HACANN: GATED DELTANET AS THE NATIVE DYNAMICS OF CONTINUOUS ATTRACTOR NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Attention mechanisms like Gated DeltaNet drive modern large language models, yet whether such associative computations can natively emerge from biological neural circuits remains an open question. Here, we introduce Hierarchical Adaptive Continuous Attractor Neural Networks (HACANN), developing the hypothesis that continuous attractors serve as canonical building blocks for relational memory. In HACANN, coupled recurrent rate populations encode association components, enabling cues to retrieve predictions and steer error-driven corrections as new evidence arrives. Theoretically, we prove that the projected dynamics of these biologically plausible rate circuits yield a generalized gated Delta rule, exactly recovering the canonical Gated DeltaNet recurrence in a defined timing regime. Empirically, full spatial population simulations with over 4.4 million rate variables sustain both stable prediction and ongoing learning. On WikiText-2, HACANN achieves natural language prediction performance comparable to matched Gated DeltaNet models. Further semantic and visual experiments demonstrate selective revision of stored relations and activity-driven feedback calibration that restores representation learning. Together, our results provide a formal circuit foundation for modern linear attention and map out a pathway toward building large-scale language models entirely from neural population dynamics.

1 INTRODUCTION

Memory makes experience available to a later question. A familiar name retrieves what is known about a person; a new statement changes the relevant association. Attention gives this operation a computational form. The Transformer placed retrieval by content at the center of language modeling (Vaswani et al., 2017). Linear attention carries the accumulated associations in a recurrent memory (Katharopoulos et al., 2020; Ba et al., 2016; Schlag et al., 2021). Gated DeltaNet updates that memory by comparing incoming content with its retrieved prediction and correcting the discrepancy (Yang et al., 2025). Qwen3-Next uses this mechanism, and Kimi Linear extends it through Kimi Delta Attention (Qwen Team, 2025; Kimi Team, 2025). Their use in language models gives the neural basis of associative memory direct computational significance.

Continuous attractor neural networks (CANNs) describe how a neural population can retain a continuously varying quantity. Recurrent excitation and inhibition sustain an activity pattern, and inputs move its position. This organization accounts for the maintenance and updating of gaze, heading, and location (Seung, 1996; Zhang, 1996; Samsonovich & McNaughton, 1997). Population recordings and perturbations reveal the corresponding geometry and dynamics in navigation circuits (Kim et al., 2017; Chaudhuri et al., 2019; Gardner et al., 2022). Si Wu and colleagues developed a projection method for analyzing these collective motions and proposed CANNs as canonical modules of neural representation (Wu et al., 2008; Fung et al., 2010; Wu et al., 2016). The associative question is how populations that retain quantities can together retain and revise a relation.

We introduce *Hierarchical Adaptive Continuous Attractor Neural Networks* (HACANN). Each adaptive population maintains one component of an association. Together, their coordinates specify a mapping from a cue to remembered content. A cue reads this mapping and directs a prediction error back to the coordinates that contributed to its answer. The recurrent connections within a popu-

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.

lation supply persistence; connections between populations organize retrieval and revision. Learned projections compose these operations into a hierarchy. This construction gives a persistent neural activity coordinate the computational role of a changing association weight.

Starting from neural rate equations, we derive a memory coordinate carried jointly by activity, adaptation, and filtered drive. Coupled coordinates obey a generalized gated Delta rule whose gates express the response unfolded during an input. A specified timing regime recovers canonical Gated DeltaNet exactly. Full spatial simulations with over 4.4 million rate variables sustain prediction and continued learning. Language, semantic, and visual experiments connect this proposed canonical neural operation to learned behavior. Section 2 reviews the computational lineage this construction joins; Section 6 and Appendix I relate it to further neural and artificial models of memory.

2 Attention as Associative Memory

Retrieval. Attention uses three representations. A *key* is the cue under which content is stored, a *value* is that content, and a *query* specifies what is sought. Softmax attention compares a query with earlier keys and combines their values according to similarity (Vaswani et al., 2017). Its connection to associative retrieval also appears in modern Hopfield networks and distributed memory (Ramsauer et al., 2021; Bricken & Pehlevan, 2021).

Linear attention accumulates associations in a matrix. Let i index previous inputs and t the current input. Write k_i and v_i for their key and value vectors, q_t for the current query, and $\top$ for transpose. Keys and queries have the same dimension. The memory M_t and its retrieved answer are

$$M_t = \sum_{i \le t} v_i k_i^\top, \qquad M_t q_t = \sum_{i \le t} v_i (k_i^\top q_t). \tag{1}$$

The outer product stores a relation. Cue overlap determines how much each stored value contributes to the answer. This matrix has a fixed number of entries, so each new input can update and read it at a cost independent of the preceding sequence length (Katharopoulos et al., 2020). Normalized linear attention also divides by accumulated similarity; Equation (1) gives its associative numerator.

Revision. An experience can change an existing association. For current memory M, incoming key k, value v, and correction strength β, the Delta rule produces updated memory M_+ as (Schlag et al., 2021; Yang et al., 2024)

$$M_+ = M + \beta(v - Mk)k^\top. \tag{2}$$

Memory first predicts Mk. The unexplained content $v - Mk$ then guides its revision. For a unit key, the answer moves a fraction β toward the new value. Another query q changes by $\beta(v - Mk)(k^\top q)$. Thus cue overlap determines which answers change together; orthogonal cues preserve their answers. The geometry of the representation controls the generalization of an update.

Retention. Gated DeltaNet retains a fraction $\alpha \in [0, 1]$ of memory before applying a correction of strength $\beta \in [0, 1]$. For unit keys and identity matrix I, its recurrence is (Yang et al., 2025)

$$M_+ = \alpha M(I - \beta k k^\top) + \beta v k^\top. \tag{3}$$

The input determines both gates. Several memories, called *heads*, operate in different learned cue spaces, and successive layers transform their answers. A neural realization therefore needs a persistent association, a pathway that reads it, and a return pathway that converts a discrepancy into a directed change.

3 From a Neural Population to an Associative Circuit

3.1 The Coordinate Retained by an Adaptive Population

A continuous attractor separates persistence from adjustment through the geometry of its recurrent connections. Nearby neurons reinforce one another, while shared inhibition limits activity. These interactions restore the shape of a localized pattern after a small disturbance. Translating the pattern

preserves their balance. Its position can retain a value and integrate an input that moves it (Amari, 1977; Wu et al., 2008; Claudi et al., 2025).

We use the adaptive CANN studied by Si Wu and colleagues (Fung et al., 2010; Mi et al., 2014; Li et al., 2025). Let x label a neuron's preferred coordinate and a dot denote a time derivative. Synaptic activity $U(x)$ produces firing rate $r(x)$; adaptation $V(x)$ supplies delayed inhibition; filtered drive $g(x)$ moves the pattern. Their response times are τ_U, τ_V, and τ_g. The recurrent excitatory kernel is J, shared inhibition has strength κ, and adaptation has strength m. Choose $\gamma = \tau_U - m\tau_V > 0$. Write $J * r$ for spatial convolution, $[U]_+ = \max(U, 0)$, and x' for an integration coordinate. The motion profile $T_\delta[U]$ is a normalized difference of recurrent projections shifted by distance δ in opposite directions. For signed input amplitude F, the population obeys

$$\begin{aligned} &\tau_U \dot{U} = -U + J * r - V + \gamma g, \quad \tau_V \dot{V} = -V + mU, \\ &r(x) = \frac{[U(x)]_+^2}{1 + \kappa \int [U(x')]_+^2 \, \mathrm{d}x'}, \qquad \tau_g \dot{g} = F T_\delta[U] - g. \end{aligned} \tag{4}$$

Shared inhibition controls gain (Heeger, 1992; Carandini & Heeger, 2012). Two nonnegative actuator populations with fixed excitatory and inhibitory outputs implement the signed motion pathway (Appendix A). Offset pathways provide a physiological mechanism for shifting a maintained heading representation (Turner-Evans et al., 2017; 2020). The condition $\gamma > 0$ makes the relative displacement of activity and adaptation relax.

During a transient, the activity center can continue to move after an input ends. Adaptation and filtered drive carry part of the displacement still to unfold. The retained coordinate includes this pending motion. To derive it, we project onto the *translation mode*, the collective motion that shifts the pattern while preserving its shape. Let Q denote the activity center, ψ its sensitivity to a spatial perturbation, and angle brackets denote spatial integration. The joint coordinate Z satisfies

$$Z = Q + \left\langle \psi, \tau_g g - \frac{\tau_V}{\gamma}(V - mU) \right\rangle, \qquad \dot{Z} = F. \tag{5}$$

Thus Z is the position that remains when the translation transients relax. At rest, $Z = Q$. During motion, this readout preserves access to a value, consistent with stable readout amid changing activity (Druckmann & Chklovskii, 2012; Murray et al., 2017). Appendix A derives Q, ψ, and the contribution of shape deformation in the full spatial field. The projected identity in Equation (5) supplies the integrator used below.

3.2 Coupling coordinates to store a relation

Let memory entry $M_{ij} = Z_{ij}$ be the coordinate of population (i, j), where i indexes content components and j indexes cue components. Multiplying these coordinates by a key and summing retrieves associated content. Two further sets of coordinates maintain a prediction p and an evidence trace u, both with the dimension of the value v. Prediction follows the retrieved answer Mk; evidence follows incoming content. Lowercase u denotes this evidence vector, distinct from the spatial activity field U.

