The Geometry of Fast and Slow Weights In A Shared Address Space

Anonymous authors
Paper under double-blind review

Abstract

Fast weights have proven ubiquitous in memory systems of current state-space models (SSMs), but only recently have we seen advances in their theoretical understanding and in how they may relate to canonical forms of recurrent neural networks. Here, we provide a theoretical bridge between a classical connectionist view of recurrent weights as fixed connectivity, and one of fast weights, closer to modern ML literature. We use a structural connectivity approach to derive a low-rank common projection space for both slow and fast weights, relating each to mean and variance modes. In doing so, we provide a geometric and dynamical interpretation of fast weights from the perspective of a common presynaptic or address space, which reflects hard-wired pathways of inputs, and distinct subspaces for postsynaptic or content spaces. We find that delta-rule corrections, operating within the slow weight's subspace, reduce interference compared to a vanilla fast-weight update. This provides a framework for analyzing a division of labor between in-context or test-time learning, stored in fast weights, and more long-term functional attributes. We use this framework to design an SSM with competitive results in within-task learning and analyze the contributions of fast and slow modes.

1 Introduction

Modern SSMs (Dao & Gu, 2024), linear transformers (Schlag et al., 2021), and fast weight programmers (Ba et al., 2016), which do not use an explicit attention system, have rekindled an interest in adaptive RNNs with differentiable plasticity (Miconi et al., 2018; Katharopoulos et al., 2020). Early ML research focused on prescribing a role to fast weights in complementary learning systems (CLS) (Hinton & Plaut, 1987; Schmidhuber, 1992), however, recent memory-systems have increasingly relied on differential timescales Irie et al. (2025); Behrouz et al. (2025).

Fast weight models, where recurrent connectivity of the latent space changes dynamically, are diverse in form but generally present unitary rank additions per sequence step (Ba et al., 2016; Irie & Gershman, 2025). As for slow weights, we consider the case of fully static weights of classical RNN systems. Previous approaches on fast and slow weights considered both in terms of presenting cascading mechanisms (Benna & Fusi, 2016; Kaplanis et al., 2018), exercising online control (Bicknell & Latham, 2025), or serving as error correction (Yang et al., 2024b; Asabuki & Clopath, 2025).

Recent theoretical work has unveiled the dynamical changes permitted by these RNNs compared to vanilla versions or gated mechanisms (Clark & Abbott, 2024; Gaudio et al., 2026; Wakhloo et al., 2025). Concretely, the energy function of self-reinforcing Hebbian patterns produces 'trampoline' dynamics, where new well basins in the energy function become available and patterns can be stored without catastrophic forgetting by strengthening the short-term plasticity or fast-weight mechanism (Gaudio et al., 2026). Here, instead, we build on seminal work on structured connectivity in RNNs (Mastrogiuseppe & Ostojic, 2018; Schuessler et al., 2024) to treat fast weights as arising from an intermediate, variable coupling between hidden units. We thus cast the fast-slow weight duality as a distinction between a mean-centered, stationary mode (slow weights) and a variance-based, transient mode (fast weights). This yields a common address, or presynaptic, subspace alongside dual-timescale content, or postsynaptic, subspaces. This framing lets us relate slow and fast components in a kernel-agnostic manner, and through post-hoc analysis we can inspect how each component

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

absorbs a distinct facet of learning. In particular, we propose that these complementary pathways share a common address space from which both can be related.

To instantiate this address–content decomposition, we impose rank-structured fast and slow weights. Work on low-rank transformers (Chen et al., 2021; Tay et al., 2023) and SSMs (Gu et al., 2021) has proven effective in producing useful inductive biases while reducing computational and memory costs. Another line of research across SSMs and linear attention increasingly treats sequence memory as a geometry problem rather than a purely algebraic one, recasting attention as state estimation on structured latent spaces (Dowling et al., 2026) and extending expressivity through explicit geometric constraints (Su et al., 2021; Loshchilov et al., 2024).

Furthermore, our model presents a framework for studying the weight transport problem, since it distinguishes the contributions of fast and slow weights. Thus, it permits studying the timescales among parameters and designing update rules (Schmidhuber, 1992; Sun et al., 2025; Behrouz et al., 2025; von Oswald et al., 2026).

Our key contributions in this article are:

1. **A geometric theory of fast and slow weights.** We derive a rank-constrained decomposition of recurrent fast weights from time-varying modulation of a fixed connectivity structure. Fast and slow weights inherit a common presynaptic (*address*) space while differing in their postsynaptic (*content*) subspaces, providing a geometric interpretation of their interaction.
2. **Fast-slow geometry controls associative interference.** Using associative-recall probes, we find that the geometry of fast writes relative to the slow content subspace determines retention and interference. Delta-rule updates exploit the slow-weight prediction while selectively replacing fast associations along reused addresses, substantially reducing interference relative to additive fast-weight writes.
3. **FastSlow: a shared-address recurrent SSM.** Motivated by this theory, we introduce FastSlow, a recurrent sequence model in which low-rank slow and fast operators share an address space while retaining distinct content states. The resulting architecture is chunkwise parallelizable and is competitive with published baselines on the MAD sequence-modeling benchmark.
4. **A division of labor between fast and slow memory.** Through post-training interventions and geometric diagnostics on associative tasks and a preliminary small language model corpus, we find complementary roles for the two pathways: slow weights capture persistent structure and provide a predictive backbone for fast updates, whereas fast weights support rapidly changing, context-dependent associations required for in-context recall.

2 Structured fast weights

We model fast weights by considering a time-dependent synaptic weight $\nu_{ij}(t)$, which describes the effective synaptic conductance from a presynaptic neuron j to a postsynaptic neuron i due to a coupling factor c_{ij}, such that $\nu_{ij}(t) = \sum_r c_{ij}^{(r)}(t)\, w_{ij}^{(r)}$.

This coupling factor is a common process on the postsynaptic side, which consists of a temporal modulation of synaptic efficacy exercised by fluctuating contacts. These could be interpreted as synaptic contact points between the same pre- and postsynaptic neuron, of which a given pair can have up to R, each gated independently. Thus, each postsynaptic neuron observes dynamic effective presynaptic loading $\nu_{ij}(t) = \sum_r c_{ij}^{(r)}(t)\, w_{ij}^{(r)}$. The index r labels a gating channel rather than an individual synapse, in other words, a pair (i, j) may route through up to R channels, and all inputs routed through channel r on neuron i share a single modulatory drive or common mode $\eta_i^{(r)}(t)$, which carries no presynaptic dependence. Channels are modulated independently of one another. As a simplification, we summarize all the previously described computations into a single temporal process underlying the per-contact gate $c_{ij}^{(r)}$. The residual by which an individual contact departs from its shared drive $\zeta_{ij}^{(r)}$ is independent across contacts and averages away in the effective field at large N (Appendix A.1), so in the main text we write the gate per neuron and mode, $c_{ij}^{(r)} \to c_i^{(r)}$, obeying

$$p\, \dot{c}_i^{(r)}(t) = -c_i^{(r)}(t) + \eta_i^{(r)}(t) \tag{1}$$

Thus, inputs are modulated by a vector of gates $(c_i^{(1)}, \ldots, c_i^{(R)})$, rather than a single scalar. The set of synapses that are effectively active reconfigures combinatorially. For instance, a synapse onto the mode-1 branch of neuron i is gated by $\eta_i^{(1)}(t)$ regardless of mode 2, and as the R common drives fluctuate independently, different subsets of the inputs to i switch on and off together. This modulation is postsynaptic and applies before the rate nonlinearity, so it shapes what the neuron integrates rather than what it outputs. Several channels can be combined but the theory below is presented at rank one.

Following low-rank RNN literature (Mastrogiuseppe & Ostojic, 2018), we consider a network of N neurons. The structural weight of the r-th synaptic contact from neuron j to neuron i is given by

$$w_{ij}^{(r)} = \frac{1}{N} u_i^{(r)} v_j^{(r)} + \frac{\sigma_w}{\sqrt{N}} \chi_{ij}^{(r)}, \tag{2}$$

where $u_i^{(r)}$ and $v_j^{(r)}$ are the postsynaptic (left) and presynaptic (right) loadings of the r-th rank-1 mode, and $\chi_{ij}^{(r)} \sim \mathcal{N}(0,1)$ represents independent structural heterogeneity. Throughout, $\mathbf{u}$ denotes the postsynaptic/left-hand mode – the *content* side – and $\mathbf{v}$ the presynaptic/right-hand mode – the *address* side. We write U and V for the matrices stacking $\{\mathbf{u}^{(r)}\}$ and $\{\mathbf{v}^{(r)}\}$ columnwise (see Appendix A.1).

We conceive the common mode coupling as a postsynaptic mechanism model as independent Ornstein-Uhlenbeck process (Appendix A.1). We argue this is similar to dendritic computation where synaptic input currents present postsynaptic non-linear integrations before reaching the soma. We decompose the common mode drive into

$$\eta_i^{(r)}(t) = \bar{\eta}^{(r)} + \tilde{\eta}_i^{(r)}(t), \tag{3}$$

where $\bar{\eta}^{(r)}$ is a mean-centered or constant component and $\tilde{\eta}_i^{(r)}(t)$ captures temporal fluctuations, also a OU process (Appendix A.1, Eq. (A.8)). The constant contribution induces a static low-rank component in the connectivity, while the fluctuations give rise to a time-dependent fast-weight matrix (see Appendix A.1).

Specifically, the fast-weight matrix $A(t)$ is defined via its microscopic gating states (see Appendix A.1), where the per-mode contribution is the outer product $\mathbf{k}_A^{(r)}(t)\,(\mathbf{v}^{(r)})^\top$ of the fast left vector $\mathbf{k}_A^{(r)}(t) := \tilde{\boldsymbol{\eta}}^{(r)}(t) \circ \mathbf{u}^{(r)}$ ($\circ$ the Hadamard/elementwise product) with the presynaptic loading $\mathbf{v}^{(r)}$. Stacking the fast left vectors columnwise into $K(t) := [\mathbf{k}_A^{(1)}(t), \ldots, \mathbf{k}_A^{(R)}(t)]$ – also refered to as $K_A(t)$, or $K_{A,t}$ in some derivations below –, the resulting macroscopic dynamics are

$$p\,\dot{A} = -A + K(t)\,V^\top, \tag{4}$$

i.e. $A(t)$ is the filtered or leaky-integrated output of the rank-R drive stacked in $K(t)$ (Appendix A.1 gives the full $1/N$-normalized microscopic form). The fast left state $K(t)$ absorbs the postsynaptic mechanisms, both at the dendritic level and the somatic rate process, while the right/presynaptic factor V is inherited unchanged from the structural connectivity.

The mean-centred drive rescales the structural motif by a constant and, combined with the random background connectivity (Appendix A.1), defines the slow connectivity matrix

$$J = UDV^\top, \qquad D := \mathrm{diag}(\bar{\eta}^{(1)}, \ldots, \bar{\eta}^{(R)}), \tag{5}$$

where U and V stack the postsynaptic and presynaptic rank-R loadings $\{\mathbf{u}^{(r)}\}$, $\{\mathbf{v}^{(r)}\}$ columnwise. It follows naturally that the total recurrent weight matrix is $W_r = J + A(t)$.

Thus, we have shown that, in the large-N limit, contact-level fluctuations vanish from the structured low-rank field, leaving the shared-address decomposition $J = UDV^\top$ and $A(t) = K(t)V^\top$, as presented in Appendix A.3.

3 Geometrical and dynamical interpretations of slow-fast learning

Our framework captures the slow and fast weight components of individual neurons in terms of the total input received, as seen in (Fig. 2a). Each of the activations projects into a slow or structural

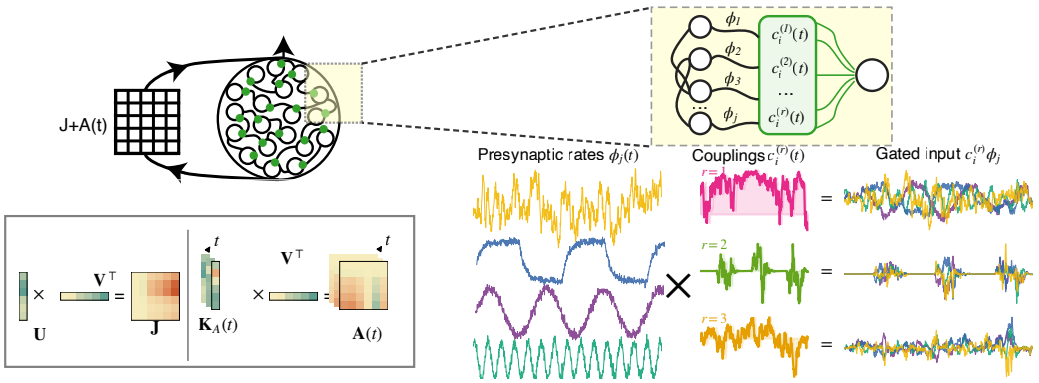


Figure 1: Postsynaptic coupling and mixture produces effective fast-slow weights in RNNs. **(Top left)** Schematic of a RNN presenting fast or time-dependent weights $A(t)$ and slow or fixed weight J. **(Top right)** Equivalent mixture model as observed by a postsynaptic neuron where coupling modes $c_i^{(r)}(t)$ are presented as intermediate components between the presynaptic activations ϕ_i and the postsynaptic activity. **(Bottom left)** Decomposition of the structural or non-random component of the connectivity $J+A$ shown as equivalent rank-1 modes and their outer-product weight matrices. K and A are update-driven traces, thus, time-varying. **(Bottom right)** Schematic examples of contact gates. Three input activations (left) are modulated by $R = 3$ different gating signals (middle) to yield a combinatorial code in the postsynaptic neuron (right).

component ($[J\boldsymbol{\phi}]_i$) and a fast, contact-gated component ($[A\boldsymbol{\phi}]_i$) whose magnitude, depends on the alignment of the fast left factor $\mathbf{k}_A$ with the postsynaptic direction $\mathbf{u}$.

Here, the slow component remains constant in contributing to a baseline of the neuron whereas the fast-weights are determined by the direction of the effective left vector $\mathbf{k}_A^{(r)}$. Comparing Eqs. (4) and (5), $A(t)$ and J share the presynaptic (right) factor $(\mathbf{v}^{(r)})^\top$ exactly, so the fast weights add no new presynaptic dimensions; the two differ only in their postsynaptic (left) factor, $\mathbf{k}_A^{(r)}(t)$ versus $\mathbf{u}^{(r)}$. The geometry is therefore controlled entirely by the alignment of $\mathbf{k}_A^{(r)}$ with $\mathbf{u}^{(r)}$ (Fig. 2b). If the common coupling drive is homogeneous across neurons, then $\mathbf{k}_A^{(r)} \propto \mathbf{u}^{(r)}$ – i.e., the fast weights merely rescale motifs in J, and produce no new latent directions. This trivially corresponds to the Hebbian (reinforcing or positive semi-definite) and anti-Hebbian (opposing or negative semi-definite) modulation, which has been studied as a self-consistent loop with neural activity in the literature – see Clark & Abbott (2024); Wakhloo et al. (2025) and Gaudio et al. (2026). On the other hand, a heterogeneous drive gives $\mathbf{k}_A^{(r)}$ components in $(\mathbf{u}^{(r)})^\perp$, expanding the postsynaptic subspace beyond the structural J. This escape is what makes the fast weights functionally distinct from being a gain modulator on the slow ones.

Next, we ask under which timescales the underlying noisy process of the contact coupling allows the emergence of fast weights. In particular, here we ask what is *fast* at the microscopic scale as determined by the intrinsic timescales of the fast weight process p and the drive timescale τ_η. In Fig. 2c we plot the stationary variance b_i, derived in Appendix A.1, for the case of a zero structural contact noise ($\sigma_\zeta = 0$). This stationary variance represents the content availability of fast weights in relation to the underlying OU process or, in the dimensionless case, p/τ_η. Our approach is within a dynamical mean-field theory (DMFT) approach, yet we find numerically that independently of this noise value the activity converges to $O(N^{-1/2})$, as seen in Fig. 2d,. Taken together, this implies that our noisy process approximation for coupling-based fast weight emergence relates both information content and noise available in a contingent variance mode to the overall macroscale process governing the fast weight RNN.

Moreover, we employed a linearized regime to show how fast weights produce changes in the eigenspectra of the RNN (Schuessler et al., 2020). Fig. 2e–f illustrates the associated spectral effect for a single timestep addition of the rank-1 fast weights in the total recurrent space, tracking the resulting

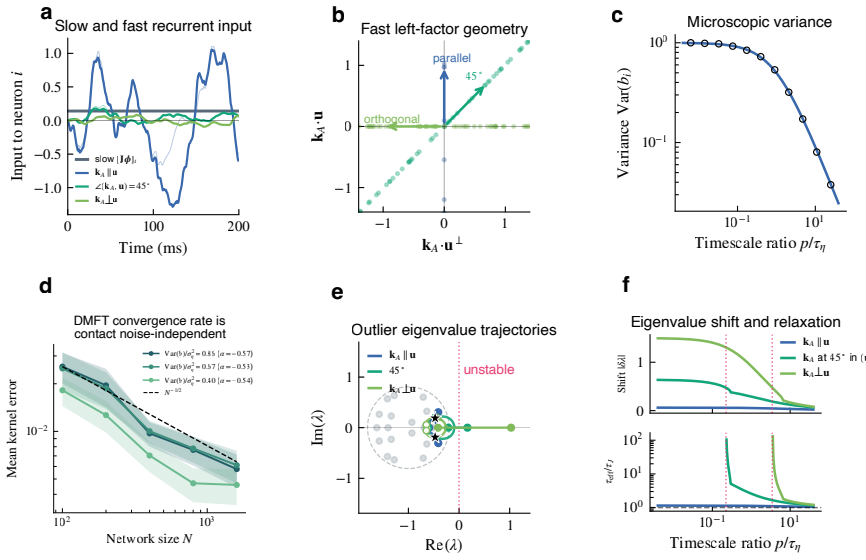


Figure 2: Intermediate couplings between presynaptic and postsynaptic terms produce subspace selectivity of fast weights. **(a)** Sample postsynaptic currents due to dendritic fast weights (colored) and the structural or slow component (black). The fast traces compare equal-norm left factors parallel to $\mathbf{u}$, at 45° toward $\mathbf{u}^\perp$, and parallel to $\mathbf{u}^\perp$ while retaining the same right factor $\mathbf{v}$. We plot a simulation example (translucent) and mean-field theory (solid). **(b)** Projection of the effective low-rank vector $\mathbf{k}_A(t) = \tilde{\boldsymbol{\eta}} \circ \mathbf{u}$ onto the postsynaptic axis ($\mathbf{u}$) and its orthogonal complement ($\mathbf{u}^\perp$). Each cloud corresponds to a modulation pattern $\tilde{\boldsymbol{\eta}}$. For uniform $\tilde{\boldsymbol{\eta}}$, $\mathbf{k}_A \parallel \mathbf{u}$ (slow mode: A rescales motifs already in J), whereas heterogeneous modulation tilts $\mathbf{k}_A$ away from $\mathbf{u}$, producing a fast-weight component $\mathbf{k}_A\mathbf{v}^\top$ whose postsynaptic factor leaves the rank-1 structure of J. The presynaptic (right) factor $\mathbf{v}$ is shared with J throughout and is not a free axis of this panel. **(c)** Stationary variance of the centered filtered OU state b_i versus p/τ_η, with $\sigma_\zeta = 0$; circles show exact stationary simulations. **(d)** Numerical check of the conditional DMFT closure for a heterogeneous gate ($\tilde{\eta}_i^{(r)}$ independent across i, Eq. (A.8)), across contact noise regimes. The convergence rate is exponent-universal and the prefactor depends on $\mathrm{Var}(b)/\sigma_\eta^2$ (see Appendix A.3). **(e)** Tracked linearized eigenvalue paths as the strength of the three $\mathbf{k}_A\mathbf{v}^\top$ choices in (b) increases; open and filled circles mark the initial and final values, the gray points show the background spectrum, and the red line is the stability boundary. **(f)** Structured-eigenvalue shift (top) and corresponding relative continuous-time relaxation time (bottom), tracked numerically for the same random-background system as panel (e) (Appendix A.2), swept by the OU-derived amplitude instead of an arbitrary strength. The fast left factor is parallel to $\mathbf{u}$, at 45° in the $(\mathbf{u}, \mathbf{u}^\perp)$ plane, or parallel to $\mathbf{u}^\perp$, while the right factor $\mathbf{v}$ remains fixed. Dotted lines mark instability onset, calculated where the effective eigenvalue crosses zero.

structured-eigenvalue shift and relaxation time as the fast left factor's alignment with $\mathbf{u}$ varies. Full equations, parameter choices, and derivations for Fig. 2 are presented in Appendix A.2.

4 Fast weights in associative-recall use slow weights for writing direction

To understand how fast weights interact with the structural weight components, we study a fast-weight system with rank-one key–value updates (Schmidhuber, 1992; Ba et al., 2016). We test a shared-right address geometry in a trained associative-recall probe, with $J = UDV^\top$ and $A_t = K_tV^\top$. Address coordinates a_t and write content w_t determine the updates to K_t.

We compare vanilla fast-weight writing (Ba et al., 2016) with a delta-rule adaptation (Widrow & Hoff, 1960; Schlag et al., 2021; Yang et al., 2024b). Previous work has shown that local prediction alignment with fixed weights can tame chaos in vanilla RNNs (Asabuki & Clopath, 2025). To assess

the functional role of the slow-weight backbone in associative recall, we also include a No-J delta control. Appendix A.4 derives how rank-one updates, projected onto the parallel and perpendicular components of the slow content subspace, compose across a trial. We impose this geometry on the left/content state K_t relative to U. In the $K \perp U$ arm, the left projector removes the slow prediction UDa_t, but the delta rule retains its state-dependent erase term and therefore does not become identical to the vanilla rule.

We define bit error and target-signed logit margin from operator theory, via a recursive definition. Thus, we can obtain an analytical derivation of the statistics of an original read after a number of slots have passed (Appendix A.4). We compare the theoretical results (dotted line) with simulations from a recurrent system that presents fast and slow weights (solid lines; sequence mixer model in Appendix B). Hence, we trained rank-constrained ($R = 4$) and one-layer (width = 16 units) vanilla and delta fast weight models on a simple associative-recall task with $n = 2^3$ bits of actual write codes and random distractors along $T = 64$ steps. As before, the address space V is shared and trained parameter of shape (16×4), and K evolves according to either write and address coincidences (vanilla) or their corresponding errors with respect to previous fast and slow weights (delta).

In associative recall, our mixer model matches the operator-theory reconstruction using its trained parameters and input sequences (Fig. 3a). As the number of slots increases with new writes and distractors, the bit error of the trained vanilla mixer increases, while the delta rule restrains this drift. Particularly, when the fast-weight content is parallel to the slow-weight content basis, delta writing contains the error (left, Fig. 3a). This advantage is smaller when fast-weight content is perpendicular to the directions spanned by the slow weights (right, Fig. 3a), where vanilla also maintains a low error rate at shorter delays. Similarly, the target margin, which measures how strongly the output supports the correct recalled bits, is more stable under delta writing with the static backbone than under vanilla writing (bottom left, Fig. 3a), while the perpendicular condition shows declining margins for all three rules (bottom right, Fig. 3a).

Held-out recall bit accuracy decreases in a vanilla fast weight rule as slots progress, while delta rule decays more slowly under random-subspace alignment with the slow weight basis (Fig. 3b). We also looked at how the slowest eigenvalue of the total recurrent operator ($J + A_n$) as a relaxation timescale, governed by the proximal map (Appendix A.4). We found that aligned delta writing produces a sustained increase in the relaxation timescale relative to J alone (Fig. 3c), comparable to orthogonal additions. This is consistent with delta error correction supporting memory maintenance by sustaining slowly relaxing modes within the slow content subspace whereas a key-value binding fast weight kernel has to displace early in the orthogonal subspace to produce better recall at high latency times.

Finally, we probe a possible division of labor by extending associative recall with a summation task. We train a single delta mixer to store scalar values under two keys and predict their sum when queried at the second key. The model must retain the key-value binding and use it to compute the answer. To test the contribution of the slow weights, we rotate J at inference. The accuracy of the arithmetic operation dropped (Fig. 3d). Nonetheless, the memory dynamics remained unaltered as measured from the hidden state itself. This points to a specialization in both weight-types, in which fast weights preserve recoverable associations while the slow-weight pathway supports their use in computation.

5 A FAST AND SLOW WEIGHT SSM HAS DISTINCT CONTRIBUTIONS TO RETRIEVAL AND PREDICTION

The geometrical and content-retrieval properties of a complementary weight system prompted us to develop a mixer unit of a SSM, FastSlow , based on the idea of a shared presynaptic or address space V^T. In light of the bio-inspired nature of our architecture, we introduced several comprehensive adaptations, explained in detail in Appendix B, among which are:

First, fast-weight systems in SSMs utilize dynamical matricial values as hidden states tracking key-value associations (Schlag et al., 2021; Irie & Gershman, 2025), where we use hidden units $h \in \mathbb{R}^n$ – i.e., a vectorial neural space. Second, and notwithstanding this, hidden states were sectioned into equal-sized parallel streams or heads, so the hidden state lives in $\mathbb{R}^{d,d_h}$, with each head acting as an independent memory partition (Dao & Gu, 2024; Arshia Afzal, 2026). We hypothesize this could

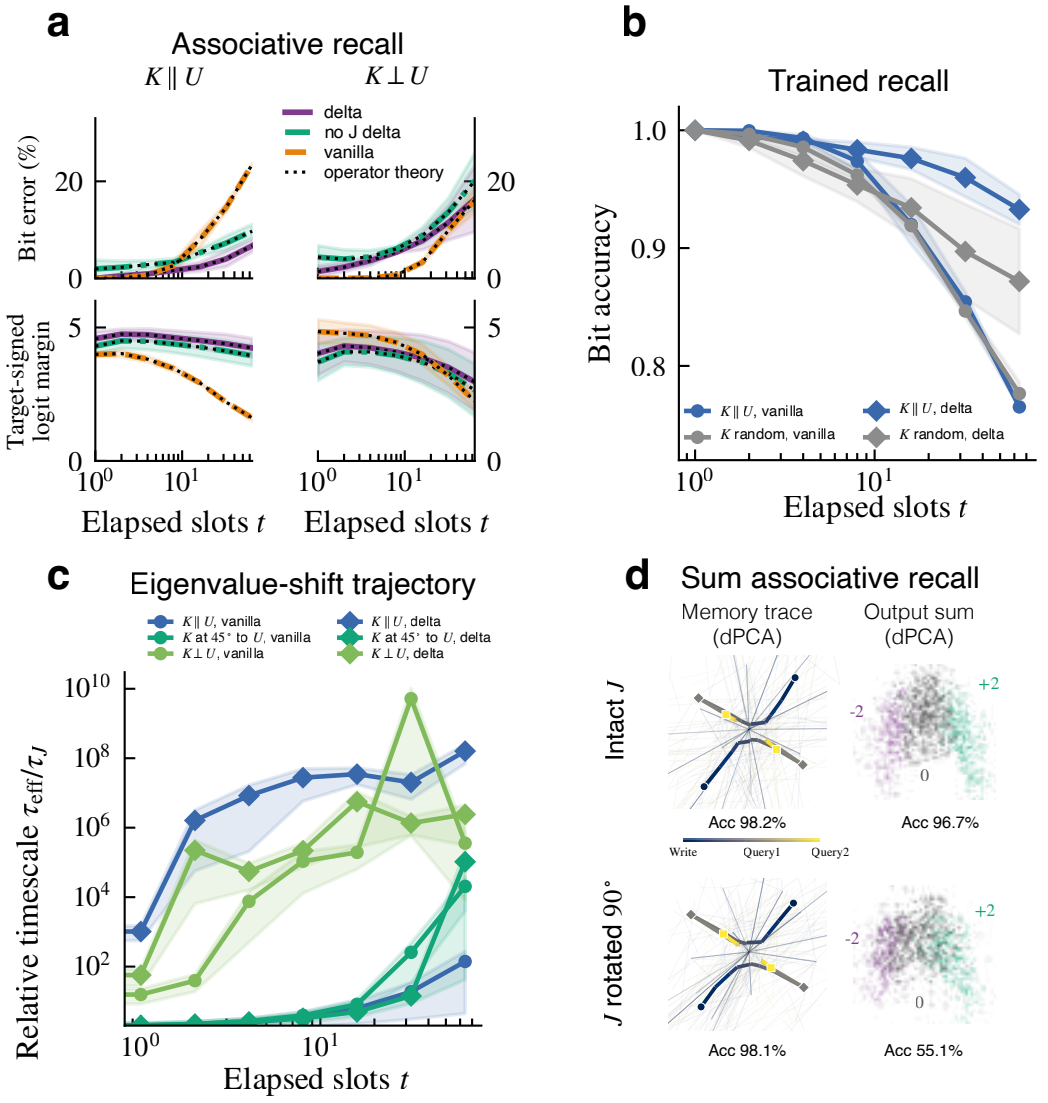


Figure 3: Geometry between fast and slow weights shapes recall and computation. **(a)** Bit error (top) and target-signed logit margin (bottom) for delta, vanilla, and no-J delta under aligned and orthogonal fast-content geometries. Dotted curves are exact operator theory reconstructions from trained parameters and held-out inputs. **(b)** Held-out bit accuracy for aligned geometry and a random-subspace control, where K_t is constrained to the span of $R = 4$ randomly selected coordinate axes in the 16-dim hidden space. Subspace is fixed per seed and shared by the vanilla and delta models. **(c)** Relative proximal relaxation timescale $\tau_{\text{eff}}(J + A_t)/\tau_{\text{eff}}(J)$ (see Appendix A.4) across writes. In (b,c), circles denote vanilla and diamonds delta. Associative recall uses $d = 16$, $R = 4$ mixers trained for 2,000 steps; lines show means and shading indicates ±SEM across $n = 3$ seeds. Held-out trials = 512 per seed and delay. **(d)** Summation of two recalled scalar values in a separately trained $d = 16$, $R = 16$ delta mixer (shown 1,165 trials with random inter-query delays). Fixed dPCA projections show memory-related hidden trajectories (left) and final hidden states colored by target sum (right), with intact or 90°-rotated J rotation. Memory: bold trajectories are mean of first write value; color tracks task progression, and segments show individual writes. Accuracies report a separate fast-memory probe averaged over both queries (left) and sum prediction (right). See Appendix A.4 for details.

Table 1: MAD task accuracies (%). Baseline rows are transcribed from MesaNet (Table 9 in von Oswald et al. (2026)).

Model	IC & Noisy	Fuzzy	Memorization	Selective Copy	Compression	Average
Mamba2	100	51.2	42.0	95.4	41.3	66.0
GLA	100	39.0	82.5	96.1	42.3	72.0
xLSTM	100	47.6	79.8	95.4	43.4	73.2
DeltaNet	100	55.5	40.8	98.8	43.3	67.7
Gated DeltaNet	100	32.7	81.7	95.7	45.0	71.0
Hawk	93.0	13.6	91.3	77.0	47.7	64.5
MesaNet	100	58.5	77.2	99.2	45.4	76.1
Hawk-MesaNet	100	30.2	85.6	99.6	52.3	73.5
Transformer	100	48.6	84.7	96.0	49.5	75.8
FastSlow (ours)	**99.28**	**17.49**	**91.20**	**97.09**	**37.01**	**73.52**

mirror functional areas that are highly disconnected from each other and produce specialization (Gershman et al., 2025). Finally, instead of computing h directly, we used a proximal approach (Liu et al., 2024; von Oswald et al., 2026). This intervention allows both an implicit solution of the hidden space through a solver – e.g., Cholesky – and mainly allows the SSM to be presented in a chunkwise format while preserving stability (Orvieto et al., 2023).

As mentioned, we force fast and slow weights to share right-hand side-modes or singular values. This means that instead of $(N \times N)$ parameters, we have $(N \times R)$ as in low-rank RNNs. The main difference is that here we save even more parameters by setting up a shared memory addressing space of $\mathbf{x}$. This is not only interpretable, but also permits direct subspace alignment between fixed slow dynamics and plastic memory updates. That is why we can do interventions on the specific shared address space. The schematic of the final model is available in Fig. A4.

We evaluated FastSlow on the MAD benchmark (Poli et al., 2024) and report published baseline accuracies (Table 1; we reproduce other model accuracies as reported by von Oswald et al. (2026); implementation and evaluation details given in Appendix C). We compare against state-of-the-art selective and gated recurrent models (Dao & Gu, 2024; De et al., 2024; Beck et al., 2024), linear-attention and fast-weight models (Yang et al., 2024b;a), test-time learning models (von Oswald et al., 2026), and the Transformer baseline (Vaswani et al., 2017). Across these model classes, FastSlow obtains a mean MAD accuracy of 73.52%, with particularly strong performance on memorization (91.20%) and selective copying (97.09%). We report these as contextual comparisons rather than a controlled ranking, since the baseline numbers are transcribed from a published table under a different training protocol (Appendix C).

We wondered what functional role the slow weights were adopting in these tasks. Therefore, we performed an ablation study on J in the hidden state dynamics – the drive J_{drive}– and in producing the local error of the delta rule – the predictive J_{pred}. We found that on average the slow weights do not affect the accuracies of MAD tasks in contrast with the effect of removing fast weights, yet this effect was strongly task-dependent (Table 2). Compression and selective copying showed the clearest dependence on J_{drive}, which decreased the accuracy by 10 % and 21 %, respectively, while J_{pred} produced decreases of 3 % in both tasks. On the opposite side, we found recall and memorization were largely unaltered by either of the slow-weight ablations. Notably, the latter seems to be insensitive to both J and A ablations, which points to either pathway redundancy in this task or a dependence on the residual path. Taken together, these ablations are consistent with A supporting content-addressable memory and J contributing when information is not directly recallable but has associated sequential transformations, as is the case in compression and copying. Interestingly, this effect appears to stem from shaping the latent state itself, rather than delta prediction error. We cannot rule out that this role is due to an 'offloading' effect, where the fixed weights acquire a long-term functional role, as in copying, while fast weights specialize solely in test-time association of values.

As an exploratory study, we trained a two-layer FastSlow on a small-corpus consisting of 1-billion-token subset of FineWeb-Edu (Penedo et al., 2024) to test this division of labor. Then, we tested our SSM in two in-context tasks with post training interventions in J and A. Given the limited corpus and single training run, we did not aim to run full benchmarking with baselines but rather aimed to gain intuition of how each weight type is parsing information. On single-needle prompts from RULER

Table 2: MAD validation accuracy and post-training ablations. Task rows show mean ± descriptive SEM across fixed difficulty settings (n in parentheses). Intact is accuracy in percent; ablations are paired accuracy changes from intact FastSlow in percentage points (pp). The final row averages the six task means equally.

Task (n)	Intact (%)	$J_{\text{drive}} = 0$ Δpp	$J_{\text{pred}} = 0$ Δpp	$J = 0$ Δpp	$A = 0$ Δpp	Hebbian writes Δpp
In-context recall (11)	99.0 ± 0.7	−0.8 ± 0.4	−0.04 ± 0.02	−0.5 ± 0.2	−92 ± 1	−66 ± 10
Noisy recall (14)	99.3 ± 0.5	−0.3 ± 0.2	−0.02 ± 0.02	−0.2 ± 0.1	−91 ± 1	−87 ± 2
Fuzzy recall (11)	17 ± 2	−1.0 ± 0.5	−0.6 ± 0.4	−1.0 ± 0.5	−9 ± 2	−9 ± 2
Memorization (6)	91 ± 6	−0.10 ± 0.09	−0.004 ± 0.004	−0.10 ± 0.09	−0.6 ± 0.6	−0.3 ± 0.3
Compression (11)	37 ± 6	−10 ± 3	−3 ± 2	−7 ± 2	−30 ± 5	−26 ± 6
Selective copying (13)	97 ± 3	−21 ± 7	−3 ± 2	−18 ± 7	−91 ± 3	−91 ± 3
Mean across tasks	73.5	-5.5	-1.2	-4.5	-52.2	-46.5

Table 3: Single-needle RULER prompts, evaluated after 1B pretraining tokens. Margin is the mean per-token log-likelihood of the true needle minus a matched absent foil (nats). Preference is the fraction of examples where the true needle scores higher. Ablation columns are paired changes from intact; preference changes are in percentage points (pp). Each length pools $n = 96$ examples across three tasks.

	Intact		$A = 0$: change		$J = 0$: change	
Context	Margin	Preference (%)	Δmargin	Δpp	Δmargin	Δpp
1K	+1.03	88.54	−1.10	−40.63	+0.06	0.00
2K	+0.82	85.42	−0.82	−34.38	+0.05	+1.04
4K	+0.34	67.71	−0.29	−14.58	+0.06	−2.08
8K	+0.27	64.58	−0.24	−17.71	−0.01	0.00
16K	+0.02	47.92	−0.10	+1.04	+0.03	−2.08

(Hsieh et al., 2024), we measured foil-controlled likelihood margin (per-token log-likelihood of a true needle value minus an equally formatted value not in context) as generated-answer accuracy was low in this small-corpus training. We find that removing fast weights eliminated the short-context retrieval signal, whereas removing slow weights did not (Table 3).

6 Discussion

Our theoretical work complements classical dynamical analysis of RNNs (Sompolinsky et al., 1988). In contrast with Hebbian or anti-Hebbian forms of fast weights (Clark & Abbott, 2024; Wakhloo et al., 2025; Gaudio et al., 2026), we show a kernel-agnostic definition of fast weights derived from the structural decomposition of presynaptic and postsynaptic terms (Mastrogiuseppe & Ostojic, 2018). To our knowledge, the microscopic description of contact-gated processes as coupling factors between neurons provides a first account of fast-weight emergence: fast weights arise as transient fluctuations of an underlying structured connectivity, rather than as a separately prescribed plastic operator. In this view, the stationary component defines the slow recurrent backbone, whereas temporal fluctuations modulate its postsynaptic expression while preserving a common presynaptic address space.

Fast weights may have a biological substrate in dendritic computation. Dendrites provide nonlinear, compartmentalized, and state-dependent integration capable of supporting local plasticity and content-addressable memory (Gordon et al., 2006; Larkum, 2013; Poirazi & Papoutsi, 2020; Sacramento et al., 2018; Wu & Maass, 2025). This makes transient, postsynaptically organized changes in effective connectivity a plausible implementation of fast weights, with local predictive rules such as the delta rule providing a candidate mechanism for online adaptation (Gordon et al., 2006; Asabuki & Clopath, 2025). In fact, our mathematical framework assumes a fixed presynaptic basis, which refers to the fixed wiring neural systems present. For delta rule, in particular, previous work on RNNs has also found this produces a biologically plausible rule for online learning (Asabuki & Clopath, 2025). In other words, fast weights operate beyond architectural or structural constraints

as they are processes situated on the postsynaptic side and acting as coupling terms between the raw presynaptic activation and the postsynaptic end process. Other theories pose that biological fast weights arise from interactions between areas (Gershman et al., 2025), in which case our coupling term would reflect more the tension between private or in-area computations and inter-area sparse and directed activity.

The idea of having in-context or test-time learning through an update rule that is subspace-aligned or uses curvature-aware preconditioners in SSMs has gained traction in the community (Behrouz et al., 2025; von Oswald et al., 2026). In normative update rule design, the avoidance of dimensionality collapse through a preconditioner seems to improve learning (Jordan et al., 2024; Boeshertz et al., 2026; Nguyen et al., 2026). Other geometrical approaches avoid this collapse entirely in the mixer section of the SSM by using orthogonal updates (Karami et al., 2025) or rotational systems (Su et al., 2024). We believe our system sets up a promising path towards a cascade-type update rule (Benna & Fusi, 2016; Kaplanis et al., 2018), where the weight transport problem translates into what fast parameters ought to be consolidated or transferred to the slow mean-modes. Future work can use this framework to capture the tension between recyclable and compositional dynamics and stability over the parameter-space.

Our results point to a limited yet diverse role of slow weights in SSM computations. We do see benefits in learning stability, especially when used in concert with error-correcting delta-type rules, and when there are output computations. For instance, the load-bearing of the computation seems to increase when an upstream computation, like a key-associated value summation, is required. Also, from MAD ablations, we see a similar case in copying or compression. Our pretraining small-corpus, even if limited in scope, points a larger role of fast-weights in general in the effect to a benchmark of recall specificity.

As the field has moved from classical RNNs to fast-weight systems, few attempts have bridged the two. Our work offers a model and a theory relating them, one that helps clarify and interpret function from a geometrical perspective.

References

Eric P. Xing Volkan Cevher Albert Gu Arshia Afzal, Aviv Bick. Raven: High-Recall Sequence Modeling with Sparse Memory Routing. 2026.

Toshitake Asabuki and Claudia Clopath. Taming the chaos gently: a predictive alignment learning rule in recurrent neural networks. *Nature Communications*, 16(1):6784, July 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-61309-9. URL https://www.nature.com/articles/s41467-025-61309-9.

Jimmy Ba, Geoffrey Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. Using Fast Weights to Attend to the Recent Past, 2016. URL https://arxiv.org/abs/1610.06258. Version Number: 3.

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*, volume 37, 2024.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. Nested Learning: The Illusion of Deep Learning Architectures, 2025. URL https://arxiv.org/abs/2512.24695. Version Number: 1.

Marcus K Benna and Stefano Fusi. Computational principles of synaptic memory consolidation. *Nature Neuroscience*, 19(12):1697–1706, December 2016. ISSN 1097-6256, 1546-1726. doi: 10.1038/nn.4401. URL https://www.nature.com/articles/nn.4401.

Brendan A Bicknell and Peter E Latham. Fast and slow synaptic plasticity enables concurrent control and learning, March 2025. URL https://elifesciences.org/reviewed-preprints/105043v1.

Gauthier Boeshertz, Razvan Pascanu, and Claudia Clopath. Overcoming rank collapse in feedback alignment. *arXiv preprint arXiv:2606.11123*, 2026.

Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying Sparse and Low-rank Attention Approximation, October 2021. URL http://arxiv.org/abs/2110.15343. arXiv:2110.15343 [cs.LG].

David G. Clark and L. F. Abbott. Theory of Coupled Neuronal-Synaptic Dynamics. *Physical Review X*, 14(2):021001, April 2024. ISSN 2160-3308. doi: 10.1103/PhysRevX.14.021001. URL https://link.aps.org/doi/10.1103/PhysRevX.14.021001.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality, May 2024. URL http://arxiv.org/abs/2405.21060. arXiv:2405.21060 [cs.LG].

Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, Arnaud Doucet, David Budden, Yee Whye Teh, Razvan Pascanu, Nando De Freitas, and Caglar Gulcehre. Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models, 2024. URL https://arxiv.org/abs/2402.19427. Version Number: 1.

Matthew Dowling, Hyungju Jeon, Cristina Savin, and Il Memming Park. Memory by Design: Probabilistic Sequence Layers, 2026. URL https://arxiv.org/abs/2605.31163. Version Number: 1.

Martina Del Gaudio, Federico Ghimenti, and Surya Ganguli. Short-term plasticity recalls forgotten memories through a trampoline mechanism, February 2026. URL http://arxiv.org/abs/2511.22848. arXiv:2511.22848 [q-bio].

Samuel J. Gershman, Ila Fiete, and Kazuki Irie. Key-value memory in the brain. *Neuron*, 113(11): 1694–1707.e1, June 2025. ISSN 08966273. doi: 10.1016/j.neuron.2025.02.029. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627325001722.

Urit Gordon, Alon Polsky, and Jackie Schiller. Plasticity Compartments in Basal Dendrites of Neocortical Pyramidal Neurons. *The Journal of Neuroscience*, 26(49):12717–12726, December 2006. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.3502-06.2006. URL https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.3502-06.2006.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently Modeling Long Sequences with Structured State Spaces, 2021. URL https://arxiv.org/abs/2111.00396. Version Number: 3.

Geoffrey E Hinton and David C Plaut. Using fast weights to deblur old memories. In *Proceedings of the ninth annual conference of the cognitive science society*, pp. 177–186, 1987.

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?, August 2024. URL http://arxiv.org/abs/2404.06654. arXiv:2404.06654 [cs.CL].

Kazuki Irie and Samuel J. Gershman. Fast weight programming and linear transformers: from machine learning to neurobiology, 2025. URL https://arxiv.org/abs/2508.08435. Version Number: 4.

Kazuki Irie, Morris Yau, and Samuel J. Gershman. Blending Complementary Memory Systems in Hybrid Quadratic-Linear Transformers, October 2025. URL http://arxiv.org/abs/2506.00744. arXiv:2506.00744 [cs.LG].

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Christos Kaplanis, Murray Shanahan, and Claudia Clopath. Continual Reinforcement Learning with Complex Synapses. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2497–2506. PMLR, July 2018. URL https://proceedings.mlr.press/v80/kaplanis18a.html.

Mahdi Karami, Razvan Pascanu, and Vahab Mirrokni. Lattice: Learning to Efficiently Compress the Memory, December 2025. URL http://arxiv.org/abs/2504.05646. arXiv:2504.05646 [cs.LG].

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention, 2020. URL https://arxiv.org/abs/2006.16236. Version Number: 3.

Dmitry Kobak, Wieland Brendel, Christos Constantinidis, Claudia E Feierstein, Adam Kepecs, Zachary F Mainen, Xue-Lian Qi, Ranulfo Romo, Naoshige Uchida, and Christian K Machens. Demixed principal component analysis of neural population data. *eLife*, 5:e10989, April 2016. ISSN 2050-084X. doi: 10.7554/eLife.10989. URL https://elifesciences.org/articles/10989.

Matthew Larkum. A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in Neurosciences*, 36(3):141–151, March 2013. ISSN 01662236. doi: 10.1016/j.tins.2012.11.006. URL https://linkinghub.elsevier.com/retrieve/pii/S0166223612002032.

Bo Liu, Rui Wang, Lemeng Wu, Yihao Feng, Peter Stone, and Qiang Liu. Longhorn: State Space Models are Amortized Online Learners, 2024. URL https://arxiv.org/abs/2407.14207. Version Number: 5.

Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. nGPT: Normalized Transformer with Representation Learning on the Hypersphere, 2024. URL https://arxiv.org/abs/2410.01131. Version Number: 2.

Francesca Mastrogiuseppe and Srdjan Ostojic. Linking Connectivity, Dynamics, and Computations in Low-Rank Recurrent Neural Networks. *Neuron*, 99(3):609–623.e29, August 2018. ISSN 08966273. doi: 10.1016/j.neuron.2018.07.003. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627318305439.

Thomas Miconi, Jeff Clune, and Kenneth O. Stanley. Differentiable plasticity: training plastic neural networks with backpropagation. 2018. doi: 10.48550/ARXIV.1804.02464. URL https://arxiv.org/abs/1804.02464. Version Number: 3.

Thai-Khanh Nguyen, Ngoc-Bich-Uyen Vo, Thieu N. Vo, Tan M. Nguyen, and Cuong Pham. MuonSSM: Orthogonalizing State Space Models for Sequence Modeling, June 2026. URL https://arxiv.org/abs/2606.30461. Version Number: 1.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting Recurrent Neural Networks for Long Sequences, 2023. URL https://arxiv.org/abs/2303.06349. Version Number: 1.

Guilherme Penedo, Hynek Kydlıček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *The thirty-eighth annual conference on neural information processing systems datasets and benchmarks track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Panayiota Poirazi and Athanasia Papoutsi. Illuminating dendritic function with computational models. *Nature Reviews Neuroscience*, 21(6):303–321, June 2020. ISSN 1471-003X, 1471-0048. doi: 10.1038/s41583-020-0301-7. URL https://www.nature.com/articles/s41583-020-0301-7.

Michael Poli, Armin W Thomas, Eric Nguyen, Pragaash Ponnusamy, Björn Deiseroth, Kristian Kersting, Taiji Suzuki, Brian Hie, Stefano Ermon, Christopher Ré, Ce Zhang, and Stefano Massaroli. Mechanistic Design and Scaling of Hybrid Architectures, 2024. URL https://arxiv.org/abs/2403.17844. Version Number: 2.

João Sacramento, Rui Ponte Costa, Yoshua Bengio, and Walter Senn. Dendritic cortical microcircuits approximate the backpropagation algorithm, 2018. URL https://arxiv.org/abs/1810.11393. Version Number: 1.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear Transformers Are Secretly Fast Weight Programmers, June 2021. URL http://arxiv.org/abs/2102.11174. arXiv:2102.11174 [cs].

Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992. doi: 10.1162/neco.1992.4.1.131.

Friedrich Schuessler, Alexis Dubreuil, Francesca Mastrogiuseppe, Srdjan Ostojic, and Omri Barak. Dynamics of random recurrent networks with correlated low-rank structure. *Physical Review Research*, 2(1):013111, February 2020. ISSN 2643-1564. doi: 10.1103/PhysRevResearch.2.013111. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.013111.

Friedrich Schuessler, Francesca Mastrogiuseppe, Srdjan Ostojic, and Omri Barak. Aligned and oblique dynamics in recurrent neural networks. *eLife*, 13:RP93060, November 2024. ISSN 2050-084X. doi: 10.7554/eLife.93060.3. URL https://elifesciences.org/articles/93060.

H. Sompolinsky, A. Crisanti, and H. J. Sommers. Chaos in Random Neural Networks. *Physical Review Letters*, 61(3):259–262, July 1988. ISSN 0031-9007. doi: 10.1103/PhysRevLett.61.259. URL https://link.aps.org/doi/10.1103/PhysRevLett.61.259.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding, 2021. URL https://arxiv.org/abs/2104.09864. Version Number: 5.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, February 2024. ISSN 09252312. doi: 10.1016/j.neucom.2023.127063. URL https://linkinghub.elsevier.com/retrieve/pii/S0925231223011864.

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 http://arxiv.org/abs/2407.04620. arXiv:2407.04620 [cs.LG].

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient Transformers: A Survey. *ACM Computing Surveys*, 55(6):1–28, June 2023. ISSN 0360-0300, 1557-7341. doi: 10.1145/3530811. URL https://dl.acm.org/doi/10.1145/3530811.

Neehal Tumma, Noel Loo, and Daniela Rus. Preconditioned DeltaNet: Curvature-aware Sequence Modeling for Linear Recurrences, April 2026. URL http://arxiv.org/abs/2604.21100. arXiv:2604.21100 [cs.LG].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, 2017. URL https://arxiv.org/abs/1706.03762. Version Number: 7.

Johannes von Oswald, Nino Scherrer, Seijin Kobayashi, Luca Versari, Songlin Yang, Maximilian Schlegel, Kaitlin Maile, Yanick Schimpf, Oliver Sieberling, Alexander Meulemans, Guillaume Lajoie, Rif A. Saurous, Charlotte Frenkel, Razvan Pascanu, Blaise Aguera y Arcas, and Joao Sacramento. MesaNet: Sequence modeling by locally optimal test-time training. In *The fourteenth international conference on learning representations*, 2026. URL https://openreview.net/forum?id=xa30nTb6c3.

Albert J Wakhloo, David G Clark, and Lf Abbott. Associative synaptic plasticity creates dynamic persistent activity, August 2025. URL http://biorxiv.org/lookup/doi/10.1101/2025.08.20.671131.

Bernard Widrow and Marcian E. Hoff. Adaptive Switching Circuits. In *1960 IRE WESCON Convention Record, Part 4*, pp. 96–104, New York, 1960. IRE.

Yujie Wu and Wolfgang Maass. Author Correction: A simple model for Behavioral Time Scale Synaptic Plasticity (BTSP) provides content addressable memory with binary synapses and one-shot learning. *Nature Communications*, 16(1):1150, January 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-56459-9. URL https://www.nature.com/articles/s41467-025-56459-9.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated Delta Networks: Improving Mamba2 with Delta Rule, 2024a. URL https://arxiv.org/abs/2412.06464. Version Number: 3.

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. Version Number: 6.

Yifei Zuo, Yutong Yin, Zhichen Zeng, Ang Li, Banghua Zhu, and Zhaoran Wang. Local linear attention: An optimal interpolation of linear and softmax attention for test-time regression. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International conference on learning representations*, volume 2026, pp. 153843–153872, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/f9af970430ac401594ce01022fccb049-Paper-Conference.pdf.

Appendix A Coupling or contact-gated as fast and slow low-rank weights

Throughout, $\mathbf{u}^{(r)}$ is the *postsynaptic* / left loading (content side) and $\mathbf{v}^{(r)}$ the *presynaptic* / right loading (address side) of the r-th rank-one mode; $U, V \in \mathbb{R}^{N\times R}$ stack them columnwise.

A.1 Contact gate, mean-fluctuation split, and the fast-weight equation

Contact model. Each synapse $j \to i$ carries R contacts with structural weight

$$w_{ij}^{(r)} = \frac{u_i^{(r)} v_j^{(r)}}{N} + \frac{\sigma_w}{\sqrt{N}} \chi_{ij}^{(r)}, \qquad \chi_{ij}^{(r)} \sim \mathcal{N}(0,1) \text{ i.i.d.}, \tag{A.6}$$

and an efficacy $\nu_{ij}(t) = \sum_r c_{ij}^{(r)}(t)\, w_{ij}^{(r)}$ set by a per-contact gate obeying the leaky first-order equation

$$p\, \dot{c}_{ij}^{(r)} = -c_{ij}^{(r)} + \eta_i^{(r)}(t) + \zeta_{ij}^{(r)}(t), \qquad \mathbb{E}[\zeta_{ij}^{(r)}(t)\zeta_{i'j'}^{(r')}(s)] = \sigma_\zeta^2\, \delta_{ii'}\delta_{jj'}\delta_{rr'}\delta(t-s). \tag{A.7}$$

The construction of the coupling factor $c_{ij}^{(r)}(t)$ rests on two elements: a postsynaptic common drive and an independent per-contact residual. Only the drive can generate structured, low-rank fast weights, since it is shared by every contact of neuron i in mode r. Thus, the residual averages away at large N (A.3) and contributes only unstructured noise to the bulk. We chose to keep this source of variability to account for synaptic selectivity – e.g., via branch effect or sparsity – in future constructions.

The drive $\eta_i^{(r)}$ is a postsynaptic process, so every contact of mode r on neuron i is steered by the same common mode signal – akin to a dendritic process in a postsynaptic neuron. The residual $\zeta_{ij}^{(r)}$ is white and independent across contacts: it records how far an individual contact falls out of line with its common drive, and its independence is what makes contact averaging exact at large N (A.3). Since (A.7) is linear in the drive and noise, and $\eta_i^{(r)}$ is shared across contacts while $\zeta_{ij}^{(r)}$ is independent across them, its causal solution splits additively into a drive-only and a noise-only filtered term. Filtering (A.7) gives $c_{ij}^{(r)} = c_i^{(r)} + \delta_{ij}^{(r)}$, where $c_i^{(r)}$ is the gate driven by $\eta_i^{(r)}$ alone and $\delta_{ij}^{(r)}$ is the filtered contact residual (OU process with covariance $D_\delta(t,s) = \sigma_\zeta^2 e^{-|t-s|/p}/(2p)$). The latter is the part of contact gating in j coming only from its own private noise $\zeta_{ij}^{(r)}$ – i.e., how far that one contact drifts from the shared gate $c_i^{(r)}$.

Mean-centred and fluctuating modes. Split the drive into its stationary mean and a centred fluctuation,

$$\eta_i^{(r)}(t) = \bar{\eta}^{(r)} + \tilde{\eta}_i^{(r)}(t), \qquad \tau_{\eta,r}\, \dot{\tilde{\eta}}_i^{(r)} = -\tilde{\eta}_i^{(r)} + \sqrt{2\tau_{\eta,r}}\, \sigma_{\eta,r}\, \xi_i^{(r)}(t), \tag{A.8}$$

with $\tilde{\eta}_i^{(r)}$ an Ornstein–Uhlenbeck (OU) process of stationary variance $\sigma_{\eta,r}^2$ and correlation time $\tau_{\eta,r}$, independent across neurons and modes. In steady state the gate mean is $\mathbb{E}[c_{ij}^{(r)}] = \bar{\eta}^{(r)}$. The mean therefore rescales the structural motif by a constant and defines the slow connectivity,

$$J = \frac{1}{N}\sum_r \bar{\eta}^{(r)} \mathbf{u}^{(r)} (\mathbf{v}^{(r)})^\top + \frac{\sigma_w}{\sqrt{N}}\boldsymbol{\chi} = \frac{1}{N} U D V^\top + \frac{\sigma_w}{\sqrt{N}}\boldsymbol{\chi}, \qquad D = \mathrm{diag}(\bar{\eta}^{(1)}, \ldots, \bar{\eta}^{(R)}), \tag{A.9}$$

Only the R-structured modes $w_{ij}^{(r)} = u_i^{(r)} v_j^{(r)}/N$ in (A.6) carry a low-rank motif, so the time-varying part of the connectivity is entirely carried by the centred gate acting on these modes. The centred part of the gate, projected onto the rank-R skeleton, defines the fast-weight matrix

$$A_{ij}(t) := \frac{1}{N}\sum_{r=1}^{R} u_i^{(r)} v_j^{(r)} (c_{ij}^{(r)}(t) - \bar{\eta}^{(r)}). \tag{A.10}$$

Fast-weight dynamics. Only $c_{ij}^{(r)}$ in (A.10) is time-dependent. Since A_{ij} is linear in the state variables $c_{ij}^{(r)}$, Itô's lemma reduces to ordinary term-by-term differentiation, with no quadratic-variation correction.

Substituting (A.8) into (A.7) gives

$$p\,\dot{c}_{ij}^{(r)} = -(c_{ij}^{(r)} - \bar{\eta}^{(r)}) + \tilde{\eta}_i^{(r)} + \zeta_{ij}^{(r)}. \tag{A.11}$$

Differentiating $A_{ij} = \frac{1}{N}\sum_r u_i^{(r)} v_j^{(r)} (c_{ij}^{(r)} - \bar{\eta}^{(r)})$ and substituting (A.11),

$$p\,\dot{A}_{ij} = \underbrace{-\frac{1}{N}\sum_r u_i^{(r)} v_j^{(r)} (c_{ij}^{(r)} - \bar{\eta}^{(r)})}_{A_{ij}} + \frac{1}{N}\sum_r (\tilde{\eta}_i^{(r)} u_i^{(r)}) v_j^{(r)} + \underbrace{\frac{1}{N}\sum_r u_i^{(r)} v_j^{(r)} \zeta_{ij}^{(r)}}_{\text{entrywise intensity } O(N^{-2})}. \tag{A.12}$$

The last term has white-noise intensity $\sigma_\zeta^2 \sum_r (u_i^{(r)} v_j^{(r)})^2 / N^2$ and its contribution to any neural field vanishes as $O_p(N^{-1/2})$ (A.3). Defining the fast left vector of mode r as the elementwise modulation

$$\mathbf{k}_A^{(r)}(t) := \tilde{\boldsymbol{\eta}}^{(r)}(t) \circ \mathbf{u}^{(r)}, \tag{A.13}$$

we obtain the macroscopic fast-weight equation

$$\boxed{p\,\dot{A} = -A + \frac{1}{N}\sum_{r=1}^{R} \mathbf{k}_A^{(r)}(t)\,(\mathbf{v}^{(r)})^\top, \qquad A(t) = \frac{1}{N} K(t) V^\top, \quad p\,\dot{K} = -K + F(t),} \tag{A.14}$$

where $F(t)$ stacks the $\mathbf{k}_A^{(r)}(t)$ columnwise and K (written K_A, or $K_{A,t}$ in discrete time, in Appendix C) is their leaky-integrated left state. K is thus a first-order low-pass filter of $F(t)$ with time constant p, decaying exponentially toward the current drive rather than integrating it exactly (discrete form in Eq. (A.22)). A is the *filtered* output of the rank-R drive, not its derivative; it shares the right/presynaptic basis V with J exactly and differs only in the left factor, $K(t)$ versus UD.

For simplicity, in other sections and in the main text (Eq. (5)), the explicit $1/N$ prefactor of (A.9) and (A.14) is absorbed into U and K.

Where the variance goes. The stationary causal solution of (A.7) is $c(t) = p^{-1}\int_0^\infty e^{-a/p}[\eta(t-a) + \zeta(t-a)]\,da$. For a stationary drive with autocorrelation C_η and uncorrelated contact noise, splitting the double integral into $a \geq b$ and $b \geq a$ gives $p^{-2}\iint e^{-(a+b)/p} C(a-b)\,da\,db = p^{-1}\int_0^\infty e^{-s/p} C(s)\,ds$, hence

$$\mathrm{Var}(c_{ij}^{(r)}) = \frac{1}{p}\int_0^\infty e^{-s/p}[C_{\tilde{\eta}^{(r)}}(s) + C_\zeta(s)]ds = \frac{\sigma_{\eta,r}^2\,\tau_{\eta,r}}{p + \tau_{\eta,r}} + \frac{\sigma_\zeta^2}{2p}, \tag{A.15}$$

where the second equality uses the exponential autocorrelation of the OU drive (A.8), $C_{\tilde{\eta}^{(r)}}(s) = \sigma_{\eta,r}^2 e^{-s/\tau_{\eta,r}}$, and white contact noise. For the latter, only half of the delta mass at $s = 0$ contributes: a causal filter integrates only the past ($a \geq 0$), so the equal-time overlap of two such filters driven by the same white noise is $\int_0^\infty e^{-2a/p}\,da = p/2$, half the two-sided value. The variance plotted in Fig. 2c is the case $\sigma_\zeta = 0$.

This completes the split. The gate mean sets the slow spectrum D, while the gate variance sets the fast-weight budget – i.e., the total fluctuation that can be expressed as structured A rather than unstructured bulk noise. The time constant p trades off tracking $\tilde{\eta}$ against averaging out ζ. A firing-driven gate, $\eta_i^{(r)} = a_H^{(r)}\phi(x_i)$, where ϕ is the activation function, obeys the same filtering identity, but its $C_{\tilde{\eta}}$ is determined jointly with the rate process, so only the integral form in (A.15) applies.

Left-factor geometry. Because A adds no presynaptic dimensions, its geometry is controlled by the alignment of each fast left vector with the slow one. For a uniform drive, $\tilde{\boldsymbol{\eta}}^{(r)} \propto \mathbf{1}$ and $\mathbf{k}_A^{(r)} \propto \mathbf{u}^{(r)}$: A merely rescales motifs J already encodes. For a heterogeneous drive, $\mathbf{k}_A^{(r)}$ acquires components in $(\mathbf{u}^{(r)})^\perp$, with

$$\cos\angle(\mathbf{k}_A^{(r)}, \mathbf{u}^{(r)}) := \frac{(\mathbf{k}_A^{(r)})^\top \mathbf{u}^{(r)}}{\|\mathbf{k}_A^{(r)}\|\,\|\mathbf{u}^{(r)}\|}, \tag{A.16}$$

substituting $\mathbf{k}_A^{(r)} = \tilde{\boldsymbol{\eta}}^{(r)} \circ \mathbf{u}^{(r)}$ gives the elementwise form $\sum_i (u_i^{(r)})^2 \tilde{\eta}_i^{(r)} / \sqrt{\sum_i (u_i^{(r)})^2}\,\sqrt{\sum_i (u_i^{(r)})^2 (\tilde{\eta}_i^{(r)})^2}$. R-independent drives give $A(t)$ full rank-R expressivity in a postsynaptic subspace that J does not span (Fig. 2).

A.2 OU specialization and outlier eigenvalue trajectory

In Fig. 2c we use the centered, externally specified OU-driven component of the gate and set the contact-noise intensity – i.e., the one referring to structural variability – to zero, $\sigma_\zeta = 0$. Thus, we can take (A.15) and exclude a nonzero gate mean.

Writing b_i for this filtered component,

$$p\dot{b}_i = -b_i + \tilde{\eta}_i, \qquad \mathrm{Var}(b_i) = \frac{\sigma_\eta^2 \tau_\eta}{p + \tau_\eta}, \tag{A.17}$$

which is the contact-noise-free term of (A.15), excluding a nonzero gate mean, contact noise, and neural-feedback self-consistency.

We follow Schuessler et al. (2020) for the outlier eigenvalues of a rank-one update to a random bulk. Linearizing around a fixed point x^*, using the same background as Eq. (5) (single mode, $R = 1$),

$$J_0 = -I + \Big(\frac{\sigma_w}{\sqrt{N}}\boldsymbol{\chi} + \frac{1}{N}\mathbf{u}\mathbf{v}^\top\Big)\,\mathrm{diag}(\phi'(x^*)), \qquad M(f) = J_0 + \frac{f}{N}\,\mathbf{k}_A\mathbf{v}^\top, \tag{A.18}$$

where f interpolates between the steady-state or slow weight case ($f = 0$) and the fast-weight perturbation. The $M(f)$ outlier eigenvalue is exactly the quantity plotted on the y-axis of Fig. 2e–f. Schuessler et al. (2020) solve for the outlier via a mean-field self-consistent condition ($N \to \infty$). For simplicity, we diagonalize $M(f)$ instead and track its structured eigenvalue by nearest-eigenvalue continuation. This means that at each f we take the eigenvalue of $M(f)$ closest in the complex plane to the one found at the previous f in the sweep, rather than solving self-consistently for the outlier root. We show $N = 20$ in the Fig. 2 example.

Both panels Fig. 2e-f plot a J_0 ($N = 20$, $\sigma_w = 0.92$, $\tau = 10$, $\Delta t = 0.2$, input $0.15\,\mathbf{v}$, $\mathbf{u} \perp \mathbf{v}$ from a fixed seed) and the same $\mathbf{k}_A \in \{\mathbf{u}, (\mathbf{u} + \mathbf{u}_\perp)/\sqrt{2}, \mathbf{u}_\perp{:=}\mathbf{v}\}$, i.e., we take $\mathbf{v}$ itself as the orthogonal direction $\mathbf{u}_\perp$, since $\mathbf{u} \perp \mathbf{v}$. Fig. 2e sweeps $f \in [-2.5, 2.5]$ directly; Fig. 2f sweeps the OU amplitude $f(p) = 2.5\sqrt{\mathrm{Var}(b_i)}$ (Eq. (A.15), $\tau_\eta = 8, \sigma_\eta = 1$) against p/τ_η, with the same target eigenvalue as panel e.

A.3 Conditional DMFT closure and numerical check

We do not solve the resulting self-consistent equations for $\kappa_r(t)$ and $C_\phi(t, s)$ in closed form, as it is outside the scope of this paper. Fig. 2d instead validates numerically that the conditional closure below holds for our contact-gating model.

Effective field. With rates $\phi_j = \phi(x_j)$ and $\tau\dot{x}_i = -x_i + h_i$, the effective recurrent input $h_i = \sum_j \nu_{ij}\phi_j$ decomposes by noise source into

$$h_i(t) = \underbrace{\sum_r u_i^{(r)} c_i^{(r)} \kappa_r(t)}_{T^{\mathrm{I}}\text{: rank-}R\text{ signal}} + \underbrace{\frac{1}{N}\sum_{r,j} u_i^{(r)} v_j^{(r)} \delta_{ij}^{(r)} \phi_j}_{T^{\mathrm{res}}\text{: contact residual}} + \underbrace{\frac{\sigma_w}{\sqrt{N}}\sum_{r,j} \chi_{ij}^{(r)}(c_i^{(r)} + \delta_{ij}^{(r)})\phi_j}_{T^{\mathrm{II}}\text{: random bulk}}, \qquad \kappa_r := \frac{1}{N}\sum_j v_j^{(r)}\phi_j, \tag{A.19}$$

where κ_r is the r-th presynaptic overlap or the effective rank-1 mode as defined by Mastrogiuseppe & Ostojic (2018). The N^{-1}-normalized structured residual T^{res} is a sum of N independent centred terms and vanishes.

In the term T^{II}, the $N^{-1/2}$-normalized bulk retains an $O(1)$ contribution. Following the standard construction for random-plus-structured recurrent networks (Mastrogiuseppe & Ostojic, 2018; Clark & Abbott, 2024), its contribution to a representative neuron can be absorbed into an effective Gaussian field, giving the single-site dynamics

$$\tau\dot{x}(t) = -x(t) + I(t) + \sum_r u^{(r)} c^{(r)}(t)\kappa_r(t) + \xi(t), \qquad \kappa_r(t) = \mathbb{E}\Big[v^{(r)}\phi(x(t))\Big], \tag{A.20}$$

with, conditional on the local gate,

$$\mathbb{E}\Big[\xi(t)\xi(s) \mid \{c^{(r)}\}\Big] = \sigma_w^2 C_\phi(t, s)\sum_r \Big[c^{(r)}(t)c^{(r)}(s) + D_\delta(t, s)\Big], \qquad C_\phi(t, s) = \mathbb{E}[\phi(x(t))\phi(x(s))]\,. \tag{A.21}$$

Because the gate multiplies both the structured and random parts of each contact, it rescales the background noise as well as the rank-R signal. Contact-specific fluctuations thus vanish from the structured term but remain in the bulk covariance through D_δ.

For a uniform gate ($k_A^{(r)} \propto u^{(r)}$), the gate factors out of population averages and the closure reduces to the R-overlaps κ_r and R-gate variables. This corresponds to the Hebbian and anti-Hebbian regime examined by Clark & Abbott (2024); Wakhloo et al. (2025); Gaudio et al. (2026). The heterogeneous case requires averaging over local gate trajectories, which we do not derive analytically.

Instead, Fig. 2d compares finite-N simulations with an exogenous (not activity-driven), heterogeneous rank-one gate against a large-sample DMFT reference, and finds the mean kernel error decays as $\approx N^{-1/2}$. This supports the closure but does not cover gates driven self-consistently by network activity.

Parameters: $N \in \{100, 200, 400, 800, 1600\}$, 6 trials per size, a 65,536-simulation reference; $p \in \{0.5, 1.5, 4.5\}$ ($\tau_\eta = 3$), giving $\mathrm{Var}(b)/\sigma_\eta^2 \in \{0.86, 0.67, 0.40\}$ and fitted slopes $\alpha = -0.57, -0.53, -0.54$.

A.4 Associative recall task

Rank-one writes. We use the single-mode $p\dot{K} = -K + F(t)$ in (A.14) and discretize it over a slot of duration Δ, with $F(t) = F_t$ held constant during the slot:

$$K_t = \lambda K_{t-1} + (1-\lambda)F_t, \qquad \lambda = e^{-\Delta/p} \in (0,1). \tag{A.22}$$

This provides the continuous-time reference for sequential rank-one writes (Schlag et al., 2021; Yang et al., 2024b). The Fig. 3 probe has no passive between-slot leak: its state changes only at a gated write, whose strength is set independently by β.

Associative-recall geometry. We use gated rank-one writes within the shared-address factorization

$$J = UDV^\top, \qquad A_t = K_t V^\top, \tag{A.23}$$

where $V \in \mathbb{R}^{d\times r}$ spans the common presynaptic/address space, $U \in \mathbb{R}^{d\times r}$ spans the slow postsynaptic/content space, and $K_t \in \mathbb{R}^{d\times r}$ is the fast content state.

We consider two constraints on the left factor (Schuessler et al., 2024): $K \parallel U$ and $K \perp U$. With orthonormal columns of U in the probe, let

$$P_U = UU^\top, \qquad P_{U^\perp} = I - UU^\top, \qquad w_t^{\parallel} = P_U w_t, \qquad w_t^{\perp} = P_{U^\perp} w_t. \tag{A.24}$$

Here $g_t \in \{0,1\}$ is the write gate, $a_t \in \mathbb{R}^r$ is a unit address coordinate, $w_t \in \mathbb{R}^d$ is the content target, and β is the learned write strength. The address and content target are learned projections of $x_t \in \{-1,0,+1\}^{2n_{\text{bits}}} \times \{0,1\}^2$; its flags specify the write and read events.

Vanilla associative rule. Vanilla fast weights use an additive outer-product write (Schmidhuber, 1992; Ba et al., 2016):

$$K_t^{\text{van},\parallel} = K_{t-1}^{\text{van},\parallel} + \beta g_t w_t^{\parallel} a_t^\top, \tag{A.25}$$

$$K_t^{\text{van},\perp} = K_{t-1}^{\text{van},\perp} + \beta g_t w_t^{\perp} a_t^\top. \tag{A.26}$$

Delta rule. The delta rule is the least-mean-square Widrow–Hoff update (Widrow & Hoff, 1960) in fast-weight form (Schlag et al., 2021; Yang et al., 2024b). Define its address-specific replacement operator

$$E_t := I - \beta g_t a_t a_t^\top. \tag{A.27}$$

The two geometry arms are

$$K_t^{\Delta,\parallel} = K_{t-1}^{\Delta,\parallel} E_t + \beta g_t \left(w_t^{\parallel} - \eta_w U D a_t\right) a_t^\top, \tag{A.28}$$

$$K_t^{\Delta,\perp} = K_{t-1}^{\Delta,\perp} E_t + \beta g_t w_t^{\perp} a_t^\top. \tag{A.29}$$

The switch $\eta_w \in \{0,1\}$ includes or excludes the slow prediction from the write residual. The trained delta model has $\eta_w = 1$; the frozen no-J delta intervention has $\eta_w = 0$ and also excludes J from the recurrent operator below.

In the $K \perp U$ arm, $P_{U^\perp} U D a_t = 0$, so changing η_w has no effect on the write. Delta nevertheless differs from vanilla because it corrects the prediction already stored in the fast state:

$$K_t^{\Delta,\perp} = K_{t-1}^{\Delta,\perp} + \beta g_t \left(w_t^\perp - K_{t-1}^{\Delta,\perp} a_t \right) a_t^\top . \tag{A.30}$$

Associative readout. Both write rules have the affine recurrence

$$K_t = K_{t-1} E_t + B_t, \tag{A.31}$$

where $E_t = I$ for vanilla and is given by (A.27) for delta. The term B_t is the additive write term in the appropriate geometry arm above. From $K_0 = 0$, iteration gives the exact operator solution

$$K_t = \sum_{j=1}^{t} B_j E_{j+1:t}, \qquad E_{j+1:t} := E_{j+1} \cdots E_t, \qquad E_{t+1:t} := I. \tag{A.32}$$

This product carries each association through subsequent writes.

We propagate (A.32) through the sequence-mixer recurrence of FastSlow (Appendix B, Fig.A4). The associative-recall probe uses the exact proximal solve with a trained, constant τ. Substituting the fast-state solution into the mixer operator gives

$$W_t := \left(\eta_r U D + \sum_{j=1}^{t} B_j E_{j+1:t} \right) V^\top, \qquad G_t := I - W_t, \qquad M_t := (I + \tau G_t^\top G_t)^{-1}. \tag{A.33}$$

Here $\eta_r = 1$ for the trained delta and vanilla models, while the no-J delta intervention sets $\eta_w = \eta_r = 0$. The geometry projector and all other learned parameters remain fixed in that intervention.

As in the write-before-read construction of Appendix B, the current write forms W_t before the hidden transition. In this probe, the tied, gated query is $c_t = \varepsilon_t V a_t$, where ε_t is the read gate. The mixer forcing and hidden-state update are

$$u_t = W_t c_t + g_t W_{\text{in}} \,\text{RMSNorm}(x_t), \qquad h_t = M_t h_{t-1} + \tau M_t G_t^\top u_t, \qquad h_0 = 0. \tag{A.34}$$

Expanding the mixer recurrence gives the decoded operator solution

$$y_t = W_o \,\text{RMSNorm}\left[\sum_{j=1}^{t} (M_t M_{t-1} \cdots M_{j+1}) \tau M_j G_j^\top u_j \right]. \tag{A.35}$$

The product preceding the $j = t$ term is I. Every M_j, G_j, and u_j in this expression is determined by the write sequence through (A.32)–(A.34). This is the solution for the exact associative-recall sequence mixer, not for the complete channel mixer of the full FastSlow block, which includes linear attention on heads.

Recall error and margin across writes. The code places the target write at array slot 0 and the query at array slot T, where T is the plotted write-to-read delay. With the recurrence above indexed from the pre-write state $K_0 = 0$, these are steps 1 and $T + 1$, respectively. Let $t_i \in \{0, 1\}$ be a target bit and $y_{T+1,i}$ its logit from (A.35). The two operator-theory curves in Fig. 3a are

$$\text{BitError}(T) = 100 \Pr_{\text{trials},i} \left[(y_{T+1,i} > 0) \neq (t_i = 1) \right], \tag{A.36}$$

$$\text{Margin}(T) = \left\langle (2t_i - 1) y_{T+1,i} \right\rangle_{\text{trials},i} . \tag{A.37}$$

Here $\Pr_{\text{trials},i}$ and $\langle \cdot \rangle_{\text{trials},i}$ denote the empirical proportion and mean over held-out trials and target bits.

Colored curves are from model traces. Dotted curves are exact operator theory rebuilt from (A.32) learned parameters and past through the recurrence and decoder. We also report bit accuracy for random-subspace control as trained geometry conditions.

Associative recall with summation. We extend the preceding task by querying two stored scalar values in $\{-1, +1\}$ and, with distractors and other writes in-between, predicting their sum at the second query – e.g., after writing `101` $\mapsto$ `+1` and `100` $\mapsto$ `-1`, querying `101` followed by `100` requires the output `0`. We train a new full-rank delta mixer of width 32 on this task, using the same write rule and proximal readout. At inference, we sample random inter-query delays and rotate J by 90° throughout writing and recurrent readout, including the proximal map M_t. Fig. 3d visualizes hidden-state trajectories using dPCA (Kobak et al., 2016), with projections held fixed across intact and rotated conditions. We report sum accuracy and separately probe whether the first stored value remains recoverable from fast memory at both queries.

Appendix B Shared-address FastSlow recurrent architecture

Shared-address fast-slow recurrent memory. Let $z_{1:T}$ be a token context. With a four-slot token interface, $e_t = \mathrm{concat}(E[z_{t,1}], \ldots, E[z_{t,4}])$; for a standard token interface, $e_t = E[z_t]$. The frontend is

$$\begin{aligned} x_t &= W_{\mathrm{proj}} e_t + b_{\mathrm{proj}}, & \bar{x}_t &= \mathrm{RMSNorm}(x_t), \\ k_t &= \mathrm{norm}[\mathrm{Conv}_k(W_k \bar{x}_t)], & v_t &= \mathrm{Conv}_v(W_v \bar{x}_t), \\ a_t &= \mathrm{norm}(P_k k_t), & \ell_t &= W_{\tau,2}\, \mathrm{SiLU}(W_{\tau,1} \bar{x}_t + b_{\tau,1}) + b_{\tau,2}, \\ \tau_t &= \mathrm{softplus}(\ell_t) > 0. \end{aligned}$$

Thus k_t is the key stream, $a_t \in \mathbb{R}^R$ is its address coordinate, and $v_t \in \mathbb{R}^{d_h}$ is the value/content target. The tied query address $a_{q,t} = a_t$ is the default shared-address choice.

Shared factorization. Let $V \in \mathbb{R}^{d_h \times R}$ be the shared right-hand address basis and let $U \in \mathbb{R}^{d_h \times R}$ be the slow content factor. Here $U = U_J D_J$ absorbs the diagonal slow spectrum. The slow and fast operators are

$$J = UV^\top, \qquad A_t = K_t V^\top, \qquad q_t = V a_t,$$

where $K_t \in \mathbb{R}^{d_h \times R}$ is the fast content state and $K_0 = 0$. Write $C = V^\top V$. The idealized orthonormal presentation has $C = I_R$, the implementation retains the actual finite-precision Gram matrix.

Predictive delta rule. Let s denote the fixed residual-prediction scale. The fast state writes only the part of the current content not predicted by its previous state and the scaled slow map:

$$\begin{aligned} \widehat{v_t} &= K_{t-1} a_t + sUCa_t, \\ d_t &= v_t - \widehat{v_t}, \\ K_t &= \gamma_t K_{t-1} + \beta_t d_t a_t^\top, \\ A_t &= K_t V^\top. \end{aligned}$$

The effective write strength is the existing bounded gate,

$$\beta_t = \beta_{\max} \tanh(\beta_{\mathrm{raw,t}} / \beta_{\max}),$$

with the write mask included in $\beta_{\mathrm{raw,t}}$. s appears only in the prediction error; it does not rescale the recurrent slow operator J.

Write-before-proximal-read recurrence. The current token write forms A_t, then $W_t = J + A_t$, before the hidden transition is constructed. The tied query injects into the shared address direction, $c_t = \varepsilon_t V a_{q,t}$. With

$$G_t = I - W_t, \qquad u_t = W_t c_t + W_{\mathrm{in}} \bar{x}_t, ,$$

the learned positive token-conditioned step defines the exact reference proximal map

$$M_t = (I + \tau_t G_t^\top G_t)^{-1}, \qquad b_t = \tau_t M_t G_t^\top u_t, \qquad h_t^{\mathrm{exact}} = M_t h_{t-1} + b_t.$$

Equivalently,

$$h_t^{\mathrm{exact}} = \arg\min_h \left\{ \tfrac{1}{2} \|h - h_{t-1}\|_2^2 + \tfrac{\tau_t}{2} \|(I - W_t)h - u_t\|_2^2 \right\}.$$

Since $G_t^\top G_t \succeq 0$, the exact proximal response satisfies $0 \prec M_t \preceq I$ for every $\tau_t > 0$.

Computing the exact response requires an $O(d_h^3)$ dense solve. In the implemented model, we approximate this response with two relaxed, diagonally preconditioned Jacobi iterates, using $O(d_h^2)$ work per token. This approximation does not, in general, equal the minimizer above, so the exact contraction bound is not asserted for the implemented transition.

This construction relates to recent work treating sequence-model recurrences as internal regression or linear-system problems, replacing exact dense solves with structured preconditioning or iterative linear algebra. Preconditioned DeltaNet derives curvature-aware updates from online least squares, using a diagonal approximation to retain chunkwise parallelism Tumma et al. (2026). MesaNet solves its recurrent test-time regression via a linear solving so that it admits chunkwise-parallel realization von Oswald et al. (2026). Local Linear Attention similarly poses an internal local regression problem with hardware-efficient blockwise primitives Zuo et al. (2026). Our construction differs in that the linear system regulates the vector-valued postsynaptic hidden dynamics induced by the fast-slow recurrent operator, while the matrix-valued fast memory K_t remains a separate plastic state.

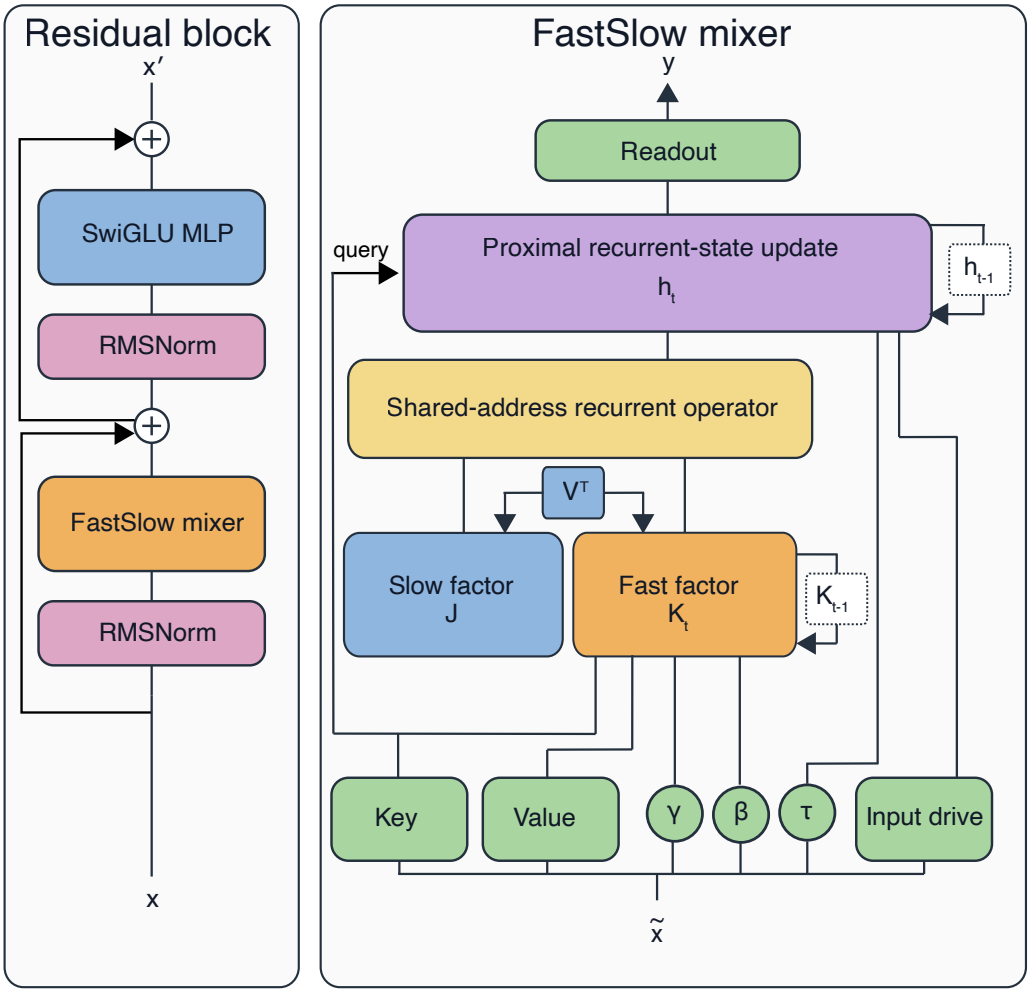


Figure A4: Schematic of the FastSlow mixer in sequential, showing the shared-address decomposition into a slow factor J and a token-updated fast factor K_t, whose combined output parameterizes the recurrent-state update h_t; the mixer sits inside the residual block (left) between RMSNorm and SwiGLU MLP layers.

Chunkwise form. For scalar decay γ_t, the Delta update is affine in K:

$$K_t = K_{t-1}P_t + \mathcal{B}_t, \qquad P_t = \gamma_t I_R - \beta_t a_t a_t^\top, \qquad \mathcal{B}_t = \beta_t[v_t - sUCa_t]a_t^\top.$$

For $\mathcal{F}_t(X) = XP_t + \mathcal{B}_t$, chronological composition is

$$(P_1, \mathcal{B}_1) \otimes_R (P_2, \mathcal{B}_2) = (P_1P_2,\ \mathcal{B}_1P_2 + \mathcal{B}_2).$$

This identity characterizes the causal fast states in a block. The implementation computes them with a causal WY solve for the delta updates, without explicitly forming or scanning the matrices P_t. The resulting K_t determines W_t and the tokenwise hidden transition.

Both the exact reference transition and its implemented Jacobi approximation are affine in the incoming hidden state. Their maps obey the chronological composition identity

$$(M_1, b_1) \otimes_L (M_2, b_2) = (M_2M_1,\ M_2b_1 + b_2),$$

where (M_t, b_t) denotes the chosen transition's affine map. The chunkwise implementation applies the approximate transition in factored form through a within-block hidden-state sweep, carrying (K, h) between blocks; it does not form or prefix-scan dense hidden maps. Padding leaves both states unchanged. The model introduces no direct A_tq_t output pathway.

Logits. The recurrent head output and existing output stack are

$$\begin{aligned}
y_t &= W_o\,\mathrm{ReadoutNorm}(h_t),\\
x_t^{\mathrm{mid}} &= x_t + W_{\mathrm{attn_out}}y_t,\\
n_t &= \mathrm{RMSNorm}_{\mathrm{ffn}}(x_t^{\mathrm{mid}}),\\
f_t &= W_{\mathrm{down}}\left[\mathrm{SiLU}(W_{\mathrm{gate}}n_t)\odot W_{\mathrm{up}}n_t\right],\\
x_t^{\mathrm{final}} &= x_t^{\mathrm{mid}} + f_t,\\
\ell_t &= \rho W_{\mathrm{vocab}}x_t^{\mathrm{final}} + b_{\mathrm{vocab}}, \qquad p(z_{t+1} \mid z_{\le t}) = \mathrm{softmax}(\ell_t).
\end{aligned}$$

Appendix C Implementation details

Architecture. FastSlow used embedding width 128 and two stacked blocks, each comprising a FastSlow sequence mixer and a SwiGLU channel mixer (Fig. A4); each sequence mixer used eight heads of dimension 16 and fast-weight rank 15. The model used RMSNorm, tied input/output embeddings, and no dropout. SwiGLU inner width was fixed at 160 before training and was not tuned by task. At vocabulary size 16, the non-compression model contained 448,084 trainable parameters, compared with 448,048 in the released four-layer multi-head GLA–SwiGLU reference. The two FastSlow blocks contained $2(8 \cdot 16 \cdot 15 + 128) = 4096$ persistent fast-weight and hidden-state scalars; finite input-history caches were accounted for separately. Compression used a final-state autoencoder: only the last encoder representation was supplied to a decoder with fixed sinusoidal positions, two width-preserving RMSNorm/linear/GELU layers, and the vocabulary readout. All sequence positions were reconstructed, without a direct same-position encoder-to-output connection.

FineWeb training uses $C = V^\top V$ in the slow prediction, while retaining $K_{t-1}a_t$ for the fast prediction. Its rankwise-decay variant replaces scalar γ_t by a diagonal right-side decay matrix. The benchmark uses the sequential scan; chunkwise evaluation is the affine regrouping above.

FastSlow combined the learned slow operator J with an online fast operator A_t. Hidden-state updates used the bounded Jacobi-2 approximation to the proximal system $Q_t = I + \tau_t G_t^\top G_t$, with $G_t = I - (J + A_t)$. Specifically, with $D_t = \mathrm{diag}(Q_t)$ and $r_t = h_{t-1} + \tau_t G_t^\top u_t$, the implementation computed $x_0 = D_t^{-1} r_t$ followed by $h_t = x_0 + \omega_t D_t^{-1}(r_t - Q_t x_0)$, where ω_t was the implemented reciprocal spectral upper bound. Thus, these experiments evaluated bounded Jacobi-2 FastSlow rather than the exact Cholesky solution.

MAD benchmark and published comparisons. We evaluated the FastSlow architecture using the released Mechanistic Architecture Design (MAD) benchmark of Poli et al. (2024) (Zenodo 10843663). We included all 66 released difficulty settings: 11 in-context recall, 14 noisy recall, 11 fuzzy recall, 6 memorization, 11 compression, and 13 selective-copying settings. Training-set sizes, sequence lengths, vocabulary sizes, and task-specific parameters followed each released setting. Every setting used its fixed 1,280-example held-out split.

For every difficulty setting, we trained the full grid of learning rates $\{10^{-4}, 5\times10^{-4}, 10^{-3}\}$ and weight decays $\{0, 0.1\}$, producing 396 runs. Each run used seed 12345, batch size 128, and 200 complete training epochs without early stopping. Optimization used AdamW with $(\beta_1, \beta_2) = (0.9, 0.98)$, $\epsilon = 10^{-8}$, gradient-norm clipping at 1, and an epoch-stepped cosine schedule toward 10^{-6}, without warmup. Model computation used bfloat16 with float32 master parameters and optimizer state. Evaluation occurred after every epoch.

For each difficulty setting, we selected the learning-rate/weight-decay pair with the highest final-epoch held-out accuracy, rather than selecting the best intermediate epoch. Task scores were the unweighted means of these selected accuracies across difficulty settings. The overall FastSlow score was the unweighted mean of the six task scores. Loss and accuracy excluded positions marked unsupervised by the released targets. All 396 runs completed with finite results. Runs used JAX on three RTX PRO 4000 Blackwell GPUs, one process per GPU; the campaign retained 42 sequential, 7 generic chunkwise, and 347 fused chunkwise runs, with execution backend recorded for each checkpoint.

Published reference results. Comparison rows for Mamba2, GLA, xLSTM, DeltaNet, Gated DeltaNet, Hawk, MesaNet, Hawk-MesaNet, and Transformer in Table 1 were transcribed from the MesaNet Table 9 (von Oswald et al., 2026); we did not rerun these models. These provide a contextual comparison rather than a controlled ranking. MesaNet's reported recipe differs from ours, including batch size 32, four learning rates, weight decays $\{0.01, 0.1\}$, and a warmup schedule. We retain separate FastSlow in-context and noisy-recall scores and identify FastSlow's six-task mean separately from the published table's aggregation (MesaNet, Appendix F.1 and Table 9).

Post-training J/Delta ablations. We selected the terminal checkpoint of the winning optimizer configuration for each of the 66 difficulty settings. Without retraining or hyperparameter reselection, we evaluated six conditions on the same 1,280 held-out examples: intact FastSlow; removal of J from the recurrent operator only; removal of J from the Delta-rule prediction only; removal

from both pathways; suppression of fast-weight writes; and replacement of Delta writes by gated Hebbian writes, removing prediction subtraction while retaining the existing write and decay gates. These interventions were applied to both sequence-mixer layers. The Hebbian condition therefore measured dependence on Delta correction in a Delta-trained model, rather than performance of a separately trained Hebbian architecture.

We recorded supervised-token accuracy, cross-entropy, and target-logit margin against the highest-scoring alternative class. Effects were expressed relative to the intact checkpoint and averaged equally across difficulty settings. Confidence intervals used 1,000 paired bootstrap resamples of examples within each setting, preserving the same sampled examples across intact and ablated conditions. These intervals describe evaluation-sample uncertainty conditional on the selected checkpoints, not variability across training seeds. Every accepted intact evaluation reproduced its saved accuracy exactly; the largest loss discrepancy was 1.23×10^{-7}. One initial compression mismatch was investigated and explicitly rerun under unchanged acceptance thresholds.

Layerwise and geometric diagnostics. On the first 128 held-out examples per setting, we scaled J to zero or one-half in each layer and measured operator norms, alignment, slow-subspace projection, prediction and residual energies, and hidden-state RMS. These post-selection diagnostics used the optimizer-selection split and show pathway dependence.

FineWeb-Edu pretraining. The language-model study used a separate two-block, 64,979,730-parameter FastSlow with width 512, four heads of width 128 per block, rank 64, learned plastic decay, soft β cap 0.01, and a chunkwise backend (chunk length 16). The slow prediction used $C = V^\top V$ while fast prediction retained $K_{t-1}a_t$ with decay γ_t as a right-side diagonal matrix. We trained from scratch on 1 billion tokens of FineWeb-Edu (Penedo et al., 2024), `sample-10BT` using `cl100k_base`, 4,096-token contexts, seed 42 and 32,768 valid tokens per update across three GPUs. AdamW used $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.1, gradient clipping at 1 and 10^{-4} z-loss. The peak learning rate was 0.005 (linear warmup to 10 M tokens, constant to 900 M, then linear decay toward 10^{-4} at 1 billion); memory and write-gate learning-rate multipliers were 0.1 and 0.5. Computation was bfloat16 with float32 master parameters and optimizer moments. Validation reused the first 1M tokens of the held-out cache in 4,096-token windows; the final 46.11 perplexity is on this monitored subset, not an untouched test set. The single checkpoint lineage included finite-spike recoveries and provides no training-seed uncertainty. Runs used JAX on three RTX PRO 4000 Blackwell GPUs for training.

AI use statement. Claude and ChatGPT were used for writing aid and literature assistance, code development, and help with mathematical derivations. All outputs were reviewed and verified by the authors, who take full responsibility for the content of the article.