (ICML)*, 2025. doi: 10.48550/arXiv.2412.18730. URL https://arxiv.org/abs/2412.18730.

Jian Yang, Shawn Guo, Wei Zhang, Tianyu Zheng, Yaxin Du, Haau-Sing Li, Jiajun Wu, Yue Song, Yan Xing, Qingsong Cai, Zelong Huang, Chuan Hao, Ran Tao, Xianglong Liu, Wayne Xin Zhao, Mingjie Tang, Weifeng Lv, Ming Zhou, and Bryan Dai. LoopCoder-v2: Only Loop Once for Efficient Test-Time Computation Scaling, June 2026. URL https://arxiv.org/abs/2606.18023.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated Linear Attention Transformers with Hardware-Efficient Training, December 2023. URL https://arxiv.org/abs/2312.06635.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated Delta Networks: Improving Mamba2 with Delta Rule, December 2024a. URL https://arxiv.org/abs/2412.06464.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing Linear Transformers with the Delta Rule over Sequence Length, 2024b. URL https://arxiv.org/abs/2406.06484.

Michael A. Yassa and Craig E. L. Stark. Pattern Separation in the Hippocampus. *Trends in Neurosciences*, 34(10):515–525, 2011. doi: 10.1016/j.tins.2011.06.006. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3183227/.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T. Freeman, and Hao Tan. Test-Time Training Done Right, May 2025. URL https://arxiv.org/abs/2505.23884.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, Lu Li, Jiajun Shi, Kaijing Ma, Shanda Li, Taylor Kergan, Andrew Smith, Xingwei Qu, Mude Hui, Bohong Wu, Qiyang Min, Hongzhi Huang, Xun Zhou, Wei Ye, Jiaheng Liu, Jian Yang, Yunfeng Shi, Chenghua Lin, Enduo Zhao, Tianle Cai, Ge Zhang, Wenhao Huang, Yoshua Bengio, and Jason Eshraghian. Scaling Latent Reasoning via Looped Language Models, October 2025. URL https://arxiv.org/abs/2510.25741.

A Additional Long-Context Results

We provide position-wise diagnostics for long-context language modeling. Per-view NVS results appear in Fig. 6.

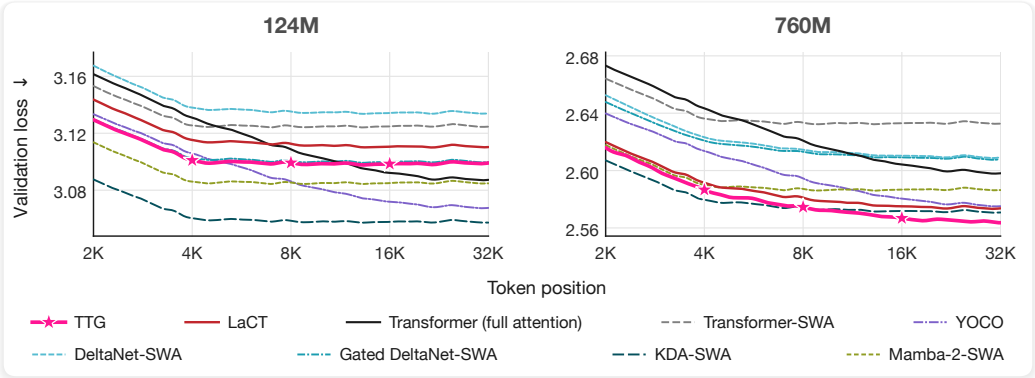


Figure A1: Per-token validation loss of TTG and eight baselines at 124M and 760M. The 760M DeltaNet-SWA and Gated DeltaNet-SWA curves are recovered, previously smoothed plot samples from the uTTT archive; the other 16 curves use complete numerical evaluation series. All inputs receive the same additional local smoothing for display on a logarithmic token-position axis from 2K to 32K. Recovered curves are not used to recompute scalar PTL. Stars highlight TTG; lower is better, with separate loss ranges for each size.