Let $\rho \geq 0$ set the acquisition and correction rate, $\lambda \geq 0$ the relaxation rate toward the association supported by current evidence, and $\theta > 0$ the prediction response time relative to evidence. A query q reads output o. We couple the coordinates through

$$\begin{aligned} &\dot{M} = -\lambda(M - uk^\top) + \rho(v - p)k^\top, \\ &\theta \dot{p} = \rho(Mk - p), \qquad \dot{u} = \rho(v - u), \qquad o = Mq. \end{aligned} \tag{6}$$

The first memory term relaxes toward the association carried by evidence; the second corrects a prediction. Population (i, j) receives force $F_{ij} = -\lambda Z_{ij} + [\lambda u_i + \rho(v_i - p_i)]k_j$. A value row shares its discrepancy, and a key column supplies the gain that determines where it acts. The same cue therefore participates in reading an association and directing its revision. Closing both rates preserves memory and traces.

Opponent populations carry signed signals. Rate modulation combines row and column inputs, with gain modulation and dendritic gating providing circuit precedents (Chance et al., 2002; Yang et al., 2016). A head with d_k key components and d_v value components uses $d_v d_k$ memory populations

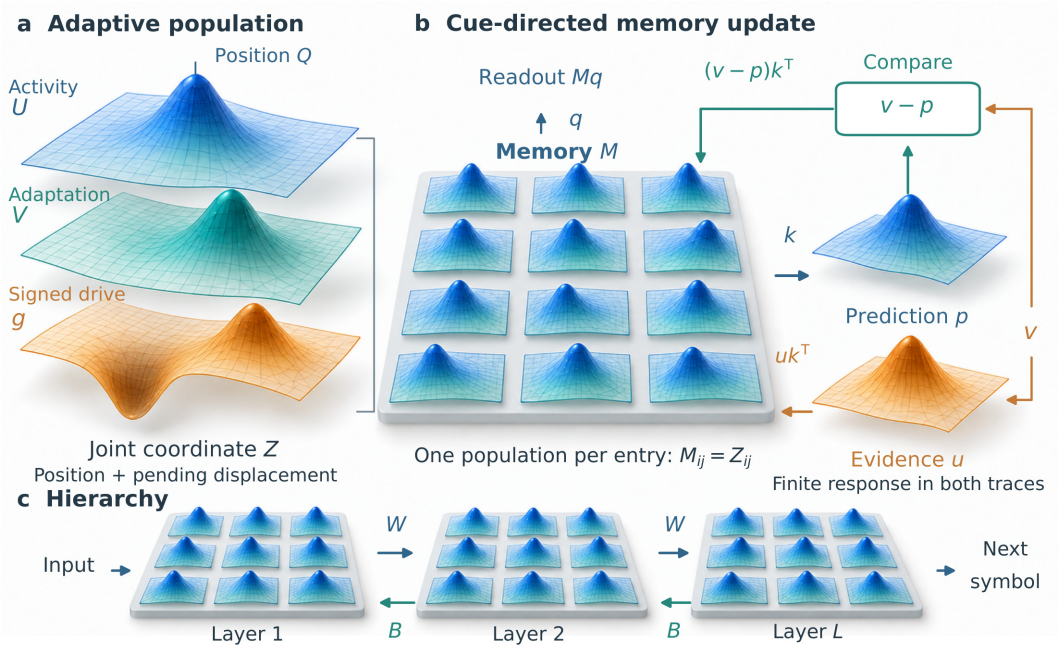

Figure 1: **Adaptive populations form an associative hierarchy.** (a) Activity U, adaptation V, and signed drive g jointly carry coordinate Z. (b) Each memory entry is a population coordinate. Cue k retrieves prediction p and directs discrepancy $v - p$ back to memory; evidence u follows the value. Query q reads Mq. (c) Forward projections W compose layers; separately stored feedback projections B support learning. Population surfaces and arrangements are schematic.

and $2d_v$ trace populations. Learned synaptic projections connect these heads into layers (Figure 1). Appendix G specifies the complete information flow and the rate realizations of input encoding, normalization, temporal filtering, nonlinear transformations, and output probabilities.

4 The Finite Neural Response Yields Gated Delta Computation

4.1 The Response to One Input

Hold the key, value, and rates fixed for an interval of duration Δ. Let s denote elapsed time, M the incoming memory, and M_+ the memory after the interval. Integrating Equation (6) gives

$$M_+ = e^{-\lambda\Delta} M + \left\{ \int_0^\Delta e^{-\lambda(\Delta - s)} [\lambda u(s) + \rho(v - p(s))] \mathrm{d}s \right\} k^\top. \tag{7}$$

The first term retains earlier memory. The second accumulates evidence and prediction error over the neural response. Every write follows the current key; orthogonal directions decay. All coefficients follow from the response times, input duration, and key norm (Appendix B). Carrying the prediction and evidence traces forward gives a generalized gated Delta recurrence, with $2d_v$ additional coordinates and $O(d_v d_k)$ computation per input. Its exact finite solution enables efficient training of the projected neural dynamics; Appendix D derives parallel evaluation.

Correspondence with Gated DeltaNet. For a unit key, take $\theta \to 0$, so prediction follows the retrieved content without lag. Set the retention fraction $\alpha = e^{-\lambda\Delta}$ and correction fraction $\beta = 1 - e^{-\rho\Delta}$. With u denoting evidence entering the interval, the response is

$$M_+ = \alpha M(I - \beta k k^\top) + [\beta v + (1 - \alpha)(1 - \beta)u] k^\top. \tag{8}$$

Clearing incoming evidence recovers canonical Gated DeltaNet, Equation (3), exactly. A decay phase followed by a correction phase gives the same rule in the fast prediction limit. The experi-

ments retain finite prediction and continuing evidence. Their recurrence expresses the circuit's full response at those timescales.

The coupling also constrains sensitivity to the incoming state. For trajectories with identical inputs, gates, and evidence, memory and prediction exchange a discrepancy through the same cue with opposite signs. The cross terms cancel in a quadratic difference energy, leaving only decay and prediction relaxation. The energy is consequently nonincreasing (Appendix B).

4.2 Learning the representations that address memory

Population motion changes the current association. Synaptic learning changes the cues and values through which experience is represented. Each forward projection W has a separately stored return projection B. Endpoint activity probes reveal the difference between their forward and return responses. Local activity products calibrate feedback (Akrout et al., 2019); for fixed forward weights, the averaged calibration dynamics approach reciprocal action, $B = W^\top$. This supplies a teaching pathway for learned representations (Lillicrap et al., 2016; Sacramento et al., 2018).

Observed targets provide output errors. Reverse replay of stored inputs and reconstructed activities propagates temporal credit through the return pathways. Eligibility traces and local synaptic response dynamics convert activity and credit into parameter changes (Bellec et al., 2020). Language memory continues across 1,024-event credit windows. Appendices C–G specify the replay, feedback calibration, and plasticity required by the complete model.

5 Experiments

The experiments connect the circuit coordinate to prediction, selective revision, and learned representations. HACANN uses finite prediction with $\theta = 0.25$. Language comparisons use three training seeds; the other studies use two. Appendix H gives all configurations, protocols, and individual outcomes.

5.1 Spatial populations sustain language memory

We first ask whether the spatial fields preserve the computation derived by projection. Every memory, prediction, and evidence coordinate of a four-layer language model is instantiated as a population field. The 17,408 coordinates each have 64 sites for excitation, adaptation, and two actuator populations, totaling 4,456,448 rate variables per stream.

Two trained models initialize paired spatial and projected implementations and continue learning for 32 windows of 1,024 bytes. Spatial execution integrates the fields; temporal credit follows the projected dynamics along their trajectory. On the complete validation article, spatial loss improves by 0.01028 and 0.01249 bits per byte (bpb). The corresponding projected improvements differ by less than 0.00008 bpb (Figure 2). The populations preserve predictive associations while the surrounding representations continue to learn.

A second experiment tests what the population retains during a transient. Equal-energy pulses move the activity center Q while preserving joint memory Z, or move Z while preserving Q. We then compare predictions with an unperturbed continuation over 64 targets. Moving Q alone produces mean Kullback–Leibler divergence 3.76×10^{-12} nats; moving Z alone produces 8.38×10^{-4} nats. The causal effect at this checkpoint follows the derived coordinate. Adaptation and filtered drive contribute information that remains available to later retrieval. This gives a physiological prediction: perturbations that separate visible position from the joint coordinate should have different effects on later recall.

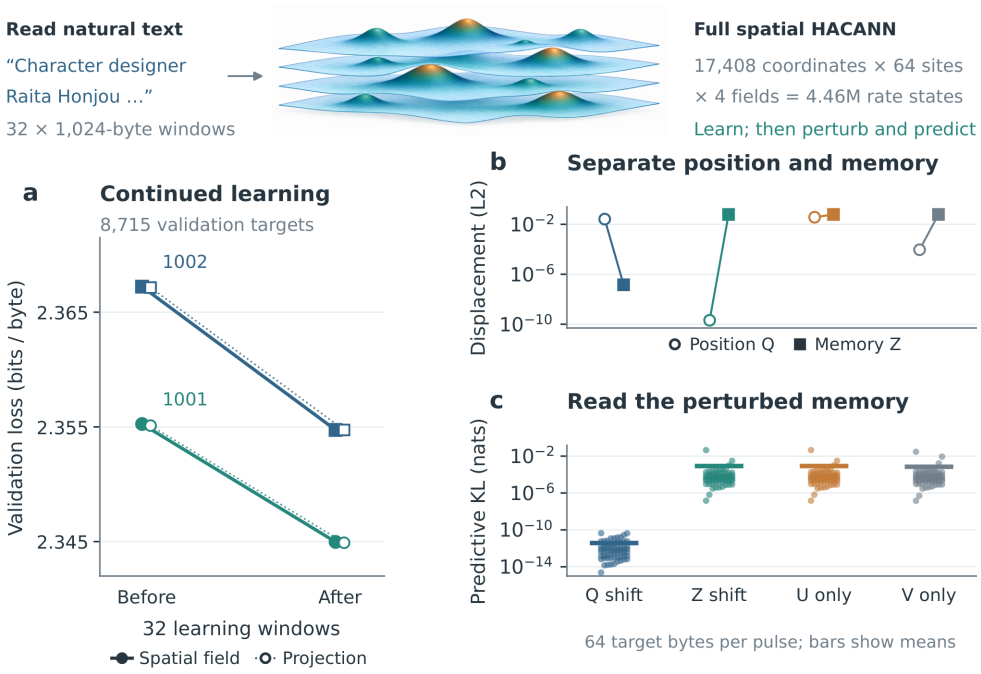

Figure 2: **Spatial populations carry predictive memory.** (a) Validation loss before and after continued learning, with paired field and projected executions for both seeds. (b) Endpoint displacements under equal-energy pulses that separate activity center Q from joint memory Z. (c) Predictive KL divergence from sham: each point is one of the same 64 subsequent targets, and horizontal marks give means. Panels (b,c) use one checkpoint and pulse direction. Full trajectories appear in Figures S1 and S2.

