FEEDBACK-GATED WEIGHT MODULATION ENABLES STATE-DEPENDENT COMPUTATION IN RECURRENT NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recurrent neural networks (RNNs) are useful models of sequential processing in cortical circuits but omit a defining feature of cortical organization: top-down feedback that dynamically regulates synaptic integration in lower order regions. In cortical pyramidal neurons, feedforward drive tends to arrive at proximal dendrites, whereas feedback carrying contextual and behaviorally relevant information often targets the distal apical dendrites. Nonlinear interactions between these streams can selectively reshape neuronal responses. To emulate this organization, we introduce a new RNN architecture, Gating on Weights with Feedback (GaWF), in which a standard RNN receives feedback gating signals computed from the subsequent layer's activations. These gates modulate individual weights on both the input and hidden weight matrices. This stands in contrast to gated RNN architectures, such as LSTMs and GRUs, which dynamically control information flow through gates computed on feedforward inputs and local hidden states that then act on unit-level representations. We evaluated GaWF on a new dynamic cluttered target-tracking task that requires simultaneous inference of target identity and location. Both the target and distractors could change dynamically. GaWF consistently outperformed conventional gated RNNs and other recurrent and state-space models, achieving higher held-out accuracy with less overfitting. Its gates produced sparse weight matrices that track the current target. Further mechanistic analyses revealed a functional division: input-weight gating predominantly tracked target location, whereas recurrent-weight gating represented target identity. Neither LSTMs nor GRUs segregated their gating signals based on the properties of the tracked target; instead, their gates remained largely undifferentiated, which may have contributed to their overfitting. These findings identify top-down connection-specific feedback gating as an effective and interpretable mechanism for context-dependent recurrent computation and highlight the functional importance of the interaction between hierarchical feedback in cortical circuits and dendritic computation.

1 INTRODUCTION

Neocortical processing depends on combining bottom-up sensory evidence about the outside world with internally generated top-down signals that carry expectations, predictions, behavioral relevance, and reinforcement-related information. Feedback pathways show strong specificity both in which neurons they innervate and in the dendritic compartments they target (Petreanu et al., 2009; Shen et al., 2022; Galloni et al., 2022). Specifically, long-range feedback often targets distal dendritic branches, where inputs are typically too remote from the cell body to drive spiking on their own but instead influence how feedforward and local inputs on proximal dendrites, capable of activating the neuron, are integrated (Behabadi et al., 2012; Larkum, 2013; Fişek et al., 2023). Such dendritic interactions are not merely a uniform gain change of neuronal output; they can instead alter stimulus tuning and selectivity (Lavzin et al., 2012; Smith et al., 2013). Consequently, top-down signals can regulate effective connectivity in lower-order circuits, supporting flexible, state-dependent computation.

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.

Contemporary applications of dendritic integration ideas to artificial neural networks have explored other directions. Dendritic credit assignment theories suggest that distal feedback carries teaching or error-related signals, whereas proximal compartments combine sensory input, enabling local learning without interpreting feedback as another feedforward input (Körding & König, 2001; Guerguiev et al., 2017; Lillicrap et al., 2020; Payeur et al., 2021). More broadly, the nonlinear computations and hierarchical integration supported by dendritic trees (Poirazi et al., 2003; Poirazi & Papoutsi, 2020; Beniaguev et al., 2021) have inspired artificial architectures that replace individual units with structured, dendrite-like subnetworks (Iyer et al., 2022; Chavlis & Poirazi, 2025). However, neither line of work has examined top-down inputs as gating signals that modulate local synaptic weights.

RNNs provide a standard abstraction of temporal integration and recurrent cortical dynamics (Mante et al., 2013; Barak, 2017). Yet, modern recurrent networks largely neglect top-down feedback and dendritic integration. LSTMs and GRUs compute gates from the current input and previous hidden state and apply them to unit-level states or outputs (Hochreiter & Schmidhuber, 1997; Cho et al., 2014); they therefore provide no explicit pathway through which a downstream representation can selectively modulate individual input or recurrent connections. Recent state-space models improve long-range sequence processing: S5 replaces banks of independent state-space models with a multi-input, multi-output state-space layer (Smith et al., 2023), whereas Mamba makes state-space parameters input-dependent to support content-selective propagation (Gu & Dao, 2024). In both architectures, top-down output feedback is still not implemented as connection-specific modulation. While some RNN architectures do support reconfiguration of local weights, such as fast weight programmers, they do so at test time using the feedforward input (Schmidhuber, 1992; Irie & Gershman, 2025).

This gap motivates a recurrent architecture in which the activation in a layer generates gates over its preceding layer's individual weights, providing a computational abstraction of feedback-dependent, dendrite-local modulation. To address this, we introduce Gating on Weights with Feedback (GaWF), a recurrent architecture that maps a higher layer's output at the preceding time step onto connection-specific gates in the lower layer. Applied to an RNN, this means using the activation from a subsequent layer to compute gating signals applied to each of the RNN's input and recurrent weights. Given the hypothesized function of such a mechanism in the brain for shaping stimulus processing in a context- and history-dependent manner, we devised a task that captures this. This task, the Clutter-Moving MNIST (CM-MNIST) task, requires continuous tracking of the location and identity of a digit moving in a straight line against other digits moving randomly. Both the target and distractor digits can change at random. This requires a model to maintain and update both the target's location and identity in the face of unsignaled changes. We compare GaWF with parameter-matched recurrent and state-space baselines on CM-MNIST and analyze its learned gates to characterize how feedback reshapes the effective input and recurrent connectivity of the lower layer.

2 GaWF architecture and the CM-MNIST task

2.1 Gating on weights with feedback

GaWF instantiates connection-specific feedback by extending an Elman-style recurrence model (Elman, 1990) with a top-down gating pathway. At each time step, the current input and previous hidden state are mapped through the input and recurrent weight matrices. GaWF applies an elementwise gate to each individual weight, where the gate matrix is an outer product expansion from the output of the subsequent layer at the previous time step and is learned jointly with the base RNN weights (Figure 1A). The GaWF mechanism can be implemented with any multilayer artificial neural network architecture, and we focus here on its application to a single-layer RNN.

Consider a GaWF-RNN cell with current input $\mathbf{x}_t \in \mathbb{R}^m$, previous hidden state $\mathbf{h}_{t-1} \in \mathbb{R}^n$, and feedback $\mathbf{f}_t \in \mathbb{R}^d$ which is the output of a subsequent layer. We concatenate the input and recurrent state and likewise combine their corresponding weight matrices:

$$\mathbf{a}_t = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{bmatrix} \in \mathbb{R}^{m+n}, \qquad \mathbf{W} = \begin{bmatrix} \mathbf{W}^{\text{in}} & \mathbf{W}^{\text{rec}} \end{bmatrix} \in \mathbb{R}^{n \times (m+n)}. \tag{1}$$

Here, $\mathbf{W}^{\text{in}} \in \mathbb{R}^{n \times m}$ contains the input weights and $\mathbf{W}^{\text{rec}} \in \mathbb{R}^{n \times n}$ contains the recurrent ones. The feedback vector defines a diagonal matrix $\mathbf{\Sigma}_t = \text{Diag}(\mathbf{f}_t)$. Two learnable matrices, $\mathbf{U} \in \mathbb{R}^{n \times d}$ and

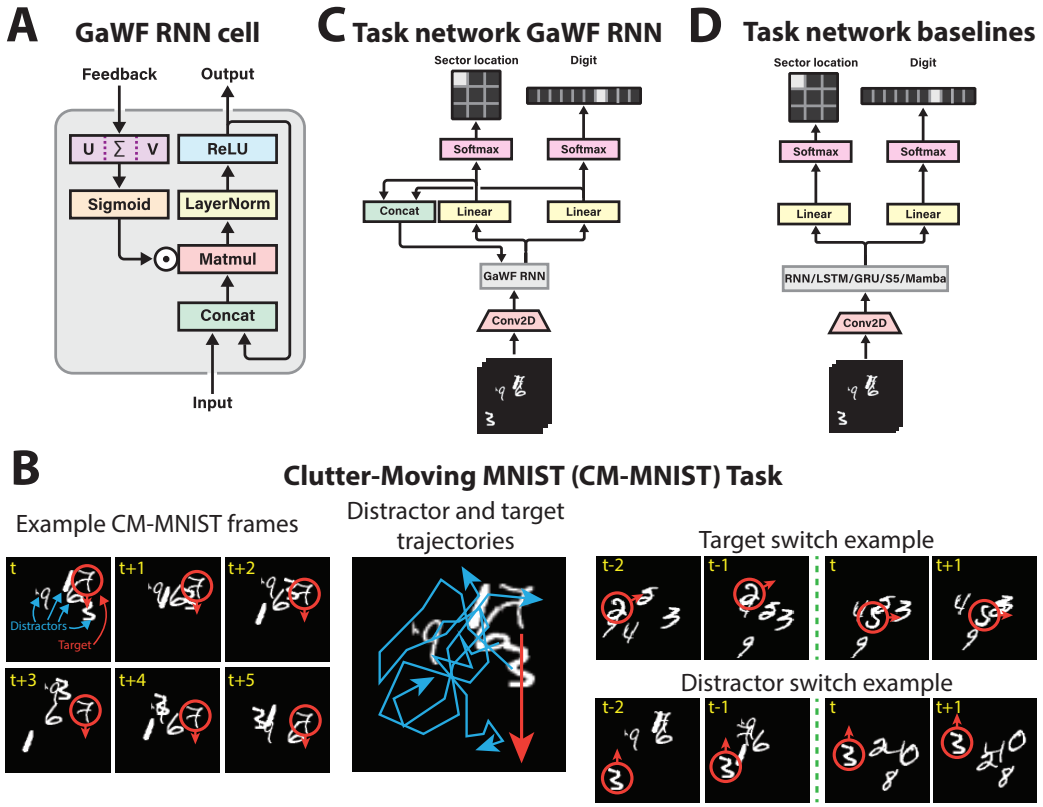


Figure 1: **GaWF architecture, shared model-comparison pipeline, and the CM-MNIST task.** **A**: in the GaWF cell, feedback from the adjacent higher-layer activation forms the connection-level gate $\mathbf{G}_t$ through $\mathbf{\Sigma}_t$, $\mathbf{U}$, and $\mathbf{V}$; element-wise multiplication with the static weight matrix determines the effective input and recurrent weights. **B**: *Left*: Example CM-MNIST frames showing that the target follows a straight trajectory while distractors change their directions on every frame. *Middle*: Schematic comparing target and distractor trajectories. *Right*: Example frames surrounding switches in target or distractors. **C**: a shared Conv2D encoder maps the current and preceding frames to $\mathbf{x}_t$, and separate linear readout heads decode the hidden state into digit-identity and sector-location logits, which are concatenated and returned as detached feedback at the next time step. **D**: the parameter-matched pipeline retains the encoder and readout heads while replacing GaWF with an Elman RNN, LSTM, GRU, S5, or Mamba, at approximately 586K parameters.