Table A1: Complete results for Table 1. PTL reports archived Book-3 tail-loss summaries (aggregation conventions below); RULER is 13-task retrieval accuracy (%) at each length. Bold marks the best bounded-state value; † denotes an unbounded-cache reference.

Model	PTL ↓	RULER ↑	4K	8K	16K	32K
124M						
Transformer (full attention) †	3.0869	16.52	23.82	20.24	13.10	8.93
YOCO †	3.0661	15.14	22.70	17.95	11.40	8.53
Transformer-SWA	3.1243	14.50	26.93	14.73	9.94	6.41
Gated DeltaNet-SWA	3.0994	12.32	23.78	12.93	8.30	4.26
DeltaNet-SWA	3.1336	11.69	22.52	11.23	7.63	5.36
KDA-SWA	**3.0560**	12.12	20.86	12.98	8.36	6.28
Mamba-2-SWA	3.0845	12.96	24.21	13.93	8.32	5.37
LaCT	3.1088	12.36	24.34	14.09	6.97	4.06
TTG (r=d=1)	3.0985	**17.38**	**28.36**	**19.80**	**13.50**	**7.84**
760M						
Transformer (full attention) †	2.5978	34.16	41.70	37.95	33.36	23.62
YOCO †	2.5749	26.39	35.81	28.12	23.17	18.45
Transformer-SWA	2.6325	22.27	45.23	22.64	12.67	8.57
Gated DeltaNet-SWA	2.6075	21.48	40.40	23.12	13.66	8.76
DeltaNet-SWA	2.6086	20.77	43.43	20.56	11.79	7.28
KDA-SWA	2.5705	22.81	43.47	24.17	14.65	8.96
Mamba-2-SWA	2.5862	23.54	**48.42**	25.49	12.60	7.65
LaCT	2.5722	24.88	46.67	27.55	15.85	9.45
TTG (r=d=1)	**2.5630**	**26.76**	43.96	**28.99**	**19.20**	**14.89**
3B						
Transformer (full attention) †	2.3881	51.87	59.10	55.71	52.20	40.48
Transformer-SWA	2.4544	30.73	64.11	29.48	18.11	11.22
Gated DeltaNet-SWA	2.3900	31.31	61.76	33.30	18.54	11.63
DeltaNet-SWA	2.3974	25.81	51.62	26.92	15.04	9.66
KDA-SWA	**2.3625**	32.31	**64.71**	32.53	18.77	13.24
Mamba-2-SWA	2.3797	30.54	61.45	32.71	17.58	10.39
LaCT	2.4464	31.1	55.4	37.8	20.5	10.7
TTG (r=d=1)	2.4855	**34.9**	59.1	**38.3**	**25.7**	**16.3**

We retain the scalar PTL values reported by each archived evaluation. Some legacy summaries average from zero-based index 29,999 and include a terminal zero placeholder, whereas the position-wise curves exclude invalid terminal entries. Their tail means therefore need not reproduce every historical scalar exactly.

B Complete Ablation Tables

Tables A2 and A3 give the complete conditioning and write/readout sweeps underlying Figs. 5 and 4.

At 124M and 760M, the main comparison and the write/depth sweep use separate archived implementation cohorts. At 124M, the main TTG row is the v5 conditioning-ablation checkpoint (TE on, per-block AdaLN, fixed noise level), paired with the batched dense-v1 LaCT reference; the sweep uses batched diffusion-v1 and dense-v4. At 760M, the main comparison uses diffusion-v2 and LaCT ref-v1, whereas the sweep uses diffusion-v1 and ref-v4. Thus the 760M main-table TTG/LaCT RULER pair (26.76/24.88) and sweep r1d1/reference pair (28.45/28.59) are different comparisons, not contradictory outcomes for one configuration. Version labels distinguish implementation cohorts and do not denote random seeds.