5.2 The neural recurrence learns natural language

We next test whether the finite neural response supports language learning. The task is next-token prediction with UTF-8 bytes as tokens. A crossed comparison uses HACANN or Gated DeltaNet, each with independently calibrated or exact feedback. All four conditions have 943,008 forward parameters, four layers of width 128, four heads, and the same surrounding processing block. Each receives 50.3 million target-byte presentations from WikiText-2 (Merity et al., 2017). Memory continues within articles; temporal credit spans 1,024 bytes. Validation selects the checkpoint, and testing covers all 1,289,061 targets in 62 articles.

HACANN with independent feedback reaches 2.2455 ± 0.0234 bpb, alongside 2.2507 ± 0.0348 for Gated DeltaNet with exact feedback. HACANN with exact feedback reaches 2.2498 bpb; Gated DeltaNet with independent feedback reaches 2.2493 bpb (Figure 3). Variation is the sample standard deviation across three seeds. Both the finite neural response and independently calibrated feedback support comparable predictive performance in this matched setting.

To identify the contribution of context, we clear selected states at fixed article positions and score the same following 64 targets. Clearing HACANN's memory matrix raises loss by 0.4610 bpb. Clearing prediction, evidence, or filter histories separately changes loss by 0.0031, 0.0026, or 0.0047 bpb. Limiting preceding history to 64 or 256 bytes raises loss by 0.0195 or 0.0025 bpb. After clearing, replaying 1,024 inputs restores loss within 0.000132 bpb of baseline (Figure S4). The associations stored during reading affect subsequent predictions, and renewed experience reconstructs them.

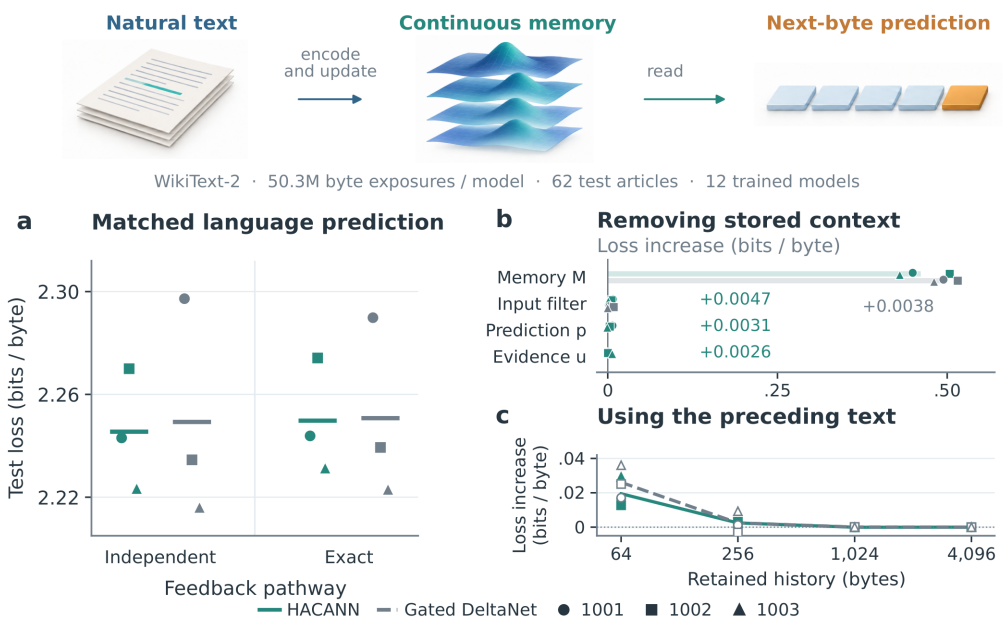

Figure 3: **Natural language prediction uses contextual memory.** (a) Test loss for both recurrent cores and feedback pathways: points show all three seeds and bars show means. (b) Loss increase after clearing individual states; points show seeds and bars show means. (c) Loss increase when only the stated preceding history is available; lines give means and points retain all seeds. Interventions compare HACANN with independent feedback and Gated DeltaNet with exact feedback over the same 1,280 targets at 20 article positions.

5.3 Meaning directs revision and preserves other relations

The Delta rule predicts selective change in a learned cue space. We test this prediction with BANKING77 customer requests (Casanueva et al., 2020). Each intent receives a temporary action code that later instructions can revise. An episode establishes four intent–code relations, queries each, revises one assignment, queries all four again, then revises a second assignment and repeats the queries. Answering requires recognizing intent across phrasings and retaining the current rule. Queries about unchanged intents measure preservation during revision.

One evaluation suite presents new combinations of familiar associations; the other withholds individual intent–code associations during training. We call these unseen mappings and unseen bindings. HACANN reaches 94.43% and 95.12% answer accuracy. Revision accuracy is 93.55% and 95.02%, while protected relations remain correct in 94.66% and 95.15% of queries (Figure 4). All twelve answers are correct in 72.85% and 75.98% of episodes. Matched Gated DeltaNet memories use the same trained HACANN encoder, held fixed, and perform similarly.

Counterfactual changes to action codes, instructions, and query meanings show that answers follow the current relation (Appendix H.3). Increasing the requirement from 12 to 40 answers reduces complete-episode success. The experiment gives cue geometry a behavioral interpretation: meaning identifies which association should change, while the others remain available.

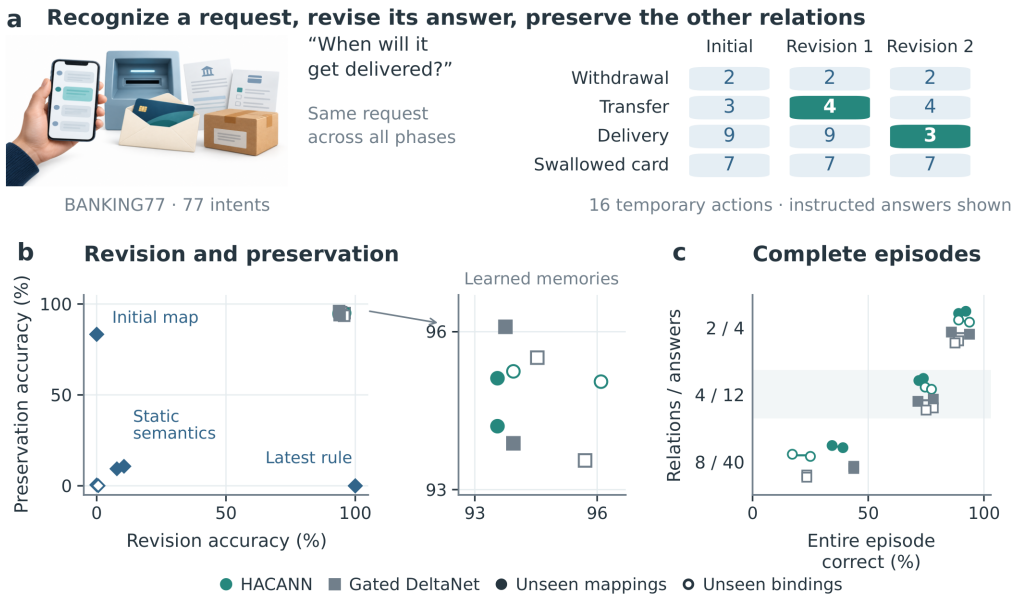

Figure 4: **Semantic revision preserves other relations.** (a) A BANKING77 request and two instructed revisions. (b) Revision and preservation accuracy for associative memories and reference policies; all seed and transfer-suite outcomes are shown. (c) Complete-episode accuracy as the relation and answer load increases. Horizontal spans connect the two observed seeds in each core and suite; the shaded row marks the training load. Filled and open marks denote unseen mappings and unseen bindings, respectively. Training uses four relations.

5.4 Feedback calibration restores representation learning

Selective memory depends on useful representations. The final experiment asks whether activity can restore the pathway that teaches a hierarchy to form them. Two HACANN models first learn eight CIFAR-10 classes from pixels (Krizhevsky, 2009). Each branches into intact feedback, damaged feedback, repaired feedback, and readout-only learning. All branches start with identical forward predictions and then receive the same ten-class exposure. Local input fields expand from four by four to six by six pixels, with new weights initially zero.

Damage rearranges 42 feedback projections while preserving their singular values. Repair uses unlabelled endpoint activity with forward synapses fixed. Calibration reduces median response mismatch from 1.404 and 1.411 to 2.51×10^{-4} and 2.50×10^{-4}. After learning, repaired models reach 80.73% and 80.89% test accuracy, alongside 80.58% and 80.92% for intact models. Damaged models reach 72.96% and 72.09%; readout-only learning reaches 69.06% and 69.09% (Figure 5). Restoring the relation between forward and return activity restores the capacity to acquire useful representations.