$\mathbf{V} \in \mathbb{R}^{d\times(m+n)}$, expand this feedback into a gate matrix with the same dimensions as the combined weight matrix:

$$\mathbf{G}_t = \sigma\left(\frac{\mathbf{U}\mathbf{\Sigma}_t\mathbf{V}}{\tau}\right) = \sigma\left(\frac{1}{\tau}\sum_{k=1}^{d} f_{t,k}\,\mathbf{u}_k\mathbf{v}_k^\top\right) \in (0,1)^{n\times(m+n)}, \tag{2}$$

where τ is the gate temperature, $\mathbf{u}_k$ is the k-th column of $\mathbf{U}$, and $\mathbf{v}_k^\top$ is the k-th row of $\mathbf{V}$. Each feedback component therefore scales one learned rank-one contribution to the pre-sigmoid gate.

The feedback-dependent effective weight matrix is obtained through a Hadamard product and, together with a bias $\mathbf{b} \in \mathbb{R}^n$, determines the hidden update:

$$\widetilde{\mathbf{W}}_t = \mathbf{G}_t \odot \mathbf{W}, \qquad \mathbf{h}_t = \phi\left(\widetilde{\mathbf{W}}_t\mathbf{a}_t + \mathbf{b}\right), \tag{3}$$

while the state nonlinearity

$$\phi(\mathbf{z}) = \mathrm{ReLU}\left(\mathrm{LN}\left(\mathbf{z}\right)\right) \tag{4}$$

applies layer normalization followed by rectification, producing a non-negative state with a population-relative threshold (Section A.1). The resulting $\mathbf{h}_t$ is both the recurrent state at the next

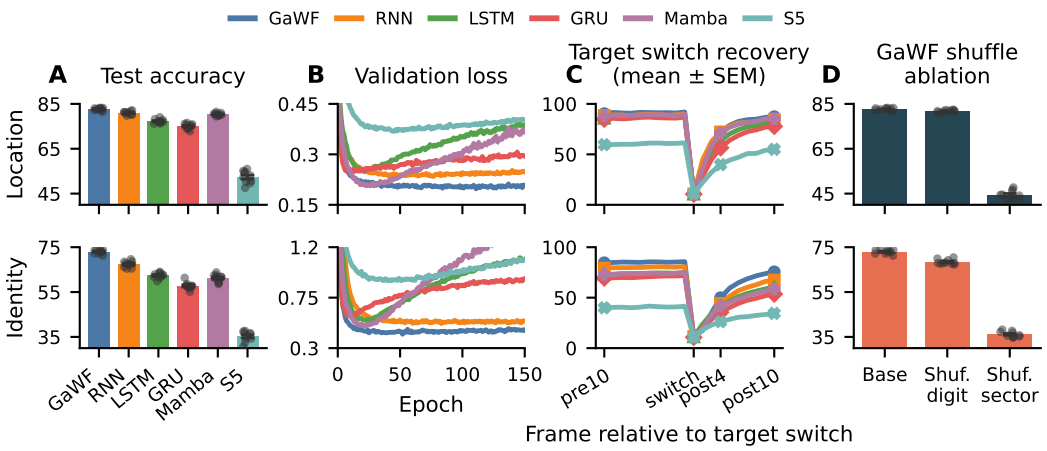


Figure 2: **GaWF outperforms parameter-matched baselines on both readouts.** Rows: sector-location (top) and digit-identity (bottom) readouts for six parameter-matched architectures. **A**: test accuracy, dots denoting individual seeds. **B**: validation-loss trajectories. **C**: accuracy aligned to target switches. **D**: GaWF accuracy under inference-time shuffling of the sector or the digit feedback segment.

step and the input to the readout heads. Partitioning the gate as $\mathbf{G}_t = [\mathbf{G}_t^{\text{in}}\ \mathbf{G}_t^{\text{rec}}]$ gives separate, time-varying modulation matrices for the input and recurrent pathways. We write g for an individual entry of $\mathbf{G}_t$ and w for the entry of $\mathbf{W}$ that it scales, which gives each connection the effective weight $\tilde{w} = gw$; values of g near 0 suppress the corresponding connections, whereas values near 1 preserve nearly their full learned strength. In this way, GaWF transforms one feedback vector into connection-specific modulations, so that inputs converging on the same unit can be gated differently.

2.2 Clutter-Moving MNIST

To evaluate RNNs under demands for continuous visual tracking and rapid recurrent-state updating, we devised the CM-MNIST task, a cluttered, target-switching extension of Moving MNIST (Srivastava et al., 2015) built from MNIST digits (LeCun et al., 1998). CM-MNIST presents a continuous sequence of grayscale 96×96 video frames, each containing one target digit and one or more moving distractors (Figure 1B). For each frame the model must report the target identity, as one of 10 digit conditions, and the target location, as one of 9 sector conditions on a 3×3 spatial grid. In each input sequence, the target follows a straight trajectory, changing direction only when it rebounds from a frame edge, and up to 12 distractor digits change direction on every frame. At random either the target or the set of distractor digits is changed. A target switch resamples the target identity, position, and speed, whereas a distractor switch replaces all distractors with a new set. The two types of switches occur without an explicit cue and can be detected only from the visual input. A target switch requires the model to replace its stored target information, identify the new target, and resume tracking its identity and location, combining temporal maintenance with rapid updating.

2.3 Model architecture and training

We adapted the generic GaWF formulation to CM-MNIST using a single GaWF recurrent layer, with feedback constructed from the two task readouts (Figure 1C). At each time step t, a shared Conv2D encoder maps two sequential input frames to $\mathbf{x}_t \in \mathbb{R}^m$, which serves as the feedforward input to the GaWF hidden layer. GaWF combines $\mathbf{x}_t$ with the previous hidden state to produce $\mathbf{h}_t \in \mathbb{R}^n$. Two linear readout heads produce 10 digit-identity logits, $\mathbf{y}_t^{\text{id}} \in \mathbb{R}^{10}$, and 9 sector-location logits, $\mathbf{y}_t^{\text{loc}} \in \mathbb{R}^9$.

The logits from the preceding time step (initialized to $\mathbf{0}$ at the start of each sequence) are concatenated to form the feedback vector, which is detached so that the readout heads learn from the task

loss, unaffected by their role as the feedback source:

$$\mathbf{f}_t = \left[\mathbf{y}^{\mathrm{id}}_{t-1};\, \mathbf{y}^{\mathrm{loc}}_{t-1}\right] \in \mathbb{R}^d, \qquad d = 19. \tag{5}$$

Digit identity and sector location are predicted through separate readout heads, and the feedback contains no additional component explicitly representing a particular digit–sector combination.

All models compared on CM-MNIST share the same Conv2D encoder and linear digit-identity and sector-location readout heads, differing only in the intervening sequence-processing module: GaWF, an Elman RNN (Elman, 1990), LSTM (Hochreiter & Schmidhuber, 1997), GRU (Cho et al., 2014), S5 (Smith et al., 2023), or Mamba (Gu & Dao, 2024) (Figure 1D). At 24 frames per second, we used a 40-h training stream and separate 40-min validation and test streams generated from disjoint MNIST image pools; the test set was excluded from model and hyperparameter selection. We matched the architectures at approximately 586K parameters and selected model-specific hyperparameters separately for each architecture by target-identity validation accuracy following grid search. All models were optimized with AdamW (Loshchilov & Hutter, 2019) for 150 epochs across 10 independent seeds.

3 Results

3.1 GaWF improves continuous target tracking using feedback

Performance against matched baselines GaWF exceeded all five parameter-matched recurrent and state-space baselines on both readouts, with a larger margin for target identity than for target location (Figure 2A). We evaluated the performance of each model on held-out test data using the checkpoint with the highest validation target-identity accuracy (Section B.2). GaWF had the highest accuracy on both target identity (73.0%) and location (82.6%), against 67.4% and 81.0% for the next best architecture, the RNN, and 35.2%–62.4% and 52.4%–80.3% for the remaining four (Figure 2A). On the identity readout, GaWF also remained the most accurate when every model was trained on a shorter movie and when all six architectures shared a hidden width of 128, a setting in which GaWF had fewer parameters than every baseline except the RNN (Section D.2). Supplying the recurrent baselines with the same feedback as an additional input did not close this gap on either readout, indicating that feedback availability alone does not account for the advantage of GaWF (Section D.4). Moreover, its validation loss curve remained low once converged, while those of several other models subsequently increased, indicating potential overfitting to the training data; after target switches, GaWF also showed the steepest post-switch recovery (Figure 2B, C).

Asymmetry between the two feedback segments Identifying the target requires locating it among the distractors, whereas locating it does not require its identity, and every architecture accordingly predicted sector location more accurately than digit identity. To test whether the two feedback segments reflected this asymmetry, we froze a trained GaWF network and shuffled either its sector or its digit segment on every frame. Sector accuracy was largely preserved under the digit shuffle (0.8 percentage points) and substantially reduced under the sector shuffle (38.1 points; Figure 2D). Digit accuracy declined modestly under the digit shuffle (4.5 points) and substantially under the sector shuffle (36.8 points). Performance on both readouts therefore depends on the gating signal, and most strongly on its sector segment.

3.2 GaWF implements quasi-binary sparse routing

Both the input- and recurrent-gate value distributions were U-shaped, with values concentrated near 0 and 1. We pooled the g^{in} and g^{rec} values across all test frames into one histogram per pathway (Section E). g^{in} values were strongly skewed toward the closed state ($g^{\mathrm{in}} < 0.1$, 67.6%; $g^{\mathrm{in}} > 0.9$, 20.7%; Figure 3A), whereas g^{rec} values were skewed toward the open state ($g^{\mathrm{rec}} < 0.1$, 18.9%; $g^{\mathrm{rec}} > 0.9$, 66.1%; Figure 3B). Gating shifted the effective weights $\tilde{w}$ toward zero in both pathways (Figure 3C, D). The gates thereby drive part of the connectivity toward zero while leaving the rest near full strength, a quasi-binary form of sparse routing.

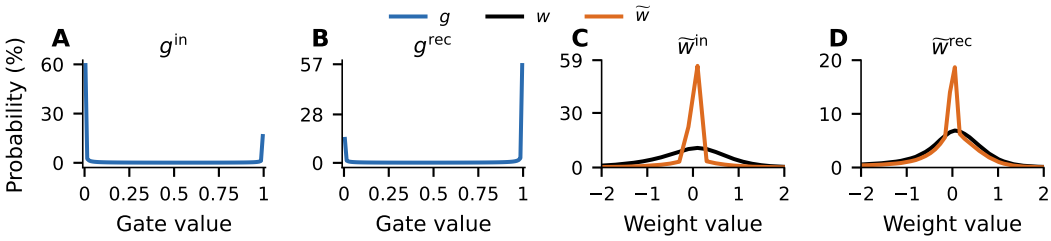


Figure 3: **U-shaped gates sparsify effective connectivity. A–D**: distributions of input-gate values, recurrent-gate values, input weights, and recurrent weights. In **C** and **D** the learned weights w are shown in black and the effective weights $\tilde{w}$ in orange.