Table A2: Complete conditioning ablation underlying Fig. 5. All rows use r=d=1. TE is the timestep embedding, Mod is AdaLN conditioning, Sch is the continuous noise schedule (off fixes $\rho \equiv 1$, labeled $t \equiv 1$ in the plots), Share reuses one conditioning stack across blocks, and Reuse computes per-token modulation only in the first block. Language modeling uses a 124M model and reports deep-context PTL and RULER accuracy (%); NVS reports PSNR on DL3DV and GSO. Params is the complete model size in millions. Bold marks the best PTL, RULER, and PSNR values across all rows.

					Language modeling (124M)							Novel view synthesis		
					Params	PTL ↓	RULER ↑					Params	PSNR ↑	
TE	Mod	Sch	Share	Reuse	(M)	30k+	4k	8k	16k	32k	avg.	(M)	DL3DV	GSO
Additive conditioning														
		✓			137.3	3.1297	19.6	12.9	8.2	4.3	11.3	48.8	16.063	23.438
					137.3	3.1205	15.5	11.9	4.9	2.9	8.8	48.8	16.149	23.728
✓		✓			138.2	3.1228	21.8	13.6	8.9	4.5	12.2	53.0	16.075	23.774
✓					138.2	3.1181	23.2	10.3	6.1	3.8	10.8	53.0	16.023	23.646
AdaLN, per-block conditioning														
	✓	✓			158.5	3.1135	**29.5**	**20.1**	12.1	5.6	16.8	55.1	16.424	24.480
	✓				158.5	3.1074	20.8	14.5	9.7	5.2	12.5	55.1	16.450	24.525
✓	✓	✓			172.7	**3.0951**	26.9	16.3	11.5	7.3	15.5	59.3	16.392	24.351
✓	✓				172.7	3.0985	28.4	19.8	**13.5**	**7.8**	**17.4**	59.3	16.428	24.398
AdaLN, shared conditioning														
✓	✓	✓	✓		140.2	3.1045	26.5	14.5	9.7	6.1	14.2	50.1	**16.474**	24.338
✓	✓		✓		140.2	3.1023	23.8	15.7	10.8	6.2	14.1	50.1	16.459	24.389
✓	✓	✓	✓	✓	140.2	3.1002	25.5	13.9	8.5	5.3	13.3	50.1	16.431	**24.538**
✓	✓		✓	✓	140.2	3.1023	28.0	15.8	10.0	6.3	15.0	50.1	16.471	24.308

Across the six pairs in Table A2, fixing $\rho \equiv 1$ lowers language-model loss in four pairs but improves retrieval in only two. This contrasts with the consistent gains from per-block AdaLN in Fig. 5.

Table A3: Complete write-repeat r and trained-depth d sweep underlying Fig. 4. Language-model cells report deep-context PTL (lower is better) and RULER accuracy in % (higher is better); the NVS cells report PSNR in dB. Bold marks the best TTG result per metric within each block. LaCT reference values appear in the language-model block headers.

	d=1		d=2		d=4	
	PTL ↓	RULER ↑	PTL ↓	RULER ↑	PTL ↓	RULER ↑
124M; LaCT 3.1086 / 13.63						
r=1	3.1053	14.58	3.1010	**16.86**	3.1063	15.79
r=2	3.0994	14.99	3.1018	16.75	3.1108	13.58
r=4	**3.0982**	14.96	3.1007	15.39	3.1095	15.19
760M; LaCT 2.5698 / 28.59						
r=1	2.5715	28.45	2.5695	25.35	—	—
r=2	2.5694	**31.18**	**2.5658**	25.45	—	—
3B; LaCT 2.4464 / 31.1						
r=1	**2.4855**	34.9	2.5065	**36.5**	—	—
r=2	2.5422	32.5	2.5225	29.7	—	—
NVS 59M, PSNR in dB						
	GSO PSNR ↑	DL3DV PSNR ↑	GSO PSNR ↑	DL3DV PSNR ↑	GSO PSNR ↑	DL3DV PSNR ↑
r=1	24.21	**16.60**	24.27	16.59	24.37	16.44
r=2	24.29	16.55	24.43	16.42	24.47	16.58
r=4	24.38	16.59	**24.55**	16.52	24.50	16.49

C Bridge Identities and Retrieval-Margin Derivation

This section records the standard denoising identities and the algebra behind Eqs. (2) and (3). All statements concern the idealized discrete mixture in (2); they are background specialized to our notation, not additional contributions.

Lemma C.1 (Bayes-optimal denoiser and velocity; background)**.** *Under* (2)*, the Bayes-optimal denoiser is the conditional mean* $m_t(x, k; h)$*. The linear bridge has constant target velocity* $v - \varepsilon$*, so its optimal velocity is* $\big(m_t(x, k; h) - x\big)/(1 - t)$*.*

Observation C.2 (Conditional-mean solution; background)**.** *Squared loss on* (K, V) *decomposes:*

$$\mathbb{E}[\|f(K) - V\|_2^2 \mid K{=}k] = \|f(k) - \mathbb{E}[V \mid K{=}k]\|_2^2 + \operatorname{tr}\operatorname{Var}(V \mid K{=}k).$$

The second term is independent of f*, so* $f^\star(k) = \mathbb{E}[V \mid K{=}k]$ *pointwise: deterministic one-shot regression averages non-degenerate bindings instead of representing their conditional distribution.*

Observation C.3 (Shared-to-specific retrieval dynamics; background)**.** *Fix* k, h *and* $0 \le t < 1$*, with finite support and positive normalized priors. Let* $\ell_i = -\|x - \beta_t v_i\|^2/(2\alpha_t^2)$*. If* $\max_i \ell_i - \min_i \ell_i \le \delta$*, then*

$$e^{-\delta} \le \frac{w_i}{\pi_i(k; h)} \le e^{\delta}, \qquad \left\| m_t - \sum_i \pi_i(k; h) v_i \right\| \le (e^{\delta} - 1) \sum_i \pi_i(k; h) \|v_i\|.$$

Thus small likelihood contrast at the current x *makes the posterior mean close to the prior mean. At* $t = 0$*, the weights equal the priors exactly. For distinct retained values and bridge samples conditioned on component* i*,* $w_i \to 1$ *in probability as* $t \uparrow 1$*; this does not assert monotone sharpening along arbitrary inputs or trained-model trajectories.*

Proof. The normalizer $\sum_j \pi_j e^{\ell_j}$ lies between $e^{\min_j \ell_j}$ and $e^{\max_j \ell_j}$, giving the ratio bound; the mean bound follows by the triangle inequality. For $x_t = \beta_t v_i + \alpha_t \varepsilon$ and $j \neq i$, $\log(w_j/w_i)$ has a negative quadratic term $-\beta_t^2 \|v_i - v_j\|^2/(2\alpha_t^2)$ and a Gaussian term of order β_t/α_t. The former dominates as $t \uparrow 1$, yielding concentration on component i. □

These are standard discrete-mixture denoising/score and conditional-flow-matching identities (Vincent, 2011; Lipman et al., 2023; Tong et al., 2024; Ambrogioni, 2024; Hoover et al., 2023; Bertrand et al., 2025), the squared-loss solution (Bishop, 1994; Sohn et al., 2015), and coarse-to-fine denoising (Wan et al., 2025; Karras et al., 2022; Kamb & Ganguli, 2025).

Proposition C.4 (Retrieval margin and denoising SNR)**.** *For* $0 \le t < 1$ *and* $\pi_i, \pi_j > 0$*, let* $\Delta_{ij} = \log(w_i/w_j)$*. Canceling the shared normalization in* (2) *gives*

$$\Delta_{ij} = \log \frac{\pi_i(k; h)}{\pi_j(k; h)} - \frac{\|x - \beta_t v_i\|^2 - \|x - \beta_t v_j\|^2}{2\alpha_t^2}.$$

Substituting $x_t = \beta_t v_i + \alpha_t \varepsilon$*, with* $\varepsilon \sim \mathcal{N}(0, I)$*, yields a cross term* $(\beta_t/\alpha_t)\langle v_i - v_j, \varepsilon\rangle$ *of zero mean and therefore Eq.* (3)*. Its standard deviation is* $(\beta_t/\alpha_t)\|v_i - v_j\|$*, so for* $0 < t < 1$*, distinct values, and an uninformative prior the margin signal-to-noise ratio is* $(\beta_t/2\alpha_t)\|v_i - v_j\|$*.*

Proposition C.4 separates a key-prior term available to any readout from a value-space term amplified by bridge SNR and value separation. It does not guarantee recovery in a trained model: a recurrent state only approximates the idealized mixture, and $\pi_i(k; h)$ degrades as bindings compete for fixed capacity. The probe of Section 4.4 makes that limit explicit. At fixed separation $\|v_A - v_B\| = 1.732$, the generative readout recovers both bindings on 0.66 of keys at load $N{=}4$ and 0.34 at $N{=}16$ at their best depths, against 0.00 for every deterministic arm; at the highest load every tested arm fails. Finally, the Gaussian bridge, the identities above, and the margin do not cover the energy-based readout (LeCun et al., 2006; Du & Mordatch, 2019); they motivate our default flow-matching realization rather than define the readout interface.

D Additional NVS Results

The main comparison uses separate r=d=1 checkpoints for GSO and DL3DV, selected using prior sampled evaluations and checkpoint availability before full-test evaluation. Both use RGB MSE supervision (LPIPS weight zero), 256×256 images, 8×8 patches, and continuous training timesteps. GSO uses four fast-weight heads and shared modulation (50.06M parameters, batch 128, step 24,000); DL3DV uses one head with unshared modulation and the second input variant (59.25M parameters, batch 64, step 18,000). These are dataset-specific configurations, not a matched architectural comparison between datasets. The qualitative examples use separate full-set reconstruction runs, whereas the main curves predict each next view from the preceding views.

Table A4: Historical GSO online-evaluation sweep. We retain the model-block families in the original comparison and include finite PSNR measured at 24K forward/backward passes, irrespective of W&B status. These are sampled online evaluations, not the full-test results in Fig. 6. Counts and configurations differ across families. Bold marks the largest PSNR in each column.

Readout family	n	Min	Median	Max
Deterministic (LaCT) readout	20	13.752	23.559	24.330
Energy-based readout	22	**24.542**	24.785	24.993
`TTG` (generative denoising)	85	24.066	**24.879**	**25.153**

The sweep includes 11 LaCT runs with fewer successful optimizer updates than forward/backward passes; all included Energy and generative runs have 24K successful updates at evaluation. Hence equal attempted-pass counts do not imply matched effective training budgets. The run-level manifest records inclusion criteria and update counts. Across 15 separately archived identical-configuration pairs, the median PSNR spread is 0.035 dB and the maximum is 0.165 dB. Four `TTG` configurations differ by at most 0.11 dB between reruns. These spreads describe observed variation and do not provide confidence intervals for the headline comparisons. In the broader sweep in Table A4, the worst generative run (24.066) remains below the best deterministic run (24.330), although no deterministic run reaches the generative median (24.879). Two separately archived configurations each of our DeltaNet and Gated DeltaNet adaptations (Yang et al., 2024b;a) of the LVSM-style backbone (Jin et al., 2025) obtain 23.952/24.729 and 24.192/24.863 dB, respectively, under historical online evaluation; the best archived growing-cache attention run reaches 26.105 dB. These are our measurements, not scores reported by the cited architecture papers.

E Analysis Details

A model trained with r=1, d=2 achieves its lowest PTL at evaluation depth S=2, with off-depth penalties below 0.003. Together with the d=1 result, this suggests that the optimum tracks the depth used in training. Iterated-depth language models show a similar early optimum (Yang et al., 2026).

The two-value probe writes two unit-sphere values with separation $\|v_A - v_B\| = 1.732$ per key into the same gated-MLP memory for every arm. Coverage is measured over 32 draws per key. `cover_both` requires at least one of the 32 reads at distance strictly below $\frac{1}{2}\|v_A - v_B\| = \sqrt{3}/2 \approx 0.866$ from each stored value; the two reads may differ, while a mean-collapsed output cannot count as success. N controls the number of keys and hence interference. The matched-noise control shares `TTG`'s squared-loss form, conditioning, widths, and denoising loop, but substitutes a learned constant for noise in both write corruption and readout. This changes the training corruption distribution; freezing readout noise on an already trained `TTG` model instead intervenes only at read time. The capacity control gives LaCT $4\times$ fast weights at matched slow-parameter count.

Reference-mixture score. For normalized decoded reads $x \in \mathbb{R}^{64}$, the archived evaluator computes

$$\mathrm{NLL}(x) = -\log\left[\tfrac{1}{2}\mathcal{N}(x; v_A, \sigma_{\mathrm{eff}}^2 I_{64}) + \tfrac{1}{2}\mathcal{N}(x; v_B, \sigma_{\mathrm{eff}}^2 I_{64})\right].$$

The logarithm is natural; scores are averaged over draws, keys, and evaluation contexts, then over training seeds, without dividing by dimension. This is an ambient-space reference-mixture score, not a normalized density of the learned sampler or a spherical likelihood. Let $n(z) = z/(\|z\|_2 + 10^{-5})$. With seed 999, the code generates $M = 200{,}000$ independent Gaussian pairs $z_j, \xi_j \in \mathbb{R}^{64}$, sets $c_j = n(z_j)$ and $u_j = n(c_j + 0.05\xi_j)$, and uses $\sigma_{\mathrm{eff}}^2 = (64M)^{-1}\sum_j \|u_j - c_j\|^2$. This gives $\sigma_{\mathrm{eff}} \approx 0.0470$. The recovered evaluator and original aggregation records identify the displayed coverage as `cover_both@0.5`.

Within each load the arms share seeds and training budgets; $N = 4, 16, 64$ use 2/1/3 seeds and 10K/15K/15K training steps, respectively. The lower loads were added after observing zero coverage at $N = 64$, so cross-load trends are descriptive.

Table A5: Complete two-value recall results underlying Fig. 8. Coverage is the fraction of keys for which 32 reads recover both stored values. Bold marks the best coverage per load; all-zero rows are left unhighlighted. "Det. arms" denotes the deterministic controls; mixture NLL is available at the displayed endpoint depths for N=4, 16.

N	S=1	2	4	8	Det. arms	NLL S=1→8
4	0.222	0.592	**0.659**	0.653	0.000	57.3 → 10.8
16	0.000	0.026	0.251	**0.344**	0.000	70.8 → 69.9
64	0.000	0.000	0.000	0.000	0.000	—

The deterministic controls never cover both values. At N=64, S=4, extra capacity moves the output closer to the normalized two-value mean (0.42 away, versus 0.52 for LaCT), while the distance to the nearer target changes from 0.94 to 0.92; neither arm recovers both values. A one-value-per-key control succeeds in both families, showing that ordinary recall remains possible at the same key count. At N=4, S=4, reusing one readout-noise vector across keys and draws within each evaluation minibatch reduces `cover_both@0.25` from 0.007 to zero; the shared vector changes between minibatches. At the half-separation threshold, at-least-one-mode coverage over 32 draws (`cover_any@0.5`) falls from 0.997 to 0.646. At N=16, increasing S from 1 to 8 expands the draw spread from 0.06 to 0.61, moves it off the mean direction (0.610 → 0.942), and raises coverage from 0.000 to 0.344. At N=64, however, no arm recovers both bindings and at S=4, `TTG`'s mixture NLL is about 3.5× LaCT's, marking write-side interference as the limiting regime.

The probe leaves one prediction untested. Equation (3) combines a key-prior term with a value-space term. Nearly colliding keys suppress the first, so the generative advantage should be larger for colliding than for well-separated keys. Because the current keys are sampled i.i.d., a direct sweep over paired-key angle remains future work.