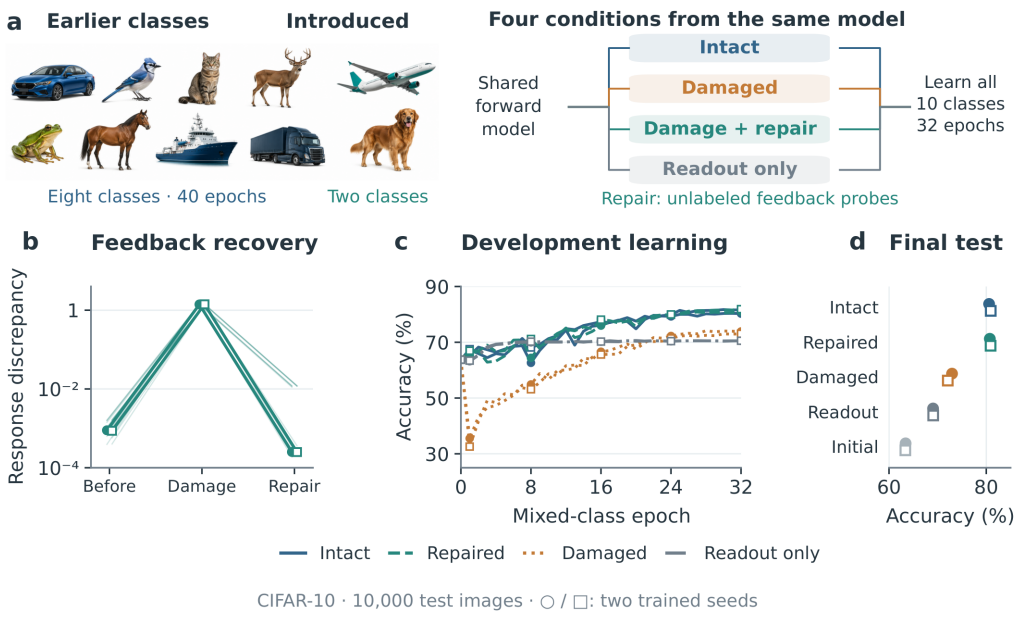

Figure 5: **Feedback repair restores representation learning.** (a) One eight-class model branches into four feedback conditions before shared ten-class learning. Category images are illustrative. (b) Response discrepancy across damage and repair: thin lines show all 42 pathways per seed and heavy lines their medians. (c) Subsequent development accuracy from identical initial predictions. (d) Final accuracy on 10,000 test images, with the initial checkpoint for reference. Markers retain both training seeds. Figures S8–S9 give class and image outcomes.

6 Discussion

HACANN gives the canonical attractor hypothesis a specific computational meaning. A population retains a coordinate; coupled coordinates retain a relation; a hierarchy learns the representations through which relations are addressed. Integrating their response produces a gated Delta computation. The correspondence connects collective neural motion to an associative mechanism used in language models.

The construction distinguishes the geometry that permits memory from the content currently remembered. Recurrent connectivity supplies a family of persistent states. Experience changes the occupied coordinates within that family, thereby changing the relation expressed by the population ensemble. This provides a route from the continuous maps studied in navigation to maps between learned representations (Stachenfeld et al., 2017; Whittington et al., 2020; 2022). The central operation is to preserve a relation while allowing its answer to a particular cue to change.

Prediction error specifies that change. For an isolated Delta correction with a unit key, the update makes the smallest change in memory that produces the prescribed correction of its answer (Appendix I.5). Other answers change according to cue overlap. Synaptic learning therefore shapes cue geometry and how an update generalizes, relating memory revision to the geometry of abstraction (Bernardi et al., 2020). This connects predictive coding's use of unexplained input (Rao & Ballard, 1999) to a concrete mechanism for revising relational memory. Here the error moves population coordinates; longer experience changes the representations used to read them.

Neural accounts of attention include hippocampal relational models, neuron–astrocyte circuits, and Hebbian memory (Whittington et al., 2022; Kozachkov et al., 2023; Ellwood, 2024); spiking language models explore another implementation route (Zhu et al., 2024; Xing et al., 2025). Extending this foundation to large language models entails larger representations and connectivity, controlled population precision, and temporal credit over longer experience. The present theory identifies the neural state, coupling, and timing through which such a hierarchy can retrieve and revise its knowledge.

AI USE STATEMENT

AI tools were used to assist with writing and language polishing, retrieve information and discover related literature, and execute code. The authors take responsibility for the final manuscript and accompanying materials.

REPRODUCIBILITY STATEMENT

The accompanying supplementary material provides experiment code, configurations, numerical data, and figure-generation scripts. Reviewers can begin with `README.md` for environment setup and follow `docs/reproduction.md` for data preparation and experiment-specific reproduction procedures. The package includes both calibrated E1 state archives, utilities for preparing the corresponding compiled states and input windows, and an independent implementation of the finite-response equations. Appendices A–F present the mathematical derivations, while Appendices G–H specify the architecture, parameter settings, and experimental protocols.

REFERENCES

Mohamed Akrout, Collin Wilson, Peter C. Humphreys, Timothy P. Lillicrap, and Douglas B. Tweed. Deep learning without weight transport. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1904.05391.

Shun-ichi Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977. doi: 10.1007/BF00337259.

Nicholas J. Audette and David M. Schneider. Stimulus-specific prediction error neurons in mouse auditory cortex. *Journal of Neuroscience*, 43(43):7119–7129, 2023. doi: 10.1523/JNEUROSCI.0512-23.2023. URL https://www.jneurosci.org/content/43/43/7119.

Jimmy Ba, Geoffrey E. Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/9f44e956e3a2b7b5598c625fcc802c36-Abstract.html.

Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications*, 11:3625, 2020. doi: 10.1038/s41467-020-17236-y. URL https://www.nature.com/articles/s41467-020-17236-y.

Silvia Bernardi, Marcus K. Benna, Mattia Rigotti, Jérôme Munuera, Stefano Fusi, and C. Daniel Salzman. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 183(4):954–967.e21, 2020. doi: 10.1016/j.cell.2020.09.031.

Simon Brandt, Paul Haider, Walter Senn, Federico Benitez, and Mihai A. Petrovici. A variational latent equilibrium for learning in neuronal circuits. arXiv preprint arXiv:2603.09600, 2026. https://arxiv.org/abs/2603.09600.

Trenton Bricken and Cengiz Pehlevan. Attention approximates sparse distributed memory. In *Advances in Neural Information Processing Systems*, volume 34, 2021. https://proceedings.neurips.cc/paper/2021/hash/8171ac2c5544a5cb54ac0f38bf477af4-Abstract.html.

Matteo Carandini and David J. Heeger. Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13(1):51–62, 2012. doi: 10.1038/nrn3136. https://www.cns.nyu.edu/~david/courses/compNeuro/Carandini-NRN2012.pdf.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 38–45, 2020. doi: 10.18653/v1/2020.nlp4convai-1.5. https://aclanthology.org/2020.nlp4convai-1.5/.

Frances S. Chance, L. F. Abbott, and Alex D. Reyes. Gain modulation from background synaptic input. *Neuron*, 35(4):773–782, 2002. doi: 10.1016/S0896-6273(02)00820-6. URL https://pubmed.ncbi.nlm.nih.gov/12194875/.

Rishidev Chaudhuri, Berk Gerçek, Biraj Pandey, Adrien Peyrache, and Ila Fiete. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nature Neuroscience*, 22:1512–1520, 2019. doi: 10.1038/s41593-019-0460-x.

Federico Claudi, Sarthak Chandra, and Ila R. Fiete. A theory and recipe to construct general and biologically plausible integrating continuous attractor neural networks. *eLife Reviewed Preprint*, 107224, 2025. doi: 10.7554/eLife.107224. https://elifesciences.org/reviewed-preprints/107224.

Shaul Druckmann and Dmitri B. Chklovskii. Neuronal circuits underlying persistent representations despite time varying activity. *Current Biology*, 22(22):2095–2103, 2012. doi: 10.1016/j.cub.2012.08.058.

Benjamin Ellenberger, Paul Haider, Federico Benitez, Jakob Jordan, Kevin Max, Ismael Jaras, Laura Kriener, and Mihai A. Petrovici. Backpropagation through space, time and the brain. *Nature Communications*, 17:66, 2026. doi: 10.1038/s41467-025-66666-z. https://arxiv.org/abs/2403.16933v4.

Ian T. Ellwood. Short-term Hebbian learning can implement transformer-like attention. *PLOS Computational Biology*, 20(1):e1011843, 2024. doi: 10.1371/journal.pcbi.1011843. https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1011843.

Yvette E. Fisher, Jenny Lu, Isabel D'Alessandro, and Rachel I. Wilson. Sensorimotor experience remaps visual input to a heading-direction network. *Nature*, 576:121–125, 2019. doi: 10.1038/s41586-019-1772-4.

Yvette E. Fisher, Michael Marquis, Isabel D'Alessandro, and Rachel I. Wilson. Dopamine promotes head direction plasticity during orienting movements. *Nature*, 612:316–322, 2022. doi: 10.1038/s41586-022-05485-4.

David J. Foster and Matthew A. Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440:680–683, 2006. doi: 10.1038/nature04587.

C. C. Alan Fung, K. Y. Michael Wong, and Si Wu. A moving bump in a continuous manifold: A comprehensive study of the tracking dynamics of continuous attractor neural networks. *Neural Computation*, 22(3):752–792, 2010. doi: 10.1162/neco.2009.07-08-824. URL https://arxiv.org/abs/0808.2341.

Richard J. Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A. Baas, Benjamin A. Dunn, May-Britt Moser, and Edvard I. Moser. Toroidal topology of population activity in grid cells. *Nature*, 602:123–128, 2022. doi: 10.1038/s41586-021-04268-7.