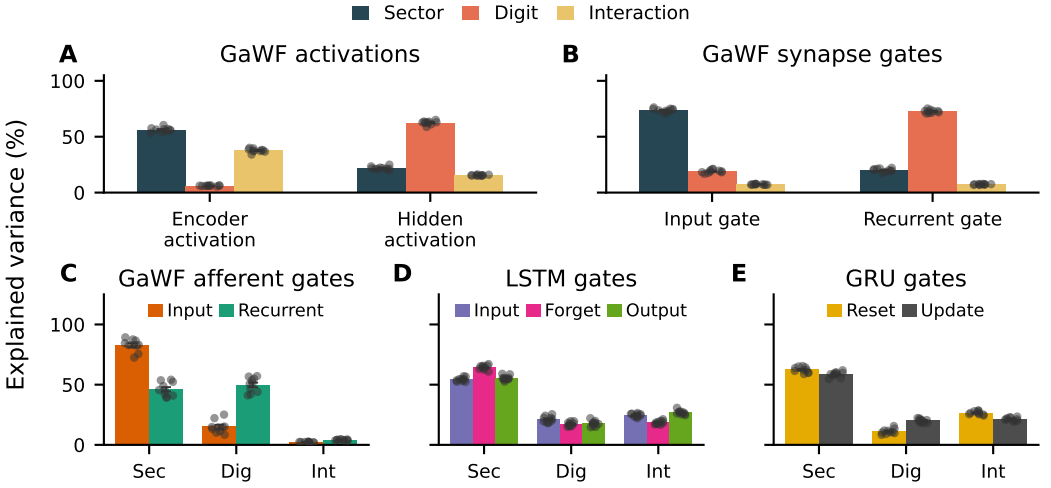


Figure 4: **The two gates carry different task variables, whereas conventional gates are less specialized. A**, **B**: balanced condition-mean variance decomposition of the encoder and hidden activations of the pathways they modulate (**A**) and of GaWF synapse-level gates (**B**). **C–E**: the same decomposition after projecting the connection-level gates onto destination units (**C**), compared with LSTM (**D**) and GRU (**E**) unit gates.

3.3 GaWF learns a complementary specialization of its two gates

To characterize which connections the quasi-binary gates select, we compared each gate with the source representation it acts on, since the input gate modulates encoder-to-hidden connections and the recurrent gate hidden-to-hidden connections. Specifically, we decomposed the variance of encoder and hidden states, and of input- and recurrent-gate values, into components associated with the target's sector location, digit identity, and their interaction.

Task-variable composition of encoder and hidden activity Encoder activations were dominated by target location and hidden activations by target identity. Encoder activations contained little identity-related variance but a large sector-by-digit interaction (location, 56.2%; identity, 6.1%; interaction, 37.7%; Figure 4A). Hidden activations showed a smaller interaction component and were dominated by target identity, with location contributing a smaller share (identity, 62.6%; location, 21.9%; interaction, 15.5%).

Task-variable composition of the two gates The synapse-level feedback gates aligned with dominant pathway-specific representations and were more strongly demixed. Target location accounted for most of the input-gate variance and target identity for most of the recurrent-gate variance (input gate: location, 73.7%; identity, 19.0%; recurrent gate: identity, 72.6%; location, 20.1%; Figure 4B).

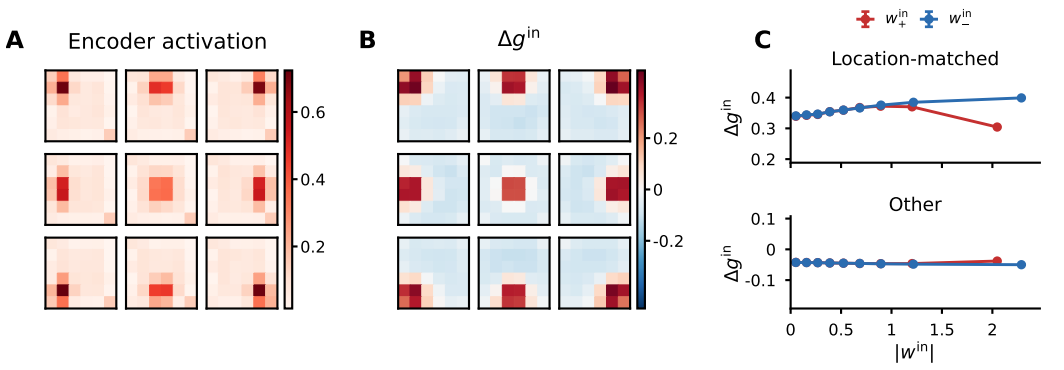


Figure 5: **Input gating is topographically aligned with target location.** **A**: channel-averaged encoder activation, with the nine maps arranged by the 3×3 target-sector layout. **B**: marginal Δg^{in} for each target sector, averaged over hidden units and channels and mapped onto the encoder spatial grid. **C**: marginal Δg^{in} versus $|w^{\text{in}}|$ for w_+^{in} and w_-^{in}, separated by location-matched (top) and other (bottom) source positions.

Both gates contained relatively small interaction components (input gate: 7.3%; recurrent gate: 7.3%). Together, the two gating pathways carry complementary task-variable specializations.

Unit-level comparison across architectures Conventional gated architectures showed less pronounced task-variable specialization. We projected the GaWF gates onto destination units so that three architectures were compared at the unit level. The projected GaWF gates retained complementary task-variable dominance, though weaker than at the synapse level (Figure 4C). Conventional gates showed limited or absent complementary specialization: location dominated all three LSTM gates as well as both GRU gates (Figure 4D, E). GaWF appears to separate task-variable control more sharply between its input and recurrent pathways than conventional gated RNNs.

3.4 Input gating selectively opens connections aligned with the target location

If the location organization of the input gate takes a spatial form, connections from encoder positions aligned with the target's current location should receive larger gate values, as biased-competition accounts predict for attention toward behaviorally relevant representations (Desimone & Duncan, 1995; Reynolds et al., 1999; Brefczynski & DeYoe, 1999). Testing this prediction requires understanding the encoder's own spatial layout.

Spatial layout of the encoder representation We first conditioned encoder activation on the target's sector and mapped it onto the encoder grid. Target location was represented topographically: across the 9 sector-location conditions the peak of encoder activity shifted systematically with the target, tracking the positions whose receptive fields overlapped its sector (Figure 5A), whereas conditioning on digit identity produced no corresponding displacement (Figure 11A).

Topographic organization of input-gate modulation To reduce the influence of connection-specific differences in overall gate level, we quantified location-dependent modulation as the deviation of each gate value from that connection's grand mean. For each connection (i, j), we defined

$$\Delta g_{ij,t}^{\text{in}} = g_{ij,t}^{\text{in}} - \overline{g}_{ij}^{\text{in}}, \tag{6}$$

where $\overline{g}_{ij}^{\text{in}}$ is the grand mean of g_{ij}^{in} over the balanced frames (Section F). For each sector location, averaging $\Delta g_{ij,t}^{\text{in}}$ over that sector's frames gives one marginal $\Delta g_{ij}^{\text{in}}$ per connection and sector, measuring how far the connection's gate departs from $\overline{g}_{ij}^{\text{in}}$ while the target occupies that sector; further averaging these values across hidden units and feature channels gives one spatial map per sector, in which positive and negative values mark connections whose gate values under the current target

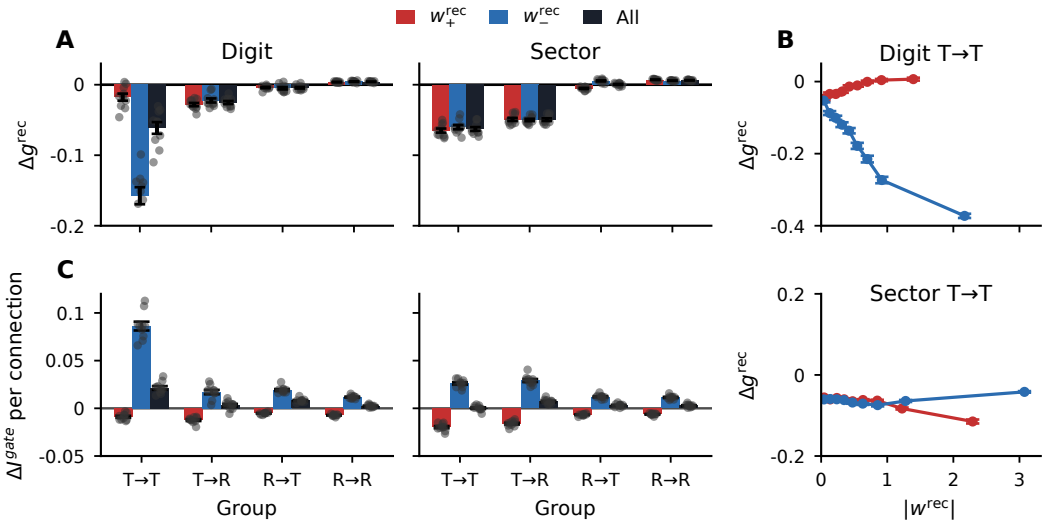


Figure 6: **Recurrent gating asymmetrically closes digit-tuned connections.** **A**: marginal Δg^{rec} under digit (left) and sector (right) conditions, grouped by source-to-destination tuning and by weight sign; T denotes units tuned to the current digit or sector and R the remainder. **B**: marginal Δg^{rec} versus $|w^{\mathrm{rec}}|$ for $T \to T$ under digit (top) and sector (bottom) conditions. **C**: connection-normalized ΔI^{gate} under digit (left) and sector (right) conditions. Red, blue, and black denote w_+^{rec}, w_-^{rec}, and all connections.

location lie above or below $\overline{g}_{ij}^{\mathrm{in}}$, respectively. The marginal Δg^{in} maps showed a compact region of positive modulation that shifted with the target sector and aligned with the encoder-activation peak (Figure 5B). Accordingly, input gating appears to follow the encoder's topographic representation.

Magnitude dependence among location-matched input connections Connections originating from encoder positions assigned to the current target sector received similarly strong increases in modulation regardless of weight sign ($\langle \Delta g \rangle_{w_+} = 0.353$; $\langle \Delta g \rangle_{w_-} = 0.366$). Within this group, however, connection magnitude affected the modulation of positive- and negative-weight connections differently: stronger positive-weight connections were slightly reduced, whereas stronger negative-weight connections were modestly enhanced ($\beta_{\Delta g \sim |w_+|} = -0.008$; $\beta_{\Delta g \sim |w_-|} = 0.017$; Figure 5C, top). At off-target locations on the encoder grid, connections were weakly closed at both weight signs, with little dependence on magnitude (Figure 5C, bottom). The input pathway therefore implements a spatial-selection rule: it preferentially opens connections whose encoder sources are aligned with the current target location.

3.5 Recurrent gating asymmetrically closes digit-tuned connections

Group dependence of recurrent-gate modulation Under digit-identity conditions, recurrent-gate modulation was concentrated on connections originating from digit-tuned units and was most negative within the tuned ensemble (digit conditions: $\langle \Delta g \rangle_{T \to T} = -0.061$; $\langle \Delta g \rangle_{T \to R} = -0.025$; $\langle \Delta g \rangle_{R \to T} = -0.004$; $\langle \Delta g \rangle_{R \to R} = 0.004$; Figure 6A, left). Repeating the analysis with sector-tuned ensembles revealed a smaller difference between groups, with $T \to T$ and $T \to R$ modulated to a comparable degree (sector conditions: $\langle \Delta g \rangle_{T \to T} = -0.063$; $\langle \Delta g \rangle_{T \to R} = -0.050$; Figure 6A, right). Recurrent gating thereby closed connections from units tuned to the current condition under both task variables, concentrating the closure within the tuned ensemble for target identity.

Sign dependence within digit-tuned ensembles Within the $T \to T$ group under digit-identity conditions, negative-weight connections received large negative modulation, whereas positive-weight ones closed weakly ($\langle \Delta g \rangle_{w_+} = -0.018$; $\langle \Delta g \rangle_{w_-} = -0.157$; Figure 6A, left). Their dependence on connection magnitude echoed the pattern at the input gate: positive-weight connections were modulated less at larger magnitudes, whereas negative-weight connections were suppressed

more strongly ($\beta_{\Delta g \sim |w_+|} = 0.037$; $\beta_{\Delta g \sim |w_-|} = -0.147$; Figure 6B, top). Gating thus reduced the effective strength of within-ensemble recurrent connections at both weight signs, but with a much larger reduction at negative weights.

Net change in recurrent current Gate modulation alone does not determine the change in recurrent drive, which also depends on connection strength and the non-negative source activity $h_{j,t-1}$. For each recurrent connection (i, j), the gate-dependent current at frame t was defined as

$$\Delta I^{\text{gate}}_{ij,t} = \Delta g^{\text{rec}}_{ij,t}\, w^{\text{rec}}_{ij}\, h_{j,t-1}, \tag{7}$$

where $\Delta g^{\text{rec}}_{ij,t}$ is defined as in Equation (6) for the recurrent gate. Because $h_{j,t-1} \geq 0$, the sign of each term is set by $\Delta g^{\text{rec}}_{ij,t}$ and w^{rec}_{ij}. The quantity is instantaneous and pre-activation, and the hidden update applies layer normalization across units, so it does not by itself establish a change in the activity of the tuned units. For each digit or sector condition, we averaged $\Delta I^{\text{gate}}_{ij,t}$ over the frames of that condition and then over the nonzero connections of a group under that condition, taken for one weight sign or for both. Among digit-conditioned $T \to T$ connections, ΔI^{gate} was near zero for positive-weight connections and strongly positive for negative-weight connections, giving a net excitatory change across all connections (-0.009, 0.086, and 0.021, respectively; Figure 6C, left). Under sector-location conditions the all-connections change stayed near zero (Figure 6C, right). The net excitatory change under digit-identity conditions therefore reflects the attenuation of negative-weight input, with positive-weight input nearly unchanged.

4 Discussion

Here we introduce the GaWF-RNN architecture, which transforms activations in the subsequent upper layer into connection-specific gates over input and recurrent weights in a lower layer. GaWF-RNN outperformed parameter-matched recurrent and state-space models in continuous target tracking, most clearly in maintaining target identity. The GaWF component was crucial to this enhanced performance: shuffling its sector segment at inference lowered accuracy on both the target's sector and identity, whereas shuffling its digit segment lowered both only slightly. The gating signal was quasi-binary, mostly closed on input and open on recurrent connections, and sparsified effective connectivity in both pathways. Interestingly, the GaWF-RNN gate signals aligned with the connections specialized for encoding different task features (i.e., target sector and identity), whereas other gated RNNs like the LSTM and GRU seemingly did not. This alignment may facilitate its generalization.

Earlier RNN variants share some elements with our approach, though none matches it exactly. For instance, multiplicative RNNs form the recurrent weight matrix by multiplying two factor matrices with an input-conditioned diagonal matrix, which resembles our construction of the feedback gating matrix (Krause et al., 2016). Likewise, our modulated RNN weight matrices are akin to the fast weights used in fast weight programmer networks; however, those methods update weights using the input and hidden state, whereas we use a feedback signal to modulate a fixed set of weights (Irie & Gershman, 2025). Using feedback as an attention-like signal has also been studied, but mainly to choose specific modules or to modulate their hidden states (Mittal et al., 2020).

The architecture introduced here brings both engineering benefits over other gated RNNs and drawbacks that still need refinement. One major benefit is that the GaWF-RNN's parameter count grows more slowly with the input and hidden-state dimensions. For the input layer, assuming feedback from a subsequent layer of the same width, only $2nm + 3n^2 + 2n$ parameters are needed—less than for an LSTM ($4nm + 4n^2 + 8n$) or a GRU ($3nm + 3n^2 + 6n$). A notable drawback is that the weight-gating matrix must be recomputed at every time step and is not readily parallelizable. Possible mitigations include precomputing the rank-1 matrices tied to each feedback activation, caching the gating matrices, or projecting the feedback vector into a lower-dimensional space. In sum, GaWF-RNN lets a higher layer gate the individual input and recurrent connections of a lower layer, yielding better continuous target tracking and more favorable parameter scaling than conventional gated RNNs.

AI use statement

In this work, we used generative AI tools to provide feedback on the design of analyses, to assist in implementing analysis code, and to assist in interpreting results. We have not used generative

AI tools to generate synthetic data sets, to develop theoretical models or conceptual frameworks, to formulate mathematical claims or assist in the writing of proofs, to propose or refine hypotheses, or to clean or reformat data sets; support for qualitative and thematic data analysis is not applicable to this work. Additionally, we used generative AI tools to draft and edit parts of the manuscript for clarity, to suggest a structure for the manuscript, to find and check related literature, and to search for information about submission policies. We have reviewed all AI-assisted work: all reported numbers were produced by analysis code that the authors reviewed and ran on our own checkpoints and were verified against our analysis records, all AI-assisted text was checked line by line against those records by the authors, and all AI-suggested analyses were run and checked by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

Section A.1 specifies the GaWF recurrence in full, including the state nonlinearity, the treatment of dropout at training and evaluation time, and the definition of hidden activity used in all analyses; Section A.2 gives the corresponding configuration of the baseline modules. An anonymized code package containing the CM-MNIST generators, shared recurrent cores, training and evaluation code, and environment setup instructions is provided as supplementary material.

References

Omri Barak. Recurrent neural networks as versatile tools of neuroscience research. *Current Opinion in Neurobiology*, 46:1–6, 2017. doi: 10.1016/j.conb.2017.06.003.

Bardia F. Behabadi, Alon Polsky, Monika Jadi, Jackie Schiller, and Bartlett W. Mel. Location-dependent excitatory synaptic interactions in pyramidal neuron dendrites. *PLoS Computational Biology*, 8(7):e1002599, 2012. doi: 10.1371/journal.pcbi.1002599.

David Beniaguev, Idan Segev, and Michael London. Single cortical neurons as deep artificial neural networks. *Neuron*, 109(17):2727–2739.e3, 2021. doi: 10.1016/j.neuron.2021.07.002.

Julie A. Brefczynski and Edgar A. DeYoe. A physiological correlate of the "spotlight" of visual attention. *Nature Neuroscience*, 2(4):370–374, 1999. doi: 10.1038/7280.

Spyridon Chavlis and Panayiota Poirazi. Dendrites endow artificial neural networks with accurate, robust and parameter-efficient learning. *Nature Communications*, 16(1):943, 2025. doi: 10.1038/s41467-025-56297-9.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179.

Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual Review of Neuroscience*, 18:193–222, 1995. doi: 10.1146/annurev.ne.18.030195.001205.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. doi: 10.1207/s15516709cog1402_1.

Mehmet Fişek, Dustin Herrmann, Alexander Egea-Weiss, Matilda Cloves, Lisa Bauer, Tai-Ying Lee, Lloyd E. Russell, and Michael Häusser. Cortico-cortical feedback engages active dendrites in visual cortex. *Nature*, 617(7962):769–776, 2023. doi: 10.1038/s41586-023-06007-6.

Alessandro R. Galloni, Zhiwen Ye, and Ede Rancz. Dendritic domain-specific sampling of long-range axons shapes feedforward and feedback connectivity of L5 neurons. *Journal of Neuroscience*, 42(16):3394–3405, 2022. doi: 10.1523/JNEUROSCI.1620-21.2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. doi: 10.48550/arXiv.2312.00752. URL https://openreview.net/forum?id=tEYskw1VY2.

Jordan Guerguiev, Timothy P. Lillicrap, and Blake A. Richards. Towards deep learning with segregated dendrites. *eLife*, 6:e22901, 2017. doi: 10.7554/eLife.22901.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Kazuki Irie and Samuel J. Gershman. Fast weight programming and linear transformers: From machine learning to neurobiology. *Transactions on Machine Learning Research*, 2025. doi: 10.48550/arXiv.2508.08435. URL https://openreview.net/forum?id=TDG8EkNmQR.

Abhiram Iyer, Karan Grewal, Akash Velu, Lucas Oliveira Souza, Jeremy Forest, and Subutai Ahmad. Avoiding catastrophe: Active dendrites enable multi-task learning in dynamic environments. *Frontiers in Neurorobotics*, 16:846219, 2022. doi: 10.3389/fnbot.2022.846219.

Konrad P. Körding and Peter König. Supervised and unsupervised learning with two sites of synaptic integration. *Journal of Computational Neuroscience*, 11(3):207–215, 2001. doi: 10.1023/A:1013776130161.

Ben Krause, Liang Lu, Iain Murray, and Steve Renals. Multiplicative LSTM for sequence modelling, September 2016. URL https://arxiv.org/abs/1609.07959.

Matthew Larkum. A cellular mechanism for cortical associations: An organizing principle for the cerebral cortex. *Trends in Neurosciences*, 36(3):141–151, 2013. doi: 10.1016/j.tins.2012.11.006.

Maria Lavzin, Sophia Rapoport, Alon Polsky, Liora Garion, and Jackie Schiller. Nonlinear dendritic processing determines angular tuning of barrel cortex neurons in vivo. *Nature*, 490(7420):397–401, 2012. doi: 10.1038/nature11451.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Timothy P. Lillicrap, Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020. doi: 10.1038/s41583-020-0277-3.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. doi: 10.48550/arXiv.1711.05101.

Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, 2013. doi: 10.1038/nature12742.

Sarthak Mittal, Alex Lamb, Anirudh Goyal, Vikram Voleti, Murray Shanahan, Guillaume Lajoie, Michael Mozer, and Yoshua Bengio. Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6972–6986, 2020.

Alexandre Payeur, Jordan Guerguiev, Friedemann Zenke, Blake A. Richards, and Richard Naud. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature Neuroscience*, 24(7):1010–1019, 2021. doi: 10.1038/s41593-021-00857-x.

Leopoldo Petreanu, Tianyi Mao, Scott M. Sternson, and Karel Svoboda. The subcellular organization of neocortical excitatory connections. *Nature*, 457(7233):1142–1145, 2009. doi: 10.1038/nature07709.

Panayiota Poirazi and Athanasia Papoutsi. Illuminating dendritic function with computational models. *Nature Reviews Neuroscience*, 21(6):303–321, 2020. doi: 10.1038/s41583-020-0301-7.

Panayiota Poirazi, Terrence Brannon, and Bartlett W. Mel. Pyramidal neuron as two-layer neural network. *Neuron*, 37(6):989–999, 2003. doi: 10.1016/S0896-6273(03)00149-1.

John H. Reynolds, Leonardo Chelazzi, and Robert Desimone. Competitive mechanisms subserve attention in macaque areas V2 and V4. *Journal of Neuroscience*, 19(5):1736–1753, 1999. doi: 10.1523/JNEUROSCI.19-05-01736.1999.

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.

Shan Shen, Xiaolong Jiang, Federico Scala, Jiakun Fu, Paul Fahey, Dmitry Kobak, Zhenghuan Tan, Na Zhou, Jacob Reimer, Fabian Sinz, and Andreas S. Tolias. Distinct organization of two cortico-cortical feedback pathways. *Nature Communications*, 13:6389, 2022. doi: 10.1038/s41467-022-33883-9.

Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations*, 2023. doi: 10.48550/arXiv.2208.04933. URL `https://openreview.net/forum?id=Ai8Hw3AXqks`.

Spencer L. Smith, Ikuko T. Smith, Tiago Branco, and Michael Häusser. Dendritic spikes enhance stimulus selectivity in cortical neurons in vivo. *Nature*, 503(7474):115–120, 2013. doi: 10.1038/nature12600.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 843–852. PMLR, 2015. doi: 10.48550/arXiv.1502.04681.

STATISTICAL CONVENTIONS

Unless stated otherwise, every reported summary is computed within each of ten independently trained models (seeds 1–10) and then averaged across them; intervals, error bars, and curve shading are the seed-level standard error $\sigma/\sqrt{10}$, and a gray point over a bar is one training seed. Colored points in the connection-level supplementary analyses are connection–condition observations and are marked to distinguish them from seed-level summaries. Every statistic in this paper excludes the first output frame of each recurrent window, at every rollout length, because the hidden state and the feedback vector are both reset at that point.

A GAWF ARCHITECTURE AND PARAMETERIZATION

A.1 MODEL DATA FLOW

Encoder At each recurrent step the encoder receives the current CM-MNIST frame stacked with the preceding one along the channel dimension, giving an input tensor of shape $(B,\, T_{\text{seq}}, 2, 96, 96)$; the channel count is the number of consecutive frames supplied per step (`num_frames`), so that the displacement between adjacent frames is present in the encoder input. A shared convolutional encoder applies $\text{Conv}_{5\times5}(2, 32)$, 2×2 max pooling, layer normalization, and ReLU, then $\text{Conv}_{3\times3}(32, 64)$, 4×4 max pooling, layer normalization, a 1×1 projection from 64 to 32 channels, ReLU, and adaptive average pooling to a 6×6 grid; the resulting $(32, 6, 6)$ activation is flattened in contiguous channel-major order to $\mathbf{x}_t \in \mathbb{R}^m$ with $m = 32 \times 6 \times 6 = 1152$.

Gated recurrent step The recurrent core is a single GaWF layer with $n = 256$ hidden units, whose hidden activation feeds separate 10-class target-identity and 9-class target-location linear readouts. Their raw logits, with no softmax applied, form the feedback vector of Equation (5), which is detached before it enters the next recurrent step, so a loss at time t does not backpropagate through $\mathbf{f}_t$ into the previous readouts, while gradients still flow through the recurrent state. $\mathbf{f}_t$ reaches the cell only through the gate and is clamped elementwise to $[-10, 10]$ before the outer-product expansion; the gate temperature is fixed at $\tau = 0.5$, and the gate multiplies the input and recurrent weights but not the biases. The two factors $\mathbf{U}$ and $\mathbf{V}$ are initialized as independent standard-normal draws scaled by 0.01; the input and recurrent pathways share $\mathbf{U}$, and two column blocks of $\mathbf{V}$ serve the two source spaces. Each rollout initializes both $\mathbf{h}_0$ and $\mathbf{f}_0$ to zero, so every gate equals 0.5 on the first

Table 1: Formal parameter-matched model and optimizer configurations. Total counts all trainable parameters in each model, including its encoder, sequence module, and two readout heads.

Model	Width	Total	LR	WD	Special optimizer rule
RNN	275	586,865	1×10^{-3}	1×10^{-5}	—
LSTM	80	584,515	1×10^{-3}	1×10^{-3}	—
GRU	105	586,695	5×10^{-3}	1×10^{-3}	—
GaWF	256	586,067	5×10^{-3}	1×10^{-3}	Feedback factors: no WD, $1\times$ base LR
Mamba	170	586,935	1×10^{-3}	1×10^{-3}	1-D parameters and $A_{\log}$: no WD
S5	256	586,963	1×10^{-3}	0	Λ, B, $\log \Delta$: $0.1\times$ base LR

frame, which also receives no recurrent input. The implementation further supports stacked GaWF layers, in which a non-final layer receives the adjacent upper-layer hidden state of the preceding step and an optional projection reduces the per-layer feedback dimension; the experiments reported here use the single-layer configuration with unprojected output feedback.

Hidden activity and dropout During training, dropout at rate 0.5 is applied to the state of Equation (4) at every step with an independent mask, so that the stochastic state enters the next update; the same tensor is passed to the readout heads without further transformation. All analyses in Section 3 were performed in evaluation mode, in which dropout is the identity, so hidden activity refers to $\mathbf{h}_t$ exactly as defined in Equation (4), the deterministic recurrent state that the network propagates at evaluation. The same quantity serves as the presynaptic activity $h_{j,t-1}$ in the recurrent-current analysis of Section 3.5, and it is non-negative by construction.

A.2 Parameter matching and hyperparameter selection

Parameter matching is defined over the complete trainable model, including the shared encoder, the recurrent core, and both readout heads. Table 1 reports each model's primary width, total trainable parameter count, and optimizer settings. The width entry is the recurrent hidden size n for RNN, LSTM, GRU, and GaWF and the model dimension d_{model} for Mamba and S5, whose state dimensions are $d_{\text{state}} = 16$ and $d_{\text{state}} = 128$, so it is read according to each architecture. All architectures instantiate the same encoder class under the same PyTorch initialization rules with independently drawn parameter tensors, and the five baselines carry no output-feedback pathway. They also differ in where the state nonlinearity acts: the RNN applies the same layer normalization, rectification, and dropout inside its recurrence as GaWF, whereas LSTM and GRU pass their native layer outputs and Mamba and S5 their block outputs to the readout heads without an additional normalization block. The Mamba baseline uses one Mamba-v1 block with convolution width 4, expansion factor 2, a residual connection, and zero inter-layer dropout; the S5 baseline uses one S5 layer with the same residual and inter-layer dropout settings. Neither state-space baseline applies an additional project-level output-dropout block before the readout heads.

Hyperparameters were selected separately for every architecture, with seed 42 on the fixed 40-minute validation split, at most 100 epochs, and early stopping with patience 15. For GaWF we searched $n \in \{64, 128, 256, 512\}$, learning rate $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$, and weight decay $\{0, 10^{-5}, 10^{-4}, 10^{-3}\}$. Each baseline width was then fixed by matching the parameter count of the selected GaWF configuration, so the baseline searches covered the same learning-rate and weight-decay sets at that single width, with 10^{-2} added to the learning-rate set for Mamba and S5. The grids therefore differ in size, at 64 configurations for GaWF, 16 for each of RNN, LSTM, and GRU, and 20 for each of Mamba and S5, with the difference falling on the width axis that matching had already fixed for the baselines; every candidate received the same per-run training budget. In every case the configuration with the highest validation target-identity accuracy was selected and then retrained with seeds 1–10 for 150 epochs without early stopping. The reported models reuse the learning rates and weight decays selected in this search, which was run on earlier implementations of each architecture, and the widths in Table 1 were re-matched to the parameter counts of the reported implementations.

B CM-MNIST GENERATION AND EVALUATION DATASETS

B.1 STANDARD TRAIN, VALIDATION, AND TEST MOVIES

The generator produces 96×96 grayscale movies at 24 frames per second, each frame holding one foreground target and a set of distractors whose size is drawn at every background-segment onset from $\{1, 2, 4, 8, 12\}$. Every object is an unscaled 28×28 MNIST image, and objects are composited by pixelwise addition with saturation at $[0, 255]$. The foreground keeps one direction within a segment at a speed drawn from $\{0, 1, 2, 3, 4, 6, 8\}$ pixels per frame, starting from a center drawn uniformly over the region in which the full image fits; at a boundary its center is clamped inside the frame and the corresponding velocity component is reversed. Distractors share a speed magnitude drawn from $\{1, 2, 4, 6, 8\}$ and obey the same boundary rule, but each redraws its direction uniformly from $[0, 2\pi)$ on every frame.

Switch intervals are drawn from an exponential distribution with a mean of 24 frames, and each event replaces either the foreground or the complete background set with equal probability. A foreground switch draws a new digit identity and exemplar, a new valid center, and a new speed while keeping the previous direction, with the positive horizontal axis used when the previous speed was zero. A background switch redraws the distractor count, identities, exemplars, centers, shared speed magnitude, and initial directions. The switch flag is attached to the frame after the replacement and the position update, so the labeled frame already shows the new objects.

Each frame is stored with the foreground identity and rendered center, the foreground and background speeds, the distractor identities, and the two switch flags. Target location is the row-major index of a 3×3 grid over the frame. For a target center rendered at pixel coordinates (x, y), with x and y running over the 96 columns and rows,

$$\mathrm{col} = \lfloor \mathrm{clip}(3x/95,\, 0,\, 2) \rfloor\,, \tag{8}$$

$$\mathrm{row} = \lfloor \mathrm{clip}(3y/95,\, 0,\, 2) \rfloor\,, \tag{9}$$

$$s = 3\,\mathrm{row} + \mathrm{col}, \tag{10}$$

where col and row are zero-based grid coordinates, the clip keeps a center on the far edge inside the last cell, and $s \in \{0, \ldots, 8\}$ is the sector label used as the location condition throughout.

The three splits are distinct realizations of this generator, drawn in train–validation–test order from a single fixed-seed stream and using disjoint exemplar pools at indices 0–39,999, 40,000–49,999, and 50,000–59,999. The training movie contains 3,456,000 raw frames and the validation and test movies 57,600 each.

B.2 JOINT-BALANCED TEST MOVIE AND INFERENCE PROTOCOLS

The behavioral analyses of Section 3.1 use a second held-out movie, drawn from the test exemplar pool and keeping the frame size, frame rate, and duration of the standard movies, in which every scheduled event replaces the foreground and the background together. Its schedule holds each of the 90 target-identity-by-target-location combinations exactly 27 times, giving 2,430 events whose frames are drawn uniformly without replacement from frame 2 onward and then sorted; the event count is therefore fixed and the intervals follow from that sampling, in place of the exponential intervals of the standard movies. At each joint switch the foreground and nine distractors carry the ten digit identities exactly once, and their onset centers cover all nine sectors, with one uniformly drawn sector holding a second object. Balance is defined at switch onset, over identity and location; the number of frames each condition occupies between switches is left free.

Six-model accuracy, switch recovery, and the feedback-shuffle analysis all use the joint-balanced movie with 512-frame rollouts. The representation and gate analyses of Section 3 use the standard test movie with 32-frame rollouts, the length used in training.

C TRAINING PROTOCOL

Training uses truncated backpropagation through time over 32 recurrent steps, with the hidden state and the feedback vector reset at the start of each non-overlapping window. The two cross-entropy

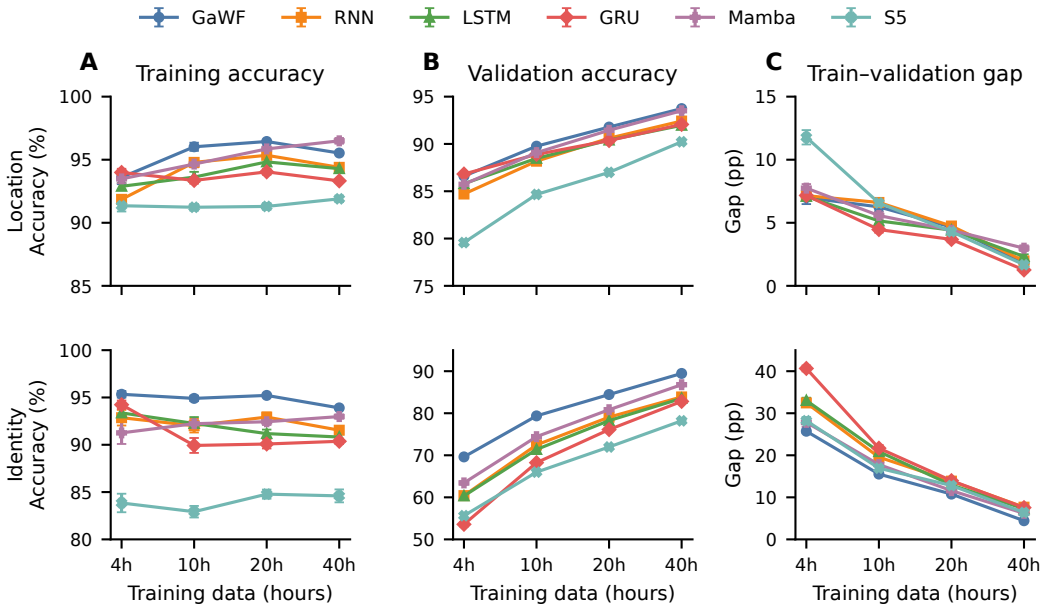


Figure 7: **Performance across training-data scales.** Rows: sector-location (top) and digit-identity (bottom) readouts. **A**: training accuracy. **B**: validation accuracy. **C**: train–validation accuracy gap.

losses, for target identity and target location, are each averaged over batch and time and then summed with equal weight. Optimization uses AdamW with no learning-rate scheduler, a batch size of 256, no gradient accumulation, and automatic mixed precision with gradient scaling; the feedback factors $\mathbf{U}$ and $\mathbf{V}$ are excluded from weight decay, and gradient values and the global gradient norm are both clipped at 1. Encoder dropout is zero and recurrent-output dropout is 0.5. Reported runs train for 150 epochs and retain the checkpoint with the highest validation target-identity accuracy. Target-location accuracy can peak at a different epoch, and that peak leaves the selection unchanged.

D Behavioral evaluation and feedback perturbations

D.1 Test accuracy and switch-aligned recovery

Evaluation uses the complete retained test trajectory, except where an analysis requires balanced condition sampling. The standard 32-frame rollout yields 57,568 output frames per seed, of which 55,769 remain after the window-initial output is excluded, and the joint-balanced 512-frame rollout retains 57,232 frames per seed. Accuracy is the fraction of retained frames whose per-frame argmax matches the corresponding label, computed separately for the identity and location readouts.

Switch-aligned curves use offsets from 10 frames before to 10 frames after a foreground switch, with the first post-switch frame denoted post1 and no offset zero. When two switches fall within twice that radius, frames that the neighboring event could contaminate are masked, and post-switch assignment takes priority over pre-switch assignment. Each offset draws on all available events, so event counts differ across offsets. The curves report absolute accuracy against offset.

D.2 Robustness of the model comparison

Two controls test whether the comparison of Section 3.1 survives a change in the training set and a change in model capacity. The first trains every model at its reported configuration on independently generated movies of 4, 10, 20, and 40 hours, which differ only in duration. The second control sets the hidden width of all six architectures to 128 and trains them on the shortest movie, so width takes the place of parameter count as the matched quantity and the models differ in size, with GaWF the second-smallest. Both keep the hyperparameters selected for the reported runs. The data-scale

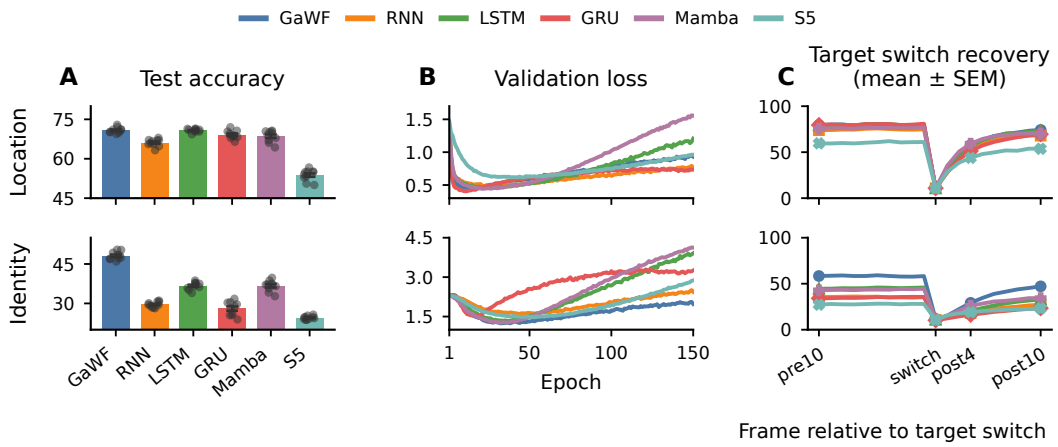


Figure 8: **Six architectures at a reduced common width of 128 on the 4-hour training movie.** Rows: sector-location (top) and digit-identity (bottom) readouts. **A**: test accuracy, dots denoting individual seeds. **B**: validation-loss trajectories. **C**: accuracy aligned to target switches.

control reports each model's own best validation accuracy per readout, so its location values come from a different epoch than the identity-selected checkpoint, whereas the width control is evaluated with the protocol of Figure 2. Both controls use earlier implementations of all six models, in which the GaWF recurrence applied a bounded nonlinearity before the normalization and the baselines passed their outputs through a normalization block, so they probe the robustness of the comparison rather than reproducing the reported configurations exactly.

On the identity readout, GaWF held the highest validation accuracy, the highest training accuracy, and the smallest train–validation gap at every training-data scale averaged over seeds (Figure 7), so its narrower gap did not come from a weaker fit to the training movie. GaWF trained on the 20-hour movie also reached a higher identity validation accuracy (84.4%) than four of the five baselines trained on the longest 40-hour movie (78.2%–83.9%), with Mamba the exception. At the reduced width, GaWF held the highest identity test accuracy by a clear margin, ended training with the lowest identity validation loss, and maintained the highest identity accuracy before and after target switches (Figure 8). The location readout separated the architectures weakly under both controls and did not reproduce the identity ordering: across scales the gap did not order the models consistently, and at the reduced width GaWF, the LSTM, the GRU, and Mamba reached similar test accuracy, with the RNN and S5 lower.

D.3 Feedback shuffling

The shuffle ablation first records the live feedback trajectory of a rollout. It then permutes the temporal indices of the selected feedback segment within each 512-frame recurrent window, independently for every sample, while the other segment remains the live detached feedback that rollout produces. The permutation preserves the marginal distribution of the segment and removes its correspondence to the current frame.

D.4 Baselines with additive feedback

To separate the availability of feedback from its use as connection-level gating, we supplied the RNN, GRU, and LSTM with the same feedback vector $\mathbf{f}_t$ that GaWF receives, namely the 19 raw readout logits of the preceding step, detached, clamped elementwise to $[-10, 10]$, and set to zero at the start of each window. Each variant projects this vector through its own learnable weight matrix and bias, $\mathbf{W}_{\mathrm{fb}}\mathbf{f}_t + \mathbf{b}_{\mathrm{fb}}$, which enters every gate and candidate of the core: the reset, update, and candidate components of the GRU and the input, forget, candidate, and output components of the LSTM. Hidden widths were re-matched to the parameter count of the reported models (RNN, $n = 272$; GRU, $n = 103$; LSTM, $n = 79$), and learning rates and weight decays were reused from

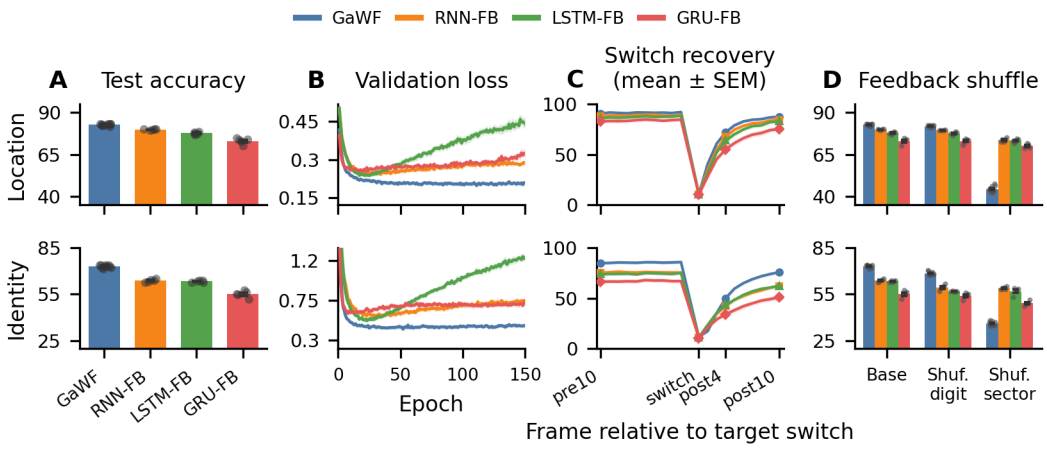


Figure 9: **GaWF outperforms recurrent baselines that receive the same feedback additively.** Rows: sector-location (top) and digit-identity (bottom) readouts for GaWF and three recurrent baselines with additive feedback. **A**: test accuracy, dots denoting individual seeds. **B**: validation-loss trajectories. **C**: accuracy aligned to target switches. **D**: accuracy under inference-time shuffling of the sector or the digit feedback segment.

the corresponding baselines without a new search. The additive-feedback cores applied no recurrent dropout, unlike GaWF and the reported RNN. The encoder, readout heads, and remaining training protocol are those of the reported baselines; each variant was trained with five seeds, and GaWF is shown with its ten reported seeds. Test accuracy, switch recovery, and feedback shuffling follow the evaluation protocol of Figure 2.

GaWF held the highest test accuracy on both readouts against all three feedback-augmented baselines, and the same ordering held when GaWF was restricted to the five matching seeds (Figure 9A). The validation loss of the feedback-augmented baselines rose after an early minimum, most steeply for the LSTM, while that of GaWF remained low (Figure 9B). After target switches, GaWF recovered to the highest identity accuracy, with the location curves of all models lying close together (Figure 9C). Shuffling the sector segment reduced GaWF accuracy substantially on both readouts, whereas the feedback-augmented baselines declined by a smaller margin under either shuffle (Figure 9D).

E GATE-VALUE AND EFFECTIVE-WEIGHT DISTRIBUTIONS

Each gate-value histogram in Figure 3A, B pools the g^{in} values (256×1152 per frame) or g^{rec} values (256×256 per frame) across every retained frame of each test trajectory, accumulated within a seed into 400 equal-width bins on $[0, 1]$. Effective-weight histograms use 500 bins on a seed-specific symmetric interval $[-\max|w|, +\max|w|]$, accumulate the static weights w and the framewise effective weights gw, and are reprojected onto a common symmetric grid across seeds. The vertical axis of Figure 3 is probability mass per bin.

Both gates carry a point mass at $g = 0.5$ before reset frames are excluded, at 3.125% under the criterion $|g - 0.5| < 10^{-6}$. The zero-feedback first frame of each recurrent window accounts for it exactly: 1,799 of the 57,568 outputs per seed are reset frames, and zero feedback gives $\sigma(0) = 0.5$ at every connection. Outside reset frames, values within that tolerance of one half account for less than 0.001% of either gate, and no destination unit is disconnected from feedback under the analogous criterion $\max_k |u_{ik}| < 10^{-6}$, so the point mass is confined to the reset frames.

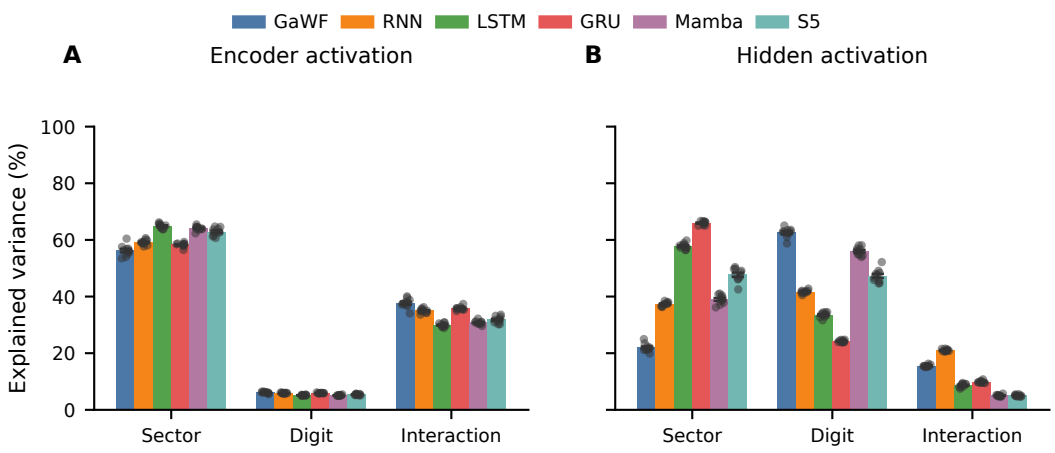


Figure 10: **Condition-mean variance decomposition of activations across six architectures. A**: encoder activations. **B**: hidden activations. Bars give the sector-location, digit-identity, and interaction fractions.

F Balanced task-variable variance decomposition

F.1 Balanced sampling and decomposition

This section and the following two share one balanced-sampling operation unless stated otherwise. Retained frames are partitioned into the 90 target-identity-by-target-location cells, whose sizes differ, and every cell is subsampled without replacement to the smallest of them, 211 frames, giving one draw of 18,990 frames in which each identity-location combination carries equal weight. Cells above that size discard part of their frames, so each statistic is computed on 20 such draws and the 20 estimates are averaged within each training seed, which keeps the reported value from depending on which frames a single draw retained. The same 20 draws serve every model and analysis, and the ten training seeds remain the only inference units.

The 90 balanced cell means are decomposed in the standard two-way form: a location component, each sector's marginal mean minus the grand mean; an identity component, each digit's marginal mean minus the grand mean; and an interaction component, what remains of a cell mean once the grand mean and both marginal deviations are removed. The condition-mean decomposition reports the sums of squares of the three components normalized by their total, so location, identity, and their interaction sum to 100% and trial-level residual variance falls outside the accounting. The trial-level decomposition multiplies those three components within each seed by one minus the residual fraction, and adds the residual sum of squares as a fraction of the total trial-level sum of squares as its fourth component, which again sums to 100%.

F.2 Activation and gate specialization

Figure 10 applies the condition-mean decomposition to the encoder and hidden activations of all six architectures. For comparison with the unit-level gates of the LSTM and the GRU, each GaWF gate is projected onto its destination unit as the unweighted mean of the gate values on all connections entering that unit, 1152 for the input gate and 256 for the recurrent gate. This comparison holds the level of description fixed and leaves connection-level granularity confounded with the use of output-derived feedback.

G Spatial organization of the input gate

We test whether location feedback preferentially opens input connections from spatially aligned encoder positions, given that encoder activation and input-gate modulation are both organized by target location. The encoder output is reshaped from its contiguous flattening order to $(32, 6, 6)$,

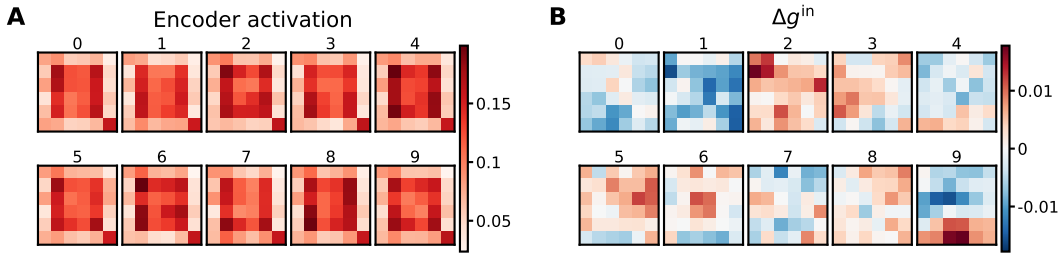


Figure 11: **Encoder activation and input-gate modulation conditioned on target identity. A**: encoder activation averaged over feature channels and mapped onto the encoder spatial grid, one map per digit. **B**: marginal Δg^{in} for each digit, averaged over hidden units and channels and mapped onto the same grid.

Table 2: **Input-gate modulation by source group and weight sign.** Entries average Δg^{in} within each sector condition and then equally over the nine conditions. Sign-specific rows restrict positive- and negative-weight connections to their shared $|w^{\text{in}}|$ range; the gap is the w_+^{in} row minus the w_-^{in} row, and All averages over all nonzero connections of the group.

	Matched	**Other**
w_+^{in}	0.3530 ± 0.0048	-0.0441 ± 0.0006
w_-^{in}	0.3658 ± 0.0050	-0.0457 ± 0.0006
Gap	-0.0127 ± 0.0010	0.0016 ± 0.0001
All	0.3595 ± 0.0049	-0.0449 ± 0.0006

and each cell of the 3×3 target-location grid maps to a non-overlapping 2×2 block within every channel. A location therefore has 128 matching sources, 32 channels at four grid positions, and 1,024 other sources. No receptive-field overlap threshold enters the assignment.

Using the balanced frames of Section F, we average the framewise deviation of Equation (6) within each sector condition, which gives the connection's marginal Δg, its marginal mean for that sector minus its grand mean $\overline{g}_{ij}^{\text{in}}$; because every condition holds the same number of balanced frames, this grand mean equals the equal-weighted mean of the connection's condition means, and the marginal Δg is the connection's location component as defined in Section F. Spatial maps average the resulting values over destination units and channels. The sign and magnitude analyses take each nonzero connection under one condition as an observation, valued at its marginal Δg, and use all observations in nine $|w|$-quantile bins shared across signs and seeds; slopes are separate ordinary least-squares fits of Δg on $|w|$ within each seed over the shared $|w|$ support of the two signs. Displayed points are a fixed-seed subsample of up to 900 observations per sign and panel, pooled over seeds (Figure 12). Group means are reported in Table 2.

For comparison, we apply the same conditioning to target identity, averaging encoder activation and the framewise deviation of Equation (6) within each digit condition, so that the marginal Δg^{in} of a digit is the connection's identity component in the sense of Section F. Across digits, the encoder maps share one spatial profile (Figure 11A), and the marginal Δg^{in} maps are spatially diffuse (Figure 11B).

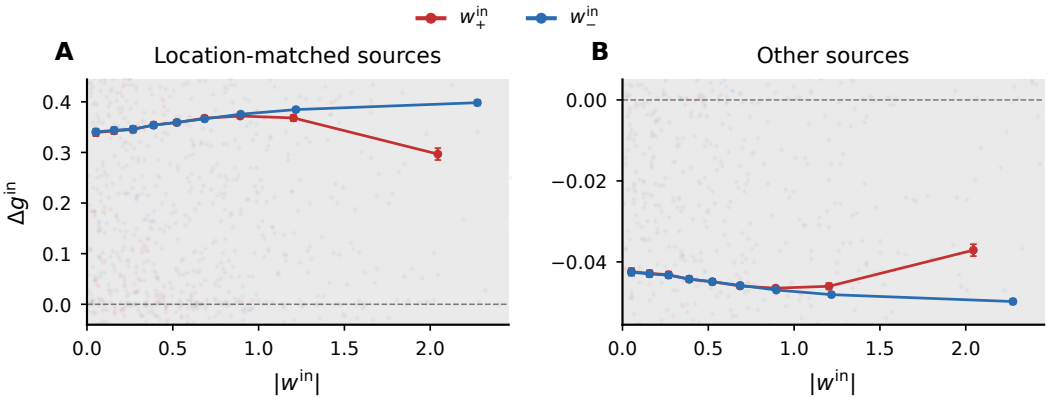


Figure 12: **Input-gate modulation against static input-weight magnitude. A**, **B**: marginal Δg^{in} versus $|w^{\text{in}}|$ for w_+^{in} and w_-^{in} at location-matched (**A**) and other (**B**) encoder sources. Faint points are a display-only subsample of connection–condition observations, and curves are nine-bin means.

H TUNED ENSEMBLES, RECURRENT-GATE MODULATION, AND RECURRENT CURRENT

H.1 CONDITION-SPECIFIC TUNED ENSEMBLES

Tuned and remainder masks are defined from validation-split hidden activations and then applied to an independent test split for the gate and current analyses that use them. For each hidden unit we run 1,000 stratified permutations: sector labels are shuffled within digit strata for location tuning, and digit labels within sector strata for identity tuning. The statistic is that factor's trial-level fraction as defined in Section F, and the one-sided p-value is $(1+k)/1001$, where k is the number of permuted statistics at least as large as the observed one. A Benjamini–Hochberg correction at $\alpha = 0.05$ is applied separately to each factor's 256 unitwise p-values, with the adjusted value at or below α defining a passed factor, and a unit is marked interaction-dominant when its interaction fraction exceeds both marginal-factor fractions. The eligible pool P for a task variable holds the units that pass that factor and are not interaction-dominant.

For each condition, T holds the top $\lceil 0.1|P| \rceil$ units of the corresponding pool ranked by standardized marginal tuning, defined for a unit as its equal-cell marginal mean for that condition, centered and scaled by the mean and standard deviation of its marginal means across that factor's levels. R holds all other hidden units, so it contains the units of P outside T together with those that failed the factor test or were interaction-dominant; $|T|$ was between 25 and 26 and $|R|$ between 230 and 231 across the ten seeds. The sets are condition-specific and may overlap: across the 45 digit pairs the normalized overlap $|T_a \cap T_b|/\sqrt{|T_a||T_b|}$ was 9.8%, against 10.3% expected if the two sets were drawn independently from the same pool. Recurrent groups follow the source-to-destination order $T \to T$, $T \to R$, $R \to T$, $R \to R$, and self-connections are retained.

H.2 RECURRENT-GATE LOCALIZATION AND SIGN DEPENDENCE

Recurrent observations follow Section G, with conditions from either task variable and each observation assigned to a group by the tuning of its source and destination under that condition, so a connection whose endpoints are tuned under several conditions contributes one observation per condition. Within each group, sign-specific means are computed over the shared positive and negative $|w|$ support, and are reported in Table 3 and shown against weight magnitude in Figure 13.

Every entry of Figure 6A and C is computed within each seed in two steps: a mean over the frames of one condition and the relevant nonzero connections of the group under that condition, followed by an equal-weight mean over the ten digit or nine sector conditions. The sign-specific gate entries use the shared-$|w|$ subset, so the all-connections gate entry is not recoverable from them.

Table 3: **Recurrent-gate modulation by connection group and weight sign.** Entries of Δg^{rec} are computed as in Figure 6A. Sign-specific rows restrict positive- and negative-weight connections to their shared $|w^{\mathrm{rec}}|$ range; the gap is the w_+^{rec} row minus the w_-^{rec} row, and All averages over all nonzero connections of the group.

	$T \to T$	$T \to R$	$R \to T$	$R \to R$
Target identity				
w_+^{rec}	-0.0176 ± 0.0050	-0.0282 ± 0.0023	-0.0039 ± 0.0009	0.0038 ± 0.0003
w_-^{rec}	-0.1575 ± 0.0121	-0.0223 ± 0.0029	-0.0050 ± 0.0017	0.0044 ± 0.0004
Gap	0.1398 ± 0.0083	-0.0058 ± 0.0029	0.0011 ± 0.0016	-0.0006 ± 0.0004
All	-0.0614 ± 0.0083	-0.0253 ± 0.0022	-0.0043 ± 0.0011	0.0041 ± 0.0003
Target location				
w_+^{rec}	-0.0650 ± 0.0029	-0.0492 ± 0.0022	-0.0053 ± 0.0007	0.0070 ± 0.0003
w_-^{rec}	-0.0601 ± 0.0030	-0.0500 ± 0.0017	0.0055 ± 0.0007	0.0057 ± 0.0002
Gap	-0.0049 ± 0.0021	0.0008 ± 0.0010	-0.0107 ± 0.0007	0.0013 ± 0.0002
All	-0.0627 ± 0.0027	-0.0495 ± 0.0019	0.0005 ± 0.0006	0.0063 ± 0.0002

Table 4: **Connection-normalized gate-dependent recurrent current by connection group and weight sign.** Entries of ΔI^{gate} are computed as in Figure 6C; All averages over all nonzero connections of the group and is not the sum of the rows above.

	$T \to T$	$T \to R$	$R \to T$	$R \to R$
Target identity				
w_+^{rec}	-0.0089 ± 0.0011	-0.0123 ± 0.0010	-0.0052 ± 0.0003	-0.0072 ± 0.0003
w_-^{rec}	0.0862 ± 0.0046	0.0170 ± 0.0023	0.0192 ± 0.0010	0.0117 ± 0.0005
All	0.0214 ± 0.0019	0.0032 ± 0.0014	0.0081 ± 0.0006	0.0024 ± 0.0004
Target location				
w_+^{rec}	-0.0197 ± 0.0012	-0.0161 ± 0.0009	-0.0068 ± 0.0003	-0.0062 ± 0.0003
w_-^{rec}	0.0260 ± 0.0013	0.0291 ± 0.0016	0.0121 ± 0.0007	0.0114 ± 0.0006
All	0.0002 ± 0.0009	0.0072 ± 0.0008	0.0031 ± 0.0005	0.0028 ± 0.0005

H.3 GATE-DEPENDENT RECURRENT CURRENT

Equation (7) holds each gate at its grand mean $\overline{g}_{ij}^{\mathrm{rec}}$, defined as in Equation (6), while retaining the observed hidden activity, so it is an instantaneous decomposition and not a rollout with the gates held fixed. Weight-class entries average $\Delta I_{ij,t}^{\mathrm{gate}}$ over the nonzero positive- or negative-weight connections of a group and the all-connections entry over their union, so within each condition the latter is a connection-count-weighted mean of the two class entries and equals neither their sum nor their equal-weight average (Table 4).

The same per-connection terms also sum over the connections entering a destination unit, an intermediate level of aggregation that carries the number of connections a group holds rather than dividing it out. Because $|T|$ lies between 25 and 26 while R holds the remaining hidden units, the four groups differ by about an order of magnitude on this axis. Figure 14 reports $\Delta I_{ij,t}^{\mathrm{gate}}$ summed over the connections entering each destination unit and then averaged over destination units, taken for one weight sign or for both; here the all-connections entry is the sum of the two class entries.

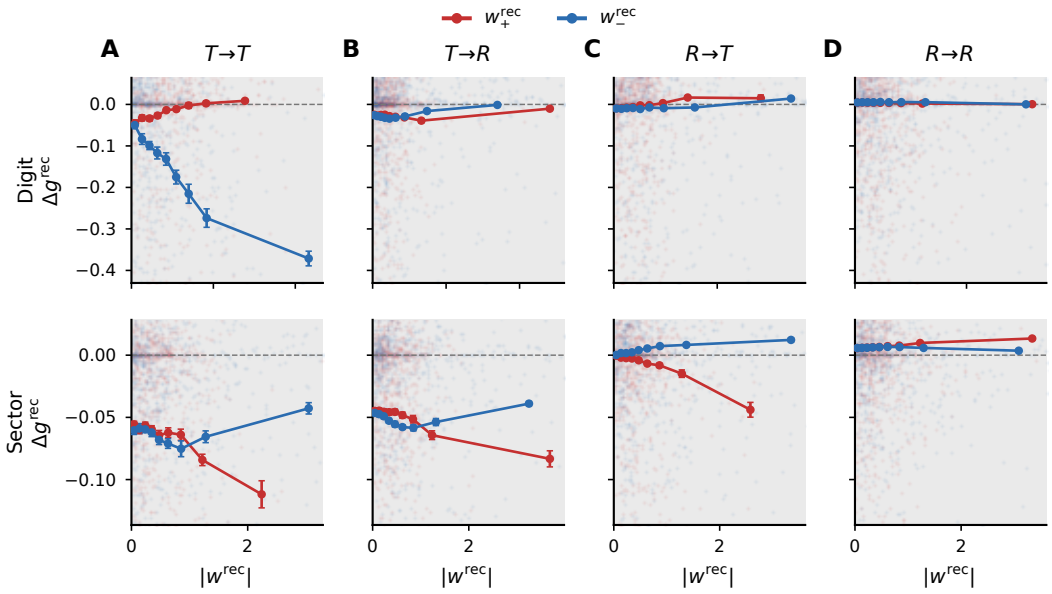


Figure 13: **Recurrent-gate modulation against static recurrent-weight magnitude.** Rows: digit (top) and sector (bottom) conditions. **A–D**: marginal Δg^{rec} versus $|w^{\mathrm{rec}}|$ for w_+^{rec} and w_-^{rec}. **A**: $T \to T$. **B**: $T \to R$. **C**: $R \to T$. **D**: $R \to R$. Faint points are subsamples of connection–condition observations, and curves are nine-bin means.

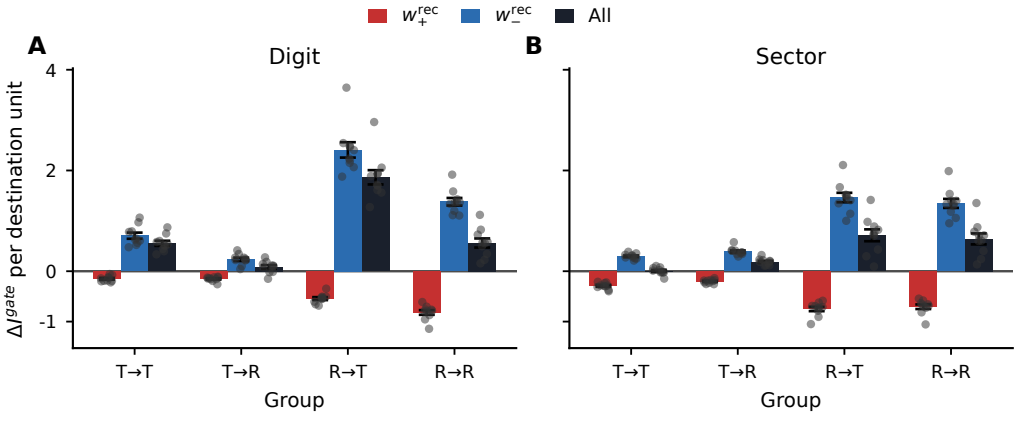


Figure 14: **Gate-dependent recurrent current per destination unit.** **A**: digit conditions. **B**: sector conditions. Bars give ΔI^{gate} summed over the connections entering each destination unit and then averaged over destination units, for w_+^{rec}, w_-^{rec}, and all connections of a group, in the source-to-destination order $T \to T$, $T \to R$, $R \to T$, $R \to R$. Under this normalization the all-connections entry is the sum of the two class entries.