

One Layer, Both Gaps: A Persistent-Modulation Recurrence that Generalizes Copy *and* Tracks Non-Solvable Group State

Nicholas Rosdahl
Independent Researcher
nicholasrosdahl@yahoo.com

Abstract

Two failure modes are routinely used to separate sequence-architecture families. On the **copy** task, attention re-reads its input losslessly while any fixed-width recurrence is bandwidth-bound [1]. On non-solvable group word problems such as **A5** — NC^1 -complete state tracking [2] — a genuine sequential nonlinear state is essential: constant-depth transformers learn only length-bounded shortcuts [3], and diagonal selective state-space models are provably confined to TC^0 [4]. The two gaps pull in opposite directions and are commonly taken to imply that no single recurrent layer can hold both. We give a counterexample at the level of one block design. DABSN, the converged design of the Adaptive Bias State Network family, stores memory as persistent modulation of a per-unit operating baseline and exposes, over a single shared content score, three retrieval primitives plus a nonlinear consolidation recurrence — each behind a learned scalar gain initialized so the apparatus begins as a no-op. Trained separately per task with no structural change, instances (a) generalize copy to $50\times$ training length (0.961 ± 0.035 accuracy at length 3200 over three seeds, where a RoPE transformer collapses to 0.049 at $2\times$), and (b) track the standard A5/60 word problem from length 256 to length 16,384 ($64\times$) with perfect accuracy over two seeds, while an LSTM trained on the same protocol decays to 0.139 and RoPE attention is at chance by length 2048. Removing only the recurrence’s nonlinear state feedback — the input-controlled TC^0 regime, same block and parameter budget — collapses A5 to chance even at the training length, causally isolating the nonlinear state as the mechanism. Reading the converged diagnostics shows the optimizer selects the appropriate primitive within each training run, the off-task pathway idling. The same train-short/evaluate-long flatness recurs across multi-query associative recall, key-value retrieval, burst-anomaly detection, and character-level language modeling. We argue persistent-modulation memory is a third substrate, not a recurrence-attention compromise; seed counts and scope are stated per claim.

1 Introduction

The dominant sequence architectures are separated by two complementary, theoretically grounded gaps.

The **copy** task — emit the next token of a previously seen stream — is the canonical case where attention beats recurrence: a transformer attends back to the matching position and re-reads it losslessly, so its accuracy is independent of stream length, whereas a recurrence of width H must compress the entire history into H numbers and degrades once the content exceeds that budget [1]. Copy is the *memory-bandwidth* corner.

The **A5 word problem** — maintain the running product in a non-solvable permutation group — is the canonical case where recurrence beats attention: it is NC^1 -complete [2], so a constant-depth, log-precision transformer (in TC^0) cannot express it at arbitrary length and instead learns shortcuts that hold only up to the training length [3]. Crucially, *diagonal/ selective* state-space

models such as Mamba [5] are also confined to TC^0 and provably cannot track non-solvable state at growing length [4]. This is the *state-tracking* corner.

These two gaps pull in opposite directions — copy rewards lossless re-reading, group word problems reward a genuine sequential nonlinear state — and are commonly used to argue that the field faces an inherent trade-off: a layer good at one is structurally bad at the other.

Contribution. We present a single recurrent block — one architecture, trained separately per task with no per-task structural change — that closes *both* gaps:

1. **Architecture (§3).** DABSN stores memory as persistent modulation of a per-unit bias baseline, separating the expressed activation y_t from a persistent bias state b_t . Its readout computes one shared content score and exposes three associative primitives — *content*, *induction* (gather the successor of the matched write), and *position-from-association* — plus a re-attached nonlinear diagonal recurrence. Each pathway has a learned scalar gain initialized so the apparatus begins as a no-op.
2. **Both gaps, one block design (§5.1–§5.3).** Trained on copy, the block generalizes to $50\times$ training length (three seeds) via the induction read with the recurrence idle; trained on the standard A5/60 word problem, the *same block design* tracks to $64\times$ training length. These are separately trained instances of one identical design, not one checkpoint doing both at once. We read the mechanism directly off the learned gains — in each separately trained model the off-task pathway’s gain goes to ≈ 0 — and verify it causally by eval-time knockout: zeroing the induction gain sends a copy-trained checkpoint to exact chance at every length, while zeroing its other pathways changes nothing.
3. **Length generalization as an emergent property (§5.4).** The same train-short/evaluate-long flatness appears across MQAR, key–value recall, burst-anomaly detection, and character-level language modeling, where a RoPE transformer cliffs and gated recurrences sit near chance — supporting the claim that length-robustness is a property of the substrate, not a per-task trick.

We are explicit about scope (§7): comparisons follow an each-model-at-its-own-best-config protocol, several rows are single-seed, and the A5/Mamba separation is theorem-backed rather than measured head-to-head here.

2 Related Work

State-space models and state-tracking. Structured SSMs (S4, S5) and selective SSMs [5] achieve strong long-range modeling with linear-time recurrence, but their diagonal, input-controlled linear state places them in TC^0 : Merrill et al. [4] show this excludes non-solvable group word problems such as A5 at growing length. Grazi et al. [6] extend the reachable class of linear RNNs by allowing negative eigenvalues, unlocking some state-tracking (e.g., parity), but non-solvable word problems remain outside fixed-depth diagonal linear designs. Our A5 result is the complementary positive: a *nonlinear* recurrence that tracks the standard A5/60 protocol at $64\times$ training length.

Transformers and copy shortcuts. Jelassi et al. [1] isolate copy as the regime where attention’s lossless re-reading beats recurrence. Liu et al. [3] show attention learns automaton shortcuts that fail beyond the training length — precisely the cliff our RoPE baseline exhibits at $2\times$ on copy. Our copy result is a recurrence that does *not* cliff, via an explicit induction read.

Attention–recurrence hybrids. Griffin/Hawk interleave gated linear recurrences with local attention [7]; Jamba interleaves Mamba and attention layers [8]. Such hybrids can cover both corners *by composition*: two distinct modules, each paying its own cost, with the division of labor fixed by the architecture. Extended recurrences — xLSTM’s exponential gating and matrix memory [9], delta-rule fast-weight states [10, 11], RWKV [12] — enrich the recurrent state itself but do not expose an induction-style successor read. DABSN is neither: a single block, containing no attention layer, whose retrieval primitives read the recurrence’s *own committed writes* over position-free content keys, with every pathway gated to a no-op at initialization and adopted only if it reduces task loss (§3.5).

Associative recall and induction. Multi-query associative recall [13] is the standard synthetic for in-context retrieval; induction heads [14] are the attention circuit for “find the previous occurrence and output what followed.” DABSN bakes an induction primitive directly into a recurrent block — the same match-then-successor computation, but over position-free content keys carried in recurrent state, which is why it is length-flat.

Fast-weight and modulation memories. The persistent bias state is related to fast-weight programmers and gain-modulation memories [10, 15, 16], but differs in treating memory as a retention/write/plasticity/budget design space over a baseline-modulating state, and in exposing multiple read primitives selectable by the optimizer.

3 Method

3.1 The DABSN state equation

DABSN separates the *expressed activation* y_t from a *persistent bias state* b_t . The activation expresses the current input under the present operating conditions; the bias state persists and changes the conditions under which future activations form. The family is one template:

$$\begin{aligned} y_t &= \tanh(Wx_t + b_t) \\ b_{t+1} &= R_t \odot b_t + \beta + M_t \odot (Ay_t), \quad b_0 = 0, y_0 = 0 \end{aligned} \tag{1}$$

with a *retention* operator R_t , a *write/plasticity* operator M_t , a learned baseline β , and activity-to-bias feedback Ay_t . The separation is not notational: in the family’s lineage, decoding the output from b_T alone outperforms decoding from y_T alone on memory tasks (with the concatenation $[y_T, b_T]$ best when stored memory must be interpreted through current context) — direct evidence that the persistent state, not the expressed activation, is what carries the memory. Variants of the family differ in R_t and M_t ; the converged design used throughout this paper is the budgeted, admission-gated member described next.

3.2 The governor core: budgeted, gated plasticity

At each step, an admission gate s_t , a novelty trace n_t , a slow saturation trace c_t , and an energy budget $e_t \in [0, 1]$ co-regulate plasticity:

$$\begin{aligned}
s_t &= \text{hardsigmoid}(W_g x_t + U_g y_t) && \text{write admission} \\
n_t &= \tanh(|A y_t - b_t|) && \text{novelty} \\
c_t &= \rho_c c_{t-1} + (1 - \rho_c) [n_t(1 - e_t)] && \text{saturation (slow)} \\
n_t^{\text{eff}} &= n_t (1 - \sigma(\theta_s) c_t) && \text{saturation downregulates novelty} \\
p_t &= s_t e_t && \text{plasticity = admission} \times \text{budget} \\
b_{t+1} &= (1 - \alpha) b_t + \beta + \lambda (p_t \cdot A y_t) && \text{write into persistent memory} \\
e_{t+1} &= \text{clamp}(e_t + r_t(1 - e_t) - k_t p_t, 0, 1) && \text{spend-and-recover budget} \\
m_t &= p_t \cdot (A y_t) && \text{the committed write}
\end{aligned} \tag{2}$$

m_t is the object every downstream memory reads. The write cost k_t and recovery r_t are exp-tanh functions of $(s_t, y_t, b_t, n_t^{\text{eff}}, c_t)$.

3.3 The unified read: one score, three primitives

Over an admitted set of writes (admission below), a single content score is computed once and shared by three reads that differ only by an added term and by which value bank they gather (all causal-masked, softmax over admitted j):

$$\begin{aligned}
q_i &= \text{normalize}(A y_i), \quad k_j = \text{normalize}(m_j), \quad \text{scale} = (\text{softplus}(\log\text{-temp}) + 1)\sqrt{H} \\
\text{content}[i, j] &= \text{scale} \cdot (q_i \cdot k_j) + \text{keybias}_j \\
\text{sig}[i, j] &= g_{\text{sig}}(\xi_i \cdot \xi_j), \quad \xi_t = \text{normalize}([\bar{e}_t, \bar{c}_t, \bar{n}_t, \bar{p}_t]) \in \mathbb{R}^4 \\
\text{SHORT}_i &: \text{softmax}_j(\text{content} + \text{adm}_j), \text{ values } m_j, \text{ mask } j \leq i \\
\text{PERM}_i &: \text{softmax}_j(\text{content} + \text{sig} + \text{adm}_j), \text{ values } m_j, \text{ mask } j \leq i \\
\text{INDUCT}_i &: \text{softmax}_j(\text{content} + \text{adm}_j), \text{ values } m_{j+1}, \text{ mask } j < i \\
\text{retrieved}_i &= g_{\text{short}} \text{SHORT}_i + g_{\text{perm}} \text{PERM}_i + g_{\text{ind}} \text{INDUCT}_i
\end{aligned} \tag{3}$$

- **Content/PERM** read the matched write (the token itself) using a position-free content key plus a *position-from-association* key ξ — the *trace signature*, read off the recurrence’s own internal traces. The state is the position; there is no positional encoding.
- **Induction** keeps the same content match but gathers the *successor* m_{j+1} under a strict-causal mask ($j < i$, so $j+1 \leq i$ is available): “find where this token occurred before, output what followed it.” This is the copy primitive; the plain content read returns the matched token itself, which is useless when the task is to emit the *next* token.

Admission. Each write carries a score adm_j computed from the full set of internal traces (plasticity, novelty, saturation, budget); only writes within a *learned relative window* of the per-sequence best are readable: $\text{member}_j = [\text{adm}_j \geq \max_k \text{adm}_k - \text{softplus}(w)]$, with w initialized wide. The window is relative, so the admitted set never collapses to one write and never starves the read; the admitted size n_{max} emerges per task ($\approx T$ on incompressible copy, small on selective recall), making the read $O(T \cdot n_{\text{max}})$ rather than $O(T^2)$.

Memory scaling. The admitted write set is the block’s readable memory, and its size is task-adaptive by design. On incompressible copy this is information-theoretically forced: emitting T tokens losslessly requires $T \log_2 |V|$ recoverable bits, which no fixed-width state can hold, so the admitted set grows to $\approx T$ and the fixed-width bandwidth bound of Jelassi et al. [1] is sidestepped rather than contradicted — that bound applies to recurrences that compress history into H numbers, and on copy DABSN does not. On selective-recall tasks the admitted set stays small and the block operates as a compact recurrence. What is constant across both regimes — and what we argue produces the length-flatness — is the *addressing*: retrieval into the admitted set is purely content- and trace-based, with no positional table of any kind, so nothing in the read mechanism is calibrated to the training length. The copy claim of this paper should be read accordingly: not that a fixed-width state beats the bandwidth bound, but that position-free addressing over state-carried writes removes the length cliff that positional addressing exhibits.

3.4 The recurrence: an explicit nonlinear scan

The unified read is purely associative and has no sequential nonlinear state, so it cannot track non-solvable group composition at length. A gated consolidation recurrence is therefore attached as an explicit diagonal scan in hidden space:

$$\begin{aligned}
\text{expected}_t &= \rho_e \text{expected}_{t-1} + (1 - \rho_e) m_t && \\
\text{pe}_t &= \tanh(|m_t - \text{expected}_{t-1}|) && \text{surprise} \\
a_t &= \rho_a a_{t-1} + (1 - \rho_a)(1 - \text{pe}_t), \quad a_0 = 1 && \text{confidence trace} \\
\text{eff}_t &= \text{retain} + (1 - \text{retain}) \cdot g_a \cdot a_t && \text{retention gate} \\
\text{long}_t &= \text{eff}_t \text{long}_{t-1} + (1 - \text{eff}_t) (p_t \cdot \text{pe}_t \cdot m_t) && \\
\text{read}_t &= \text{out}_t + g_{\text{rec}} \text{long}_t &&
\end{aligned} \tag{4}$$

Because this is a diagonal linear-state primitive, a fused GPU scan kernel reproduces it exactly (verified against a CPU gradient check).

3.5 The gated residual and superset-at-init

All retrieval is folded into the core through one gated residual $z_t = y_t + \gamma \text{read}_t$, with γ initialized near zero. At initialization the entire read/recurrence apparatus is a no-op ($z_t \approx y_t$); training dials it in per task. Every added pathway (induction, recurrence, trace signature) likewise begins inert via its gain, so a pathway is adopted only if it reduces loss. This makes the design lineage itself an ablation grid: each component was added inert against the previous family member and benchmarked across the task matrix, so a stage that did not help would simply have trained back to its inert initialization — that each was instead adopted, and improved on its predecessor, is per-component evidence of contribution. **No mechanism routes the reads by task** — the gains are ordinary learned parameters.

4 Experimental Setup

Model. DABSN in its *sequence* (seq) read geometry — the geometry this paper describes. Compact copy/recall stress configs use approximately 17–31k parameters depending on hidden width ($h48$ – $h96$) and task head; the standard A5/60 headline uses the wider $h256$, **seq:256:256** config with 378,198 parameters. The *field* and *hybrid* variants appearing in the three-seed stress pass (§5.4)

Table 1: Copy accuracy vs. evaluation length (vocab 64, train length 64, $h48$, 1000 steps; mean $\pm$ std over 3 seeds). Chance = $1/64 = 0.0156$. GRU and LSTM sit at chance at every length above the training length (Fig. 1). Machine-readable result tables are included as ancillary CSV files with this release.

eval length	64	128 (2 $\times$)	256	512 (8 $\times$)	1024	2048	3200 (50 $\times$)
DABSN (seq)	.985	.992	.996	.997	.995	.984	.961 $\pm$.035
Transformer (RoPE)	1.000	.049 $\pm$.014	.063	.046	.033	.025	.022 $\pm$.001

change only the read geometry over the same block. The verified-CSV model names `dabsn_seq`, `dabsn_field`, and `dabsn_hybrid` denote this same architecture; internal implementation names in archival logs are mapped to this public model name.

Protocol. Each model is run at its own best configuration — the fair protocol for a family whose empirical sweet spot is low hidden width with a higher learning rate; gated-recurrent and attention baselines prefer lower learning rates. Train at a short length, evaluate at multiples of it (train-short/evaluate-long). Chance is reported per task, and seed counts are stated per result.

Baselines. GRU, LSTM, and transformers with absolute and RoPE (relative) positional encodings. We treat RoPE as the primary attention opponent deliberately: the absolute-PE transformer falls to chance the instant evaluation exceeds training length, whereas RoPE extrapolates far better and is the honest opponent.

5 Results

5.1 Copy — the memory-bandwidth gap (induction read)

Copy is fed x_k and must emit x_{k+1} . The plain content read literally cannot answer it (it returns the matched token); with the induction read, copy is length-flat. Table 1 reports the three-seed verified pass at vocabulary 64, $h48$, train length 64, evaluated to 50 $\times$; Figure 1 (left) plots the full curves.

The RoPE transformer solves the training length perfectly and collapses at 2 $\times$; DABSN holds 0.96–1.00 out to 50 $\times$. A converged single-seed run at $h96$ (4000 steps) reaches accuracy 0.9999, 0.9999, 0.9994, and 0.9990 at lengths 64, 128, 256, and 512. The field-geometry variant of the same block reaches 0.992 ± 0.004 at 50 $\times$, so the behavior is a property of the block, not of one read geometry.

5.2 A5 — the standard state-tracking gap (recurrence)

A5 supervises the running group product densely (60-way classification at every position, no final-state shortcut). The A5 run uses the same block design, trained separately with train length 256, cosine-decayed learning rate 0.003, 60,000 steps, $h256$ and DABSN layer spec `seq:256:256`; chance = $1/60 = 0.0167$:

DABSN is length-flat through length 16,384, while the LSTM positive control learns the training distribution perfectly but decays after the training horizon. RoPE attention is already near chance by length 2048 and longer attention evals are skipped by the configured T4 quadratic-attention budget.

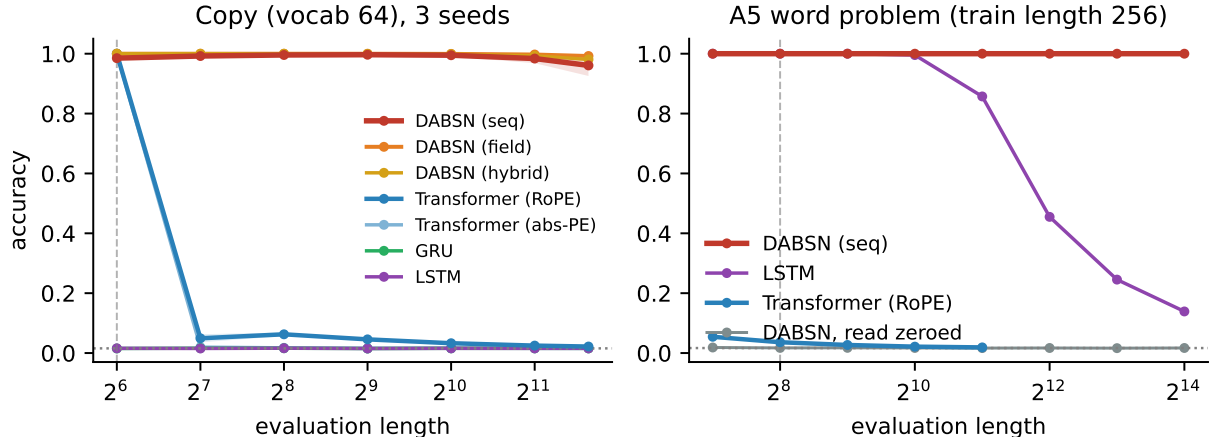


Figure 1: **One block design, both gaps.** Left: copy accuracy vs. evaluation length (train length 64, vocab 64; mean $\pm$ std, 3 seeds). Attention (RoPE) collapses at $2\times$; gated recurrences (GRU/LSTM) sit at chance; DABSN is length-flat to $50\times$. Right: standard A5/60 word-problem accuracy (dense 60-way supervision; train length 256, single seed). DABSN remains perfect to length 16,384 ($64\times$), while the LSTM positive control decays at long length and RoPE attention is at chance. “DABSN, read zeroed” is an eval-time causal control: the same trained checkpoint re-evaluated with the gated-residual read gain γ (§3.5) zeroed, removing the entire retrieval output at once (all primitives and the consolidation channel; distinct from the per-pathway gain knockouts of Table 3). Accuracy falls to chance at every length: the tracked state is expressed through the read pathway, and the intact curve is not an artifact of the task head.

Second seed. A second training run at seed 1, identical `seq:256:256` config, reproduces the headline exactly: 1.000 at every length from 128 to 16,384 ($64\times$), matching seed 0 (Fig. 2, left). The A5/60 result is thus two-seed, not single-seed.

The nonlinear state is necessary: a linear-state ablation. The theory attributes A5 to the block’s *nonlinear* sequential state (§3.4); we test this causally by removing that nonlinearity and nothing else. In the linear-state variant every transition coefficient of the core scan is a function of the input alone — the persistent state b_t never feeds back into the gates, novelty, energy law, or write — so the state evolves as a diagonal linear system with input-controlled coefficients, the regime Merrill et al. [4] place in TC^0 . The parameter count, learning rate, step budget, and length curriculum are unchanged (`seq:256:256`, 378,198 params, seed 1). The ablated block does not merely lose length generalization — it never learns A5 *at all*, sitting at chance at the training length (0.026 train accuracy) and at every evaluation length (Fig. 2, right; Table 2). With every other factor held fixed, deleting the nonlinear state feedback destroys the capability outright, demonstrating — rather than inferring by elimination (§5.3) — that the nonlinear recurrence, not the reads or the parameter budget, is what tracks A5.

5.3 Mechanism: the optimizer picks the primitive

Nothing routes the reads — the gains are ordinary learned parameters — yet each separately trained model recruits only the pathway its task needs. We show this two ways: causally on the A5/60 headline itself, and as a per-pathway double dissociation on copy.

Table 2: A5/60 accuracy vs. evaluation length, from the machine-readable ancillary result tables. DABSN (seq) is two-seed (both seeds identical to the reported precision); the linear-state ablation is the seed-1 input-controlled (TC^0) variant of the same block. LSTM/RoPE are the single-seed headline baselines.

eval length	128	256	512	1024	2048	4096	8192	16384 (64 $\times$)
DABSN (seq)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
DABSN, linear-state	0.036	0.026	0.022	0.019	0.018	0.017	0.017	0.017
LSTM	1.000	1.000	1.000	0.995	0.857	0.455	0.245	0.139
Transformer (RoPE)	0.054	0.036	0.027	0.021	0.019	skipped	skipped	skipped

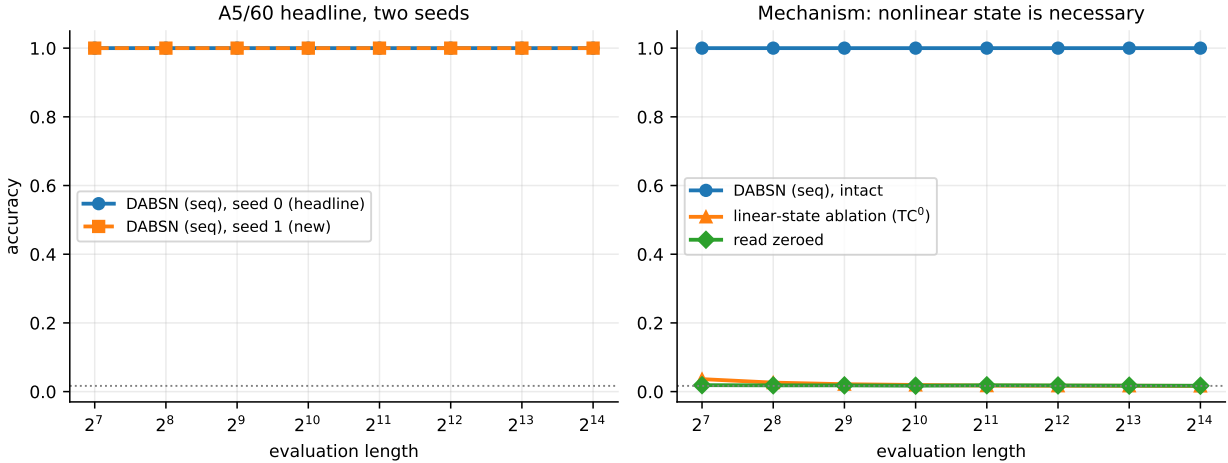


Figure 2: **A5/60: two-seed reproduction and the linear-state ablation.** Left: the seq:256:256 block is perfect and length-flat to 64 $\times$ at both seeds (seed 0 headline, seed 1 new run). Right: two causal controls on the seed-1 checkpoint. The input-controlled *linear-state* ablation of the same block (the TC^0 regime of Merrill et al. [4]) fails to learn A5 even at the training length, and the “read zeroed” control (§3.5) is at chance; the intact block is at 1.000 throughout. Chance = $1/60 = 0.0167$.

A5/60: the nonlinear state carries it. The two ablations of §5.2 establish the state-tracking mechanism directly on the A5/60 checkpoint. Removing the nonlinear state feedback (the linear-state variant) collapses A5 to chance even at the training length, and zeroing the whole read pathway (read zeroed) is likewise at chance. The capability needs both the block’s nonlinear recurrent state and the read that expresses it; no associative read alone suffices, exactly as the theory requires (§3.4), because the unified read has no sequential nonlinear state and cannot carry non-solvable composition at growing length.

Copy: a per-pathway double dissociation. On copy the assignment is visible in the converged gains and confirmed causally by knockouts. The induction gain trains up ($0.1 \rightarrow 0.79$) while the recurrent consolidation channel stays idle ($g_{\text{rec}} = 0.017$) — expected, since the content read returns the matched token rather than its successor, so only the induction read can emit the *next* token (§3.3). Eval-time knockouts zero one pathway gain on the trained checkpoint, all other weights untouched (Table 3).

Zeroing the induction gain sends copy to exact chance at every length, while zeroing the consol-

Table 3: Eval-time knockout of one pathway gain on the same trained copy checkpoint (verified-stress config: $h48$, train length 64, 1000 steps, single seed, evaluated to length 512; chance = 0.0156). Zeroing the induction gain sends copy to chance; zeroing the consolidation or signature gain changes nothing. Machine-readable result tables are included as ancillary CSV files with this release.

task	condition	64	128	256	512
copy	intact	0.985	0.992	0.996	0.997
	$g_{\text{ind}} = 0$	0.017	0.016	0.015	0.016
	$g_{\text{rec}} = 0$	0.985	0.992	0.996	0.997
	$g_{\text{sig}} = 0$	0.985	0.992	0.996	0.997

idation channel or the trace signature changes nothing — a full double dissociation: the induction read is necessary, and the recurrent consolidation channel is causally inert on copy, exactly as the gain values indicated. The two results together are the mechanism claim in its sharpest form: copy is solved by a single discrete, removable primitive — the induction read — while state tracking is solved by the nonlinear state itself, a substrate no read gain can remove because it is not a pathway. Deleting the state’s nonlinear feedback destroys A5 outright (§5.2); deleting the induction read destroys copy; and in each model the off-task pathway is idle. One block; the optimizer recruits the right computation per task, with no mechanism routing the reads.

5.4 Length generalization is an emergent property

The same train-short/evaluate-long flatness appears across unrelated task families (each model at its own config; chance noted):

Table 4: Emergent length-flatness across task families (single-seed converged runs; see Table 5 for the 3-seed pass). On character LM, DABSN trails RoPE *in-distribution* at context 128 by 0.057 bpc — the win is out-of-distribution length robustness, not in-distribution quality.

Task	train→eval	factor	DABSN	best baseline
MQAR (recall)	64→2000	31×	1.000	RoPE 0.178; abs-PE chance
Key-value ($h16$)	64→512	8×	1.000	GRU 0.081; transf. 0.055
Burst-anomaly ($h16$)	64→1024	16×	1.000	transf./LSTM/coRNN $\approx$ 0.49
Char-LM (Shakespeare)	128→2048	16×	bpc $\approx$ 2.37 ($\approx$ flat)	RoPE 5.86 (near-random)

Verified three-seed stress pass. After the initial draft, we ran a three-seed length stress pass on the DABSN read geometries at $h48$, train length 64 (Table 5, Fig. 3). These are not the same checkpoints as the converged headline runs above; they verify that the length-flat behavior of the block is not a single lucky seed and is shared across its read geometries.

Each benchmark trains each model from scratch on the short horizon, then evaluates that same instance zero-shot at longer horizons on the same task generator and scoring rule: copy supervises only the output-copy positions, MQAR scores only query positions, and key-value scores the final queried value. The DABSN rows differ only in read geometry; baselines use the same input stream and task head. The interpretation is mechanistic: attention re-derives bindings from positions it was never trained on and decays; DABSN carries the binding in recurrent state, so length is irrelevant.

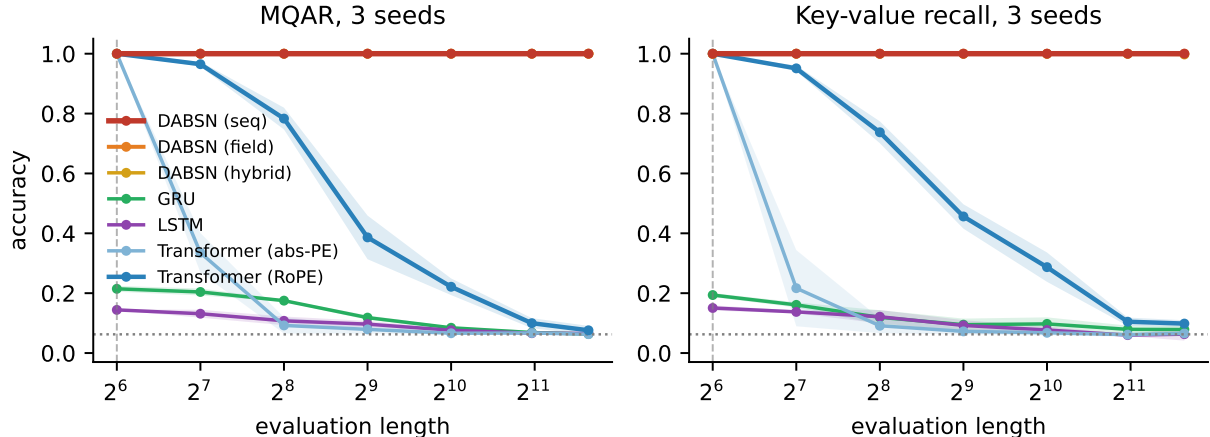


Figure 3: **Verified 3-seed stress pass to $50\times$.** MQAR (left) and key-value recall (right), train length 64, $h48$, evaluated to length 3200. All three DABS read geometries hold 1.000 ± 0.000 (one hybrid key-value seed dips to 0.992); RoPE decays toward chance; GRU/LSTM/abs-PE sit at chance.

Table 5: Verified 3-seed length stress at $50\times$ ($h48$, train length 64; mean $\pm$ std, $n=3$). Copy/MQAR use 1000 training steps; key-value uses 2000. The machine-readable result tables record exact steps, seeds, learning rates, and run identifiers. Chance: 0.0156 copy, 0.0625 MQAR/KV.

Task	geometry	DABS at $50\times$	best baseline at $50\times$
Copy (vocab 64)	seq	0.961 ± 0.035	RoPE 0.022 ± 0.001
	field	0.992 ± 0.004	
MQAR	seq/field/hybrid	1.000 ± 0.000	RoPE 0.076 ± 0.015
Key-value	seq/field	1.000 ± 0.000	RoPE 0.098 ± 0.011

6 Discussion

The standard framing treats the copy and state-tracking gaps as evidence for a trade-off: a layer optimized for lossless re-reading is structurally poor at sequential nonlinear state, and vice versa. Our result is that a single persistent-modulation recurrence holds both, because it carries two distinct computational primitives in one block — an associative induction read and a nonlinear diagonal recurrence — and lets the optimizer weight them per task. The induction read supplies the lossless “re-read the successor” operation copy needs without a positional table; the nonlinear recurrence supplies the genuine sequential state A5 needs without attention’s length-bounded shortcut. Because the off-task pathway goes idle — and knockout shows it is causally inert, not merely down-weighted (§5.3) — this is not an ensemble paying for both at once; it is one substrate expressing the right computation.

A natural objection is that any attention-recurrence hybrid [7, 8] also covers both corners. The distinction is structural. A hybrid composes two modules — a softmax attention layer over token embeddings and a separate recurrent layer — with the division of labor fixed by the architecture, and it inherits attention’s positional length-cliff on copy-style re-reading (Table 1). DABS contains no attention layer: its reads gather the recurrence’s own committed writes m_j under one shared content score; the keys are position-free functions of recurrent state, so nothing in the read references

absolute or relative position; the read is admission-bounded rather than dense; and the entire apparatus is gated to a no-op at initialization, so each primitive is *recruited by the optimizer*, not allocated by the designer. The observable consequence is exactly the experiment: the module hybrids inherit for re-reading cliffs at $2\times$, while the induction read is flat at $50\times$.

We therefore read the evidence as: **persistent-modulation memory is a third substrate**, distinct from “attention that trades away state” and “recurrence that trades away recall,” rather than a compromise between them. The emergent length-flatness across four further task families (§5.4) is consistent with this: length-robustness follows from carrying bindings in state rather than re-deriving them from position.

7 Limitations

1. **Seeds.** A5/60 state tracking is now two-seed (both seeds perfect and length-flat to $64\times$; §5.2, Fig. 2). Burst-anomaly and character LM remain single-seed. The *h96* copy headline is single-seed, but the verified *h48* copy, MQAR, and key-value stress passes are 3-seed. Single-seed rows are *demonstrated, not proven*; wider seed counts on the remaining single-seed rows are the primary hardening step.
2. **Protocol.** Comparisons are each-model-at-its-own-best-config. This is the fair protocol for a low-hidden/high-LR family, but rankings are configuration-dependent; a single shared grid would not reproduce them. A matched-config baseline sweep is an immediate next step.
3. **The Mamba comparison is theorem-backed** [4], not a measured head-to-head here; we cite the impossibility result rather than running Mamba on A5.
4. **Mechanism evidence.** On the A5/60 headline the state-tracking mechanism is shown directly and causally: the linear-state ablation and the read-zeroed control each collapse it to chance (§5.2, Fig. 2). The copy-side per-pathway double dissociation (Table 3) is single-seed and evaluated to length 512; extending it to the $16\times$ horizon, and logging the per-pathway gains on the A5/60 run itself, are straightforward future hardening.
5. **Scope.** Copy is shown at vocabulary 64 only. MQAR has a sharp solve/no-solve basin and CUDA is not fully deterministic under seeding; the verified pass solves all three seeds to $50\times$, but larger seed counts remain future hardening.
6. **Speed.** The implementation includes fused scan/read kernels, but this paper does not report a controlled throughput or latency study. No speed comparison with vendor GRU/LSTM is claimed; kernel performance remains implementation- and hardware-dependent.

8 Availability

Machine-readable result tables for this paper are included as ancillary CSV files. The public code repository is <https://github.com/BleedingXiko/dabsn>. It includes the framework source, native C++/OpenMP and Triton/CUDA kernels, package tests, runnable paper reproductions, machine-readable results, and paper build source. Checkpoints and full training artifacts are retained by the author and are available on reasonable request.

References

- [1] Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In *International Conference on Machine Learning (ICML)*, 2024.
- [2] David A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1 . *Journal of Computer and System Sciences*, 38(1):150–164, 1989.
- [3] Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *International Conference on Learning Representations (ICLR)*, 2023.
- [4] William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In *International Conference on Machine Learning (ICML)*, 2024.
- [5] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- [6] Riccardo Grazi, Julien Siems, Jörg K. H. Franke, Arber Zela, Frank Hutter, and Massimiliano Pontil. Unlocking state-tracking in linear RNNs through negative eigenvalues. In *International Conference on Learning Representations (ICLR)*, 2025.
- [7] Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024.
- [8] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.
- [9] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended long short-term memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [10] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning (ICML)*, 2021.
- [11] Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations (ICLR)*, 2025.
- [12] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, et al. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.
- [13] Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. *arXiv preprint arXiv:2312.04927*, 2023.
- [14] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, et al. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

- [15] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.
- [16] Jimmy Ba, Geoffrey Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.