Timo Gierlich, Andreas Baumbach, Akos F. Kungl, Kevin Max, and Mihai A. Petrovici. Spike-based alignment learning solves the weight transport problem. *Nature Communications*, 17:8699, 2026. doi: 10.1038/s41467-026-74460-8. URL https://arxiv.org/abs/2503.02642.

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

Paul Haider, Benjamin Ellenberger, Laura Kriener, Jakob Jordan, Walter Senn, and Mihai A. Petrovici. Latent equilibrium: A unified learning theory for arbitrarily fast computation with arbitrarily slow neurons. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/94cdbdb84e8e1de8a725fa2ed61498a4-Abstract.html.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated DeltaNet-2: Decoupling erase and write in linear attention. arXiv preprint arXiv:2605.22791, 2026. URL https://arxiv.org/abs/2605.22791.

David J. Heeger. Normalization of cell responses in cat striate cortex. *Visual Neuroscience*, 9(2):181–197, 1992. doi: 10.1017/S0952523800009640.

Loreen Hertäg and Claudia Clopath. Prediction-error neurons in circuits with multiple neuron types: Formation, refinement, and functional implications. *Proceedings of the National Academy of Sciences*, 119(13):e2115699119, 2022. doi: 10.1073/pnas.2115699119. URL https://www.pnas.org/doi/10.1073/pnas.2115699119.

Kazuki Irie, Francesco Faccio, and Jürgen Schmidhuber. Neural differential equations for learning to program neural nets through continuous learning rules. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/fc09b26b85ab3abb2832bd555a2e4215-Abstract-Conference.html.

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

Sung Soo Kim, Hervé Rouault, Shaul Druckmann, and Vivek Jayaraman. Ring attractor dynamics in the Drosophila central brain. *Science*, 356(6340):849–853, 2017. doi: 10.1126/science.aal4835.

Sung Soo Kim, Ann M. Hermundstad, Sandro Romani, L. F. Abbott, and Vivek Jayaraman. Generation of stable heading representations in diverse visual scenes. *Nature*, 576:126–131, 2019. doi: 10.1038/s41586-019-1767-1.

Kimi Team. Kimi Linear: An expressive, efficient attention architecture. arXiv preprint arXiv:2510.26692, 2025. https://arxiv.org/abs/2510.26692.

Leo Kozachkov, Ksenia V. Kastanenka, and Dmitry Krotov. Building transformers from neurons and astrocytes. *Proceedings of the National Academy of Sciences*, 120(34):e2219150120, 2023. doi: 10.1073/pnas.2219150120. https://pmc.ncbi.nlm.nih.gov/articles/PMC10450673/.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009. https://cave.cs.toronto.edu/kriz/cifar.html.

Yujun Li, Tianhao Chu, and Si Wu. Dynamics of continuous attractor neural networks with spike frequency adaptation. *Neural Computation*, 37(6):1057–1101, 2025. doi: 10.1162/neco_a_01757. URL https://doi.org/10.1162/neco_a_01757.

Timothy P. Lillicrap, Daniel Cownden, Douglas B. Tweed, and Colin J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7:13276, 2016. doi: 10.1038/ncomms13276. URL https://www.nature.com/articles/ncomms13276.

Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503:78–84, 2013. doi: 10.1038/nature12742. URL https://www.nature.com/articles/nature12742.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1609.07843.

Yuanyuan Mi, C. C. Alan Fung, K. Y. Michael Wong, and Si Wu. Spike frequency adaptation implements anticipative tracking in continuous attractor neural networks. In *Advances in Neural Information Processing Systems*, volume 27, 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/57c76ace5d28ad312d72e9d5923fb097-Abstract.html.

Gianluigi Mongillo, Omri Barak, and Misha Tsodyks. Synaptic theory of working memory. *Science*, 319(5869):1543–1546, 2008. doi: 10.1126/science.1150769. URL https://pubmed.ncbi.nlm.nih.gov/18339943/.

John D. Murray, Alberto Bernacchia, Nicholas A. Roy, Christos Constantinidis, Ranulfo Romo, and Xiao-Jing Wang. Stable population coding for working memory coexists with heterogeneous neural dynamics in prefrontal cortex. *Proceedings of the National Academy of Sciences*, 114(2):394–399, 2017. doi: 10.1073/pnas.1619449114.

Marino Pagan, Vincent D. Tang, Mikio C. Aoi, Jonathan W. Pillow, Valerio Mante, David Sussillo, and Carlos D. Brody. Individual variability of neural computations underlying flexible decisions. *Nature*, 639:421–429, 2025. doi: 10.1038/s41586-024-08433-6.

Qwen Team. Qwen3-Next-80B-A3B-Instruct: Model card. Official model documentation, 2025. URL https://huggingface.co/Qwen/Qwen3-Next-80B-A3B-Instruct. Accessed September 21, 2026.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. Extended author version, arXiv:2008.02217v3, 2021. https://arxiv.org/abs/2008.02217v3.

Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, 1999. doi: 10.1038/4580. https://pubmed.ncbi.nlm.nih.gov/10195184/.

João Sacramento, Rui Ponte Costa, Yoshua Bengio, and Walter Senn. Dendritic cortical microcircuits approximate the backpropagation algorithm. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/1dc3a89d0d440ba31729b0ba74b93a33-Abstract.html.

Ábel Ságodi, Guillermo Martín-Sánchez, Piotr Sokół, and Il Memming Park. Back to the continuous attractor. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2136. https://arxiv.org/abs/2408.00109.

Alexei Samsonovich and Bruce L. McNaughton. Path integration and cognitive mapping in a continuous attractor neural network model. *Journal of Neuroscience*, 17(15):5900–5920, 1997. doi: 10.1523/JNEUROSCI.17-15-05900.1997. URL https://www.jneurosci.org/content/17/15/5900.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366, 2021. URL https://proceedings.mlr.press/v139/schlag21a.html.

H. Sebastian Seung. How the brain keeps the eyes still. *Proceedings of the National Academy of Sciences*, 93(23):13339–13344, 1996. doi: 10.1073/pnas.93.23.13339.

Davide Spalla, Isabel Maria Cornacchia, and Alessandro Treves. Continuous attractors for dynamic memories. *eLife*, 10:e69499, 2021. doi: 10.7554/eLife.69499.

Kimberly L. Stachenfeld, Matthew M. Botvinick, and Samuel J. Gershman. The hippocampus as a predictive map. *Nature Neuroscience*, 20(11):1643–1653, 2017. doi: 10.1038/nn.4650.

Daniel Turner-Evans, Stephanie Wegener, Hervé Rouault, Romain Franconville, Tanya Wolff, Johannes D. Seelig, Shaul Druckmann, and Vivek Jayaraman. Angular velocity integration in a fly heading circuit. *eLife*, 6:e23496, 2017. doi: 10.7554/eLife.23496. URL https://elifesciences.org/articles/23496.

Daniel B. Turner-Evans, Kristopher T. Jensen, Saba Ali, Tyler Paterson, Arlo Sheridan, Robert P. Ray, Tanya Wolff, J. Scott Lauritzen, Gerald M. Rubin, Davi D. Bock, and Vivek Jayaraman. The neuroanatomical ultrastructure and function of a biological ring attractor. *Neuron*, 108(1):145–163.e10, 2020. doi: 10.1016/j.neuron.2020.08.006. URL https://pubmed.ncbi.nlm.nih.gov/32916090/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. https://arxiv.org/abs/1706.03762.

Arnau Vivet and Alex Arenas. Emergent self-attention from astrocyte-gated associative memory dynamics. arXiv preprint arXiv:2604.25481, 2026. https://arxiv.org/abs/2604.25481.

James C. R. Whittington, Timothy H. Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy E. J. Behrens. The Tolman-Eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5):1249–1263.e23, 2020. doi: 10.1016/j.cell.2020.10.024.

James C. R. Whittington, Joseph Warren, and Timothy E. J. Behrens. Relating transformers to models and neural representations of the hippocampal formation. In *International Conference on Learning Representations*, 2022. https://arxiv.org/abs/2112.04035.

Nathan R. Wilson, Caroline A. Runyan, Forea L. Wang, and Mriganka Sur. Division and subtraction by distinct cortical inhibitory networks in vivo. *Nature*, 488:343–348, 2012. doi: 10.1038/nature11347. URL https://pubmed.ncbi.nlm.nih.gov/22878717/.

Klaus Wimmer, Duane Q. Nykamp, Christos Constantinidis, and Albert Compte. Bump attractor dynamics in prefrontal cortex explains behavioral precision in spatial working memory. *Nature Neuroscience*, 17(3):431–439, 2014. doi: 10.1038/nn.3645. URL https://pubmed.ncbi.nlm.nih.gov/24487232/.

Si Wu, Kosuke Hamaguchi, and Shun-ichi Amari. Dynamics and computation of continuous attractors. *Neural Computation*, 20(4):994–1025, 2008. doi: 10.1162/neco.2008.10-06-378.

Si Wu, K. Y. Michael Wong, C. C. Alan Fung, Yuanyuan Mi, and Wenhao Zhang. Continuous attractor neural networks: Candidate of a canonical model for neural information representation. *F1000Research*, 5:156, 2016. doi: 10.12688/f1000research.7387.1.

Xingrun Xing, Boyan Gao, Zheng Liu, David A. Clifton, Shitao Xiao, Wanpeng Zhang, Li Du, Zheng Zhang, Guoqi Li, and Jiajun Zhang. SpikeLLM: Scaling up spiking neural network to large language models via saliency-based spiking. In *International Conference on Learning Representations*, 2025. https://proceedings.iclr.cc/paper_files/paper/2025/hash/510e7d39fce008a3e31de54b8f5be9ac-Abstract-Conference.html.

Guangyu Robert Yang, John D. Murray, and Xiao-Jing Wang. A dendritic disinhibitory circuit mechanism for pathway-specific gating. *Nature Communications*, 7:12815, 2016. doi: 10.1038/ncomms12815.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems*, volume 37, pp. 115491–115522, 2024. doi: 10.52202/079017-3668. URL https://arxiv.org/abs/2406.06484.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.06464.

Kechen Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory. *Journal of Neuroscience*, 16(6):2112–2126, 1996. doi: 10.1523/JNEUROSCI.16-06-02112.1996. URL https://www.jneurosci.org/content/16/6/2112.

Cheng Zhang, Minnan Luo, Zesheng Yang, Ming Li, Yong-Jin Liu, and Qinghua Zheng. Controlled dynamics attractor transformer. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. https://arxiv.org/abs/2606.15207.

Rui-Jie Zhu, Qihang Zhao, Guoqi Li, and Jason K. Eshraghian. SpikeGPT: Generative pre-trained language model with spiking neural networks. *Transactions on Machine Learning Research*, 2024. https://openreview.net/forum?id=gcf1anBL9e.

APPENDICES

The theory follows the computation from a neural field to a changing association. The architecture then specifies how these associations form a complete network. Experimental protocols, outcomes, and neural predictions follow. Every entry links to its section; page numbers also link to their destinations.

THEORY AND ARCHITECTURE

Derivations	**Page**
Notation and route through the derivation	17
A. The population coordinate and its spatial dynamics	18
A.1. Adaptive neural field and stationary pattern	18
A.2. Joint translation mode and memory readout	19
A.3. Exact residual under spatial deformation	20
A.4. Two positive actuator populations	21
B. The finite neural response and its Delta recurrence	22
B.2. Solution over one input interval	23
B.3. Explicit response coefficients	23
B.4. Exact augmented Delta form	25
B.5. Persistent evidence and canonical Gated DeltaNet	25
B.6. Conditional dissipation	26
C. Credit dynamics and local plasticity	27
C.2. Exact discrete credit for the finite event	27
C.4. Reciprocity learned from activity responses	29
D. Parallel evaluation through causal elimination	30
E. Reconstructing past states from their updates	31
F. Finite input formation and transmission through layers	32
Complete model	
G. Architecture, representations, and synaptic learning	34
G.1. Forward equations for the complete hierarchy	34
G.2. Rate circuits for every processing operation	36
G.3. Information flow, dimensions, and parameter counts	37
G.4. State, supervision, and temporal credit	38
G.5. Synaptic development and learning	39

EXPERIMENTS AND NEURAL INTERPRETATION

Protocols and observations **Page**
H. Experimental methods and complete results 40
H.1. Spatial populations, learning, and coordinate interventions 41
Language prediction under correlated population current 44
H.2. Language prediction and document context 45
H.3. Semantic grounding and selective revision 47
H.4. Visual learning and restoration of feedback 52
H.5. Measured quantities and figure construction 55

Neural interpretation
I. Neural interpretation and experimental predictions 55
Canonical modules and the maintenance of relations 55
Circuit operations and biological realization 56
Reading memory during changing activity 57
Prediction error and relational memory 58
Cue geometry and the scope of a correction 59
Response times and continuing evidence 60
Persistence, precision, and stability 61
Reciprocal activity and hierarchical learning 61
Connections to neural and artificial memory models 62

Supplementary figures
J. Supplementary figures 64
S1. Population range, learning, residuals, and noise 64
S2. Coordinate trajectories and predictive consequences 65
S3. Language validation and test outcomes 66
S4. Language history, articles, and computational cost 67
S5. Semantic revision and protection references 68
S6. Semantic counterfactual outcomes 69
S7. Visual learning trajectories 70
S8. Visual outcomes for every class 71
S9. Retained and acquired visual knowledge 72

NOTATION AND ROUTE THROUGH THE DERIVATION

The derivation follows the physical computation. A recurrent population retains a coordinate. Connections between populations turn these coordinates into a mapping from cues to content. The response to one input then gives a finite memory update. The first two tables collect the population and computational notation; each quantity is introduced again where it enters the derivation. Symbols used only for a local calculation are defined in that calculation.

Population quantity	Meaning
x, t	Preferred coordinate within a population and physical time.
U, V, g, r	Synaptic activity, adaptation, filtered input, and firing rate.
J, κ, m	Recurrent excitation kernel, divisive inhibition strength, and adaptation strength.
τ_U, τ_V, τ_g	Response times of activity, adaptation, and input.
γ	$\tau_U - m\tau_V > 0$; sets translation gain and adaptation relaxation.
a, J_0, A_0	Recurrent kernel width, integrated excitation strength, and stationary bump amplitude.
δ, c_δ	Pathway displacement and its response normalization, $c_\delta = e^{-\delta^2/(6a^2)}$.
$T_\delta[U]$	Spatial drive formed by oppositely shifted recurrent projections.
G_+, G_-	Nonnegative actuator populations whose difference supplies g.
$Q, \psi = DQ$	Activity center and its sensitivity to a spatial perturbation.
Z, F, R_Z	Joint memory coordinate, driving force, and residual in the spatial motion law $\dot{Z} = F + R_Z$.

Computation quantity	Meaning
d_k, d_v	Numbers of key and value components in one head.
k, q, v, o	Current key, query, value, and retrieved output.
M, p, u	Relation matrix, prediction trace, and evidence trace.
λ, ρ, θ	Memory turnover rate, acquisition rate, and relative prediction response time.
Δ, n	Input duration and squared key norm $n = k^\top k$. The widths of a dense projection use locally defined n, m in Appendix C.
α, β	Retention $e^{-\lambda\Delta}$ and acquisition fraction $1 - e^{-\rho\Delta}$; β is the correction fraction for an equilibrated prediction and unit key.
A, E	Five dimensional event generator and its response $E = e^{\Delta A}$.
$a, b, c, d; e, f, g, h$	Response coefficients for the memory write and continuing prediction; given explicitly in Appendix B.
μ_+, μ_-	The two eigenvalues of the coupled memory and prediction response.
W, B	Learned forward projection and independently stored feedback projection.
e, s	Local eligibility trace and squared plasticity-drive trace.
P, a_p, a_u	Derivatives of the retained loss with respect to memory, prediction, and evidence; the continuous credit variables.

Appendix A derives the population coordinate and its spatial residual. Appendix B gives the complete finite response, explicit coefficients, and Gated DeltaNet regime. Appendix C derives temporal credit and local plasticity. Appendices D–F give parallel evaluation, replay, finite key formation, and layer timing. Appendices G and H specify the hierarchy and every experiment. Appendix I develops the neural interpretation and testable predictions.

A THE POPULATION COORDINATE AND ITS SPATIAL DYNAMICS

A continuous attractor retains a value in the position of a stable activity pattern. Its recurrent connections restore the pattern's shape after a disturbance, while allowing its position to change. Adaptation and filtered input introduce a further distinction: the visible position can continue to move after the external input has ended. We derive the coordinate that is conserved as these transients relax, then calculate how an applied force changes it. Coupling such coordinates gives the associative memory studied in the main text.

The derivation has two levels. Linear projection onto the translation mode gives the scalar law used to construct the associative circuit. An exact identity for the spatial field then expresses the deviation from that law in terms of observable population quantities. The finite response in Appendix B integrates the projected circuit over an input interval.

A.1 THE ADAPTIVE NEURAL FIELD

Each scalar memory uses the same recurrent population model. Let x index neurons by preferred coordinate and let t denote time. The fields $U(x,t)$, $V(x,t)$, and $g(x,t)$ represent synaptic activity, adaptation, and filtered input; $r(x,t)$ is the firing rate. A dot denotes differentiation with respect to t. The nonnegative parameters m and κ set adaptation and divisive inhibition. The response times τ_U, τ_V, τ_g are positive, with $\gamma = \tau_U - m\tau_V > 0$. For a translation invariant excitatory kernel J, write $(J * r)(x) = \int J(x - x')r(x')\,\mathrm{d}x'$. Write $[U]_+ = \max(U, 0)$ for rectified activity. The field equations are

$$\tau_U \dot{U} = -U + J * r - V + \gamma g, \qquad \tau_V \dot{V} = -V + mU, \tag{9}$$
$$r(x) = \frac{[U(x)]_+^2}{1 + \kappa \int [U(x')]_+^2\,\mathrm{d}x'}, \qquad \gamma = \tau_U - m\tau_V > 0.$$

The kernel supports a stable family of localized activity patterns, as in the Wu continuous attractor construction (Fung et al., 2010; Mi et al., 2014; Li et al., 2025). The task uses a common kernel across memory coordinates. Its parameters set the population shape and response, while the connections between populations set the task computation.

Let $a > 0$ be the spatial width of the recurrent kernel, $J_0 > 0$ its integrated strength, $A_0 > 0$ the bump amplitude, and z the bump center. For a Gaussian kernel and a positive stationary bump, write

$$J(x) = \frac{J_0}{\sqrt{2\pi}a} e^{-x^2/(2a^2)}, \qquad U_0(x - z) = A_0 e^{-(x-z)^2/(4a^2)}, \qquad V_0 = mU_0. \tag{10}$$

Let r_0 be the stationary firing profile and let $D_0 = 1 + \kappa A_0^2 a\sqrt{2\pi}$ be its normalization denominator. Convolution of the two Gaussians gives

$$J * r_0 = \frac{J_0 A_0^2}{\sqrt{2} D_0} e^{-(x-z)^2/(4a^2)}, \qquad D_0 = 1 + \kappa A_0^2 a\sqrt{2\pi}.$$

Thus the stationary amplitude satisfies $J_0 A_0 = \sqrt{2}(1 + m)D_0$, independently of the center z. The field implementation uses $\kappa = 2/(a\sqrt{2\pi})$ and $J_0 = 3\sqrt{2}(1 + m)$, giving the initialized solution $A_0 = 1$. Translation leaves the recurrent balance unchanged. This symmetry supplies the continuously valued memory coordinate. The following projection assumes that the other shape modes relax over the chosen operating range.

The coordinate chart is chosen so that the activity remains localized away from its boundary. A ring supports an unwrapped local chart over the operating range. A finite chart consequently represents a bounded interval of the logical memory value, with a physical scale fixed for the implementation. The Gaussian identities refer to the continuous field; the simulations approximate this field on a periodic spatial grid.

The activity center Q is a weighted mean of preferred coordinates. The cubic weighting used below is chosen so that its sensitivity agrees with the left translation mode of the Gaussian population; the next subsection derives that agreement. Its normalization is $N[U] = \int [U]_+^3\,\mathrm{d}x$, assumed positive. The function ψ describes how a local activity change moves this center. Both are given by

$$Q[U] = \frac{\int x[U(x)]_+^3\,\mathrm{d}x}{N[U]}, \qquad \psi(x) = \frac{3[U(x)]_+^2(x - Q)}{N[U]}. \tag{11}$$

For a spatial perturbation h, let $DQ[U]h$ denote the directional derivative of Q at U, and let $\langle f, h\rangle = \int f(x)h(x)\,\mathrm{d}x$ denote the spatial inner product. Differentiating the numerator and denominator gives

$$DQ[U]h = \frac{3\int x[U]_+^2 h}{N} - \frac{3Q\int [U]_+^2 h}{N} = \langle \psi, h\rangle.$$

Thus $\dot{Q} = \langle \psi, \dot{U}\rangle$. Since the divisive rate normalization cancels between numerator and denominator, $Q = \int xr^{3/2}/\int r^{3/2}$. The readout uses local activity transformations, fixed spatial weights, and a normalization pool. The changing readout sensitivity can also be evaluated directly. With $R = [U]_+$ and $\dot{R} = \mathbf{1}_{U>0}\dot{U}$ away from threshold,

$$\dot{N} = 3\int R^2\dot{R}\,\mathrm{d}x, \qquad \dot{\psi} = \frac{6R\dot{R}(x-Q) - 3R^2\dot{Q}}{N} - \psi\frac{\dot{N}}{N}.$$

The cubic rate functional is differentiable at threshold, so this expression extends through threshold crossings wherever the trajectory derivative exists.

A signed scalar force $F(t)$ sets the requested rate of memory change. The spatial profile $T_\delta[U]$ converts this force into a push on the two sides of the activity bump; $\delta > 0$ is a fixed pathway displacement. The actuator has response time τ_g:

$$\tau_g\dot{g} = FT_\delta[U] - g. \tag{12}$$

The force F is formed from the currently read memory coordinate. The spatial operator T_δ is constructed from shifted recurrent projections below. With adaptation displacement $A_{\mathrm{ad}} = V - mU$, define

$$Z = Q + \left\langle \psi, \tau_g g - \frac{\tau_V}{\gamma}A_{\mathrm{ad}}\right\rangle. \tag{13}$$

At a stationary bump, $A_{\mathrm{ad}} = 0$ and $g = 0$, so $Z = Q$. During a response, adaptation and actuator activity carry part of the displacement represented by Z.

A.2 The joint translation mode

The power in the center readout follows from the left translation mode. Consider a symmetric recurrent kernel and the even Gaussian stationary bump $U_0(x-z) = A_0\exp[-(x-z)^2/(4a^2)]$, with adaptation mU_0. Put $D_0 = 1 + \kappa\int U_0^2$. Differentiating the stationary equation with respect to x gives

$$J * (U_0^2/D_0) = (1+m)U_0, \qquad \frac{2}{D_0}J * (U_0\partial_x U_0) = (1+m)\partial_x U_0.$$

On odd perturbations, the linearized stationary field operator is $\mathcal{L}h = (2/D_0)J*(U_0h) - (1+m)h$. For a test function ϕ, kernel symmetry gives the adjoint action $\mathcal{L}^*\phi = (2U_0/D_0)J*\phi - (1+m)\phi$, and therefore $\mathcal{L}^*(U_0\partial_x U_0) = 0$. The derivative of the common normalization pool is zero on odd perturbations. Because $\partial_x U_0 = -(x-z)U_0/(2a^2)$, the function $\psi_0 = DQ[U_0]$ is this left mode up to normalization. Translation covariance, $Q[U_0(\cdot - z)] = z$, fixes the normalization: $DQ[U_0][-\partial_x U_0] = 1$.

Adaptation and actuator dynamics determine how this mode extends to the full state. Linearizing Equation (13) about $(U_0, mU_0, 0)$ gives

$$\delta Z = \frac{\tau_U}{\gamma}\langle\psi_0, \delta U\rangle - \frac{\tau_V}{\gamma}\langle\psi_0, \delta V\rangle + \tau_g\langle\psi_0, \delta g\rangle. \tag{14}$$

The coefficients follow from the physical time constants. With $F = 0$, the linearized equations conserve this coordinate. On a translated stationary state, it equals the displacement. With normalized tangent drive, its projected motion is $\dot{Z} = F$. An activity profile can consequently continue to move while its joint memory coordinate remains fixed.

The condition $\gamma > 0$ also has a direct dynamical meaning. Let a_U, a_V, a_g denote the amplitudes of their common translation profile, and write the perturbations of U, V, g as $a_U(-\partial_x U_0)$, $a_V(-\partial_x U_0)$, and $a_g(-\partial_x U_0)$. Their unforced equations are

$$\tau_U\dot{a}_U = ma_U - a_V + \gamma a_g, \qquad \tau_V\dot{a}_V = ma_U - a_V, \qquad \tau_g\dot{a}_g = -a_g.$$

The activity and adaptation subsystem has eigenvalues 0 and $-\gamma/(\tau_U\tau_V)$. Thus the relative displacement between activity and adaptation relaxes while their joint coordinate persists. The actuator adds its own relaxing mode with eigenvalue $-1/\tau_g$.

These equations also give a direct interpretation of the coordinate. With the force closed, their conserved quantity is

$$z_{\text{joint}} = a_U - \frac{\tau_V}{\gamma}(a_V - m a_U) + \tau_g a_g. \tag{15}$$

After the relaxing modes have decayed, $a_g = 0$ and $a_V = m a_U$, so $a_U(\infty) = z_{\text{joint}}$. The adaptation and actuator terms therefore measure the displacement still to be expressed in the activity center. Equation (13) extends this translation coordinate to the current field by evaluating its sensitivity at the observed activity profile. The next identity gives the resulting motion when that profile deforms.

A.3 The exact residual in a deforming field

Equation (13) evaluates the coordinate using the differential of the current population state. Its evolution in the full spatial field has the following identity.

Proposition 1 (Motion of the joint coordinate)**.** *For a differentiable trajectory of Equations* (9) *and* (12) *within the coordinate chart,*

$$\begin{aligned}\dot{Z} = F\langle \psi, T_\delta[U]\rangle + \frac{\langle \psi, J * r - (1+m)U\rangle}{\gamma} \\ + \left\langle \dot{\psi}, \tau_g g - \frac{\tau_V}{\gamma} A_{\text{ad}} \right\rangle .\end{aligned} \tag{16}$$

Proof. Set $H = J * r - (1+m)U$. The field equations give $\tau_U \dot{U} = H - A_{\text{ad}} + \gamma g$ and $\dot{A}_{\text{ad}} = -A_{\text{ad}}/\tau_V - m\dot{U}$. Differentiating Equation (13) yields the term containing $\dot{\psi}$, together with

$$\left\langle \psi, \left(1 + \frac{m\tau_V}{\gamma}\right)\dot{U} + \frac{A_{\text{ad}}}{\gamma} - g + F T_\delta[U] \right\rangle .$$

The equality $1 + m\tau_V/\gamma = \tau_U/\gamma$ lets the field equation cancel the adaptation and actuator terms, leaving $H/\gamma + F T_\delta[U]$. □

Subtracting F from Equation (16) gives

$$\begin{aligned}\dot{Z} &= F + R_Z, \\ R_Z &= F(\langle \psi, T_\delta[U]\rangle - 1) + \frac{\langle \psi, H\rangle}{\gamma} + \left\langle \dot{\psi}, \tau_g g - \frac{\tau_V}{\gamma} A_{\text{ad}} \right\rangle .\end{aligned} \tag{17}$$

The three terms describe a changed response to the actuator, a force from shape deformation, and a change in the sensitivity of the readout. On the stationary bump family, $H = 0$ and the tangent gain is one. Retaining the translation mode together with its adaptation and actuator components gives the projected law $\dot{Z} = F$. The residual identity holds along the full spatial trajectory and supplies a direct test of this reduction. The field simulations calculate the force from their own physical readout and evaluate all three residuals along the same trajectory.

Accuracy of the translation reduction. The residual identity also states how the projection controls its error. Use $\|\cdot\|_2$ for the spatial L^2 norm, and let $H = J * r - (1+m)U$ be the force from the recurrent field. The Cauchy–Schwarz inequality gives

$$\begin{aligned}|R_Z| \le |F|\,|\langle \psi, T_\delta[U]\rangle - 1| + \frac{\|\psi\|_2 \|H\|_2}{\gamma} \\ + \|\dot{\psi}\|_2 \left(\tau_g \|g\|_2 + \frac{\tau_V}{\gamma}\|V - mU\|_2\right).\end{aligned} \tag{18}$$

For an externally prescribed force and matched initial coordinates, integration bounds the displacement error by $\int_0^t |R_Z(s)|\,\mathrm{d}s$. In the coupled circuit, the force depends on the readout. Collect all

projected coordinates in a vector z, denote their common force function by $f(z,t)$, and collect the spatial residuals in $R(t)$. The full readout obeys $\dot{z} = f(z,t) + R(t)$. If f is Lipschitz in z with constant L over the trajectory, its difference from the projected solution z_{proj} satisfies

$$\|z(t) - z_{\mathrm{proj}}(t)\| \le e^{Lt}\|z(0) - z_{\mathrm{proj}}(0)\| + \int_0^t e^{L(t-s)}\|R(s)\|\,\mathrm{d}s. \tag{19}$$

For the fixed input associative flow, the force is linear in the carried state, so a finite constant L follows from the gains and key norm. The measured spatial residuals thus describe how the field realization departs from its translation dynamics over time. The finite response derived below solves those projected dynamics exactly for the held input schedule.

A.4 Two positive actuator populations

The two input pathways excite recurrent projections shifted by the same distance δ in opposite directions. Denote their profiles by L and R, and use the positive normalization $c_\delta = e^{-\delta^2/(6a^2)}$. Then

$$L = J(x-\delta) * r, \qquad R = J(x+\delta) * r, \qquad T_\delta[U] = \frac{L-R}{2\delta(1+m)c_\delta}. \tag{20}$$

Both projections have nonnegative kernels. On the Gaussian stationary family, $L = (1+m)U_0(x - z - \delta)$ and $R = (1+m)U_0(x - z + \delta)$. Completing the square in their pairing with ψ_0 gives

$$\langle \psi_0, U_0(\cdot - z - \delta)\rangle = \delta e^{-\delta^2/(6a^2)}, \qquad \langle \psi_0, U_0(\cdot - z + \delta)\rangle = -\delta e^{-\delta^2/(6a^2)}.$$

Thus $c_\delta = e^{-\delta^2/(6a^2)}$ sets $\langle \psi_0, T_\delta[U_0]\rangle = 1$. The gain depends on population width and pathway displacement and is shared by every memory module.

Represent the signed force by the nonnegative channels $F_+ = \max(F,0)$ and $F_- = \max(-F,0)$, so that $F = F_+ - F_-$. Two nonnegative actuator populations obey

$$\tau_g \dot{G}_+ = F_+ L + F_- R - G_+, \tag{21}$$

$$\tau_g \dot{G}_- = F_+ R + F_- L - G_-. \tag{22}$$

Their difference $g = (G_+ - G_-)/[2\delta(1+m)c_\delta]$ obeys Equation (12). Fixed excitatory and inhibitory output pathways implement this signed drive. Multiplication by the nonnegative force channels is a dendritic or conductance gain abstraction, consistent with the rate circuit description.

A.5 Accessible readouts and population costs

The joint readout requires access to adaptation. If V is the activity of a local adaptive inhibitory population, its output can reach both the field and the readout through fixed pathways. If adaptation is intracellular, an accessible trace provides the corresponding signal. One such realization uses the causal center trace

$$\tau_V \dot{y} = Q - y, \qquad Z_{\mathrm{tr}} = \frac{\tau_U Q - m\tau_V y}{\gamma} + \tau_g \langle \psi, g\rangle. \tag{23}$$

To derive this realization in the translation projection, put $a_V = my$. The tangent adaptation equation then gives $\tau_V \dot{y} = a_U - y$. Substitution in Equation (15) yields Equation (23), with Q supplying the center a_U. The trace is initialized consistently with the adaptation displacement; its contribution vanishes when $m = 0$. Its spatial approximation error is evaluated for the trace realization; Equation (16) gives the exact identity for the direct readout Z. Both realizations were integrated in the field experiments.

Let d_k and d_v be the numbers of key and value components. A head stores one coordinate Z_{ij} for each pair, with $i = 1,\dots,d_v$ and $j = 1,\dots,d_k$. The current key component k_j selects the column; the current value v_i, prediction p_i, and evidence u_i supply row signals. The nonnegative gains λ and ρ control memory turnover and acquisition. The force at (i,j) is

$$F_{ij} = -\lambda Z_{ij} + \big[\lambda u_i + \rho(v_i - p_i)\big]k_j. \tag{24}$$

A module receives its own coordinate, a row signal, a column signal, and the two gains. Prediction and output use row sums of $Z_{ij}k_j$ and $Z_{ij}q_j$. These connections require $O(d_v d_k)$ contacts,

multiplied by the fixed population cost of a scalar module. Each module also contains its recurrent kernel and shifted projections. Learned interlayer projections and feedback pathways contribute $O(P)$ contacts for P learned coefficients.

Opponent channels represent signed activities and weights with fixed excitatory and inhibitory pathway signs. Their cells and contacts are part of the physical realization. Rates, effective weights, and multiplicative gains operate over a bounded working range. Static normalization and input functions are rate transfer abstractions. Memory, prediction, evidence, adaptation, and actuators have explicit dynamics. The complete spatial experiment integrates the memory and both trace populations across space and evaluates the peripheral transformations as rate operations.

For the trained representation field experiment, physical values use $v_{\mathrm{phys}} = v_{\mathrm{soft}}/16$ and $M_{\mathrm{soft}} = 16Z$. This linear change of units preserves the projected flow and fits the measured trajectory within the chart. A physical readout perturbation is multiplied by 16 in software units, giving a factor 256 in its variance. The physical range and readout precision are therefore linked by the chosen scale.

B THE FINITE NEURAL RESPONSE AND ITS DELTA RECURRENCE

B.1 THE GEOMETRY OF A POPULATION UPDATE

The projected memory consists of a relation matrix $M \in \mathbb{R}^{d_v \times d_k}$, a prediction trace $p \in \mathbb{R}^{d_v}$, and an evidence trace $u \in \mathbb{R}^{d_v}$. A key $k \in \mathbb{R}^{d_k}$ addresses the relation, a value $v \in \mathbb{R}^{d_v}$ supplies current content, and a query $q \in \mathbb{R}^{d_k}$ reads the output $o \in \mathbb{R}^{d_v}$. Let $\theta > 0$ be the prediction response time relative to the evidence response time. The three populations obey

$$\begin{aligned} \dot{M} &= -\lambda(M - uk^\top) + \rho(v - p)k^\top, \\ \theta\dot{p} &= \rho(Mk - p), \qquad \dot{u} = \rho(v - u), \qquad o = Mq. \end{aligned} \tag{25}$$

The acquisition rate ρ controls correction, prediction, and evidence integration. The positive dimensionless constant θ sets the relative prediction response time. Memory population (i, j) receives the force in Equation (24). The forces on the prediction and evidence populations are, respectively,

$$F_i^p = \frac{\rho}{\theta}\left(\sum_j M_{ij}k_j - p_i\right), \qquad F_i^u = \rho(v_i - u_i).$$

Applying the scalar projected law to these three groups of populations gives Equation (25).

The key determines which matrix direction receives the write. For any fixed query q, the output therefore changes according to

$$\frac{\mathrm{d}}{\mathrm{d}t}(Mq) = -\lambda Mq + [\lambda u + \rho(v - p)](k^\top q). \tag{26}$$

The same content signal reaches every query in proportion to its overlap with the current key. An orthogonal query undergoes only retention decay. Learned keys thus determine how a local correction transfers to other inputs. This relation follows directly from the row and column connections of the populations.

There is also a direct relation to prediction error minimization. Hold k and v fixed and define the current association loss as $\mathcal{L}_k(M) = \frac{1}{2}\|v - Mk\|^2$. Its matrix gradient is

$$\nabla_M \mathcal{L}_k = (Mk - v)k^\top. \tag{27}$$

When prediction has equilibrated, the correction term is $-\rho\nabla_M\mathcal{L}_k$. The same cue is used to read the prediction and distribute its error. This reuse determines both which relation changes and how the change transfers to related cues. The finite prediction population retains the time taken to form the retrieved answer. The evidence pathway supplies an additional retained value toward which memory turns over. These two traces give the complete dynamics in Equation (25) their dependence on recent experience.

At $\rho = 0$, prediction and evidence retain their coordinates in their attractor populations. The memory can still relax toward the retained association $uk^\top$ through the pathway controlled by λ. With both gains zero, all three represented states persist.

B.2 SOLVING ONE HELD INPUT INTERVAL

One event is an input interval during which k, v, λ, ρ remain fixed for duration $\Delta \geq 0$. Its terminal states M_+, p_+, u_+ become the incoming states of the following event. Write M_0, p_0, u_0 for the incoming states and use $s \in [0, \Delta]$ for elapsed time. Because every write has right factor $k^\top$, the matrix response remains within one key direction plus its decaying incoming state. Variation of constants gives

$$M_+ = e^{-\lambda\Delta} M_0 + \left[\int_0^\Delta e^{-\lambda(\Delta - s)} \{\lambda u(s) + \rho[v - p(s)]\} \, \mathrm{d}s \right] k^\top. \tag{28}$$

Each contribution is the content signal present at time s, attenuated by the retention decay between s and the end of the event. Solving the coupled traces evaluates this integral in closed form.

Let $n = k^\top k$ be the squared key norm. Within this response calculation, let $r(s)$ denote the accumulated write vector and let $x(s)$ denote the decaying incoming memory read. Then

$$M(s) = e^{-\lambda s} M_0 + r(s) k^\top, \qquad x(s) = e^{-\lambda s} M_0 k. \tag{29}$$

For each value row, the five variables (r, p, u, v, x) obey the common linear generator

$$A = \begin{pmatrix} -\lambda & -\rho & \lambda & \rho & 0 \\ n\rho/\theta & -\rho/\theta & 0 & 0 & \rho/\theta \\ 0 & 0 & -\rho & \rho & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\lambda \end{pmatrix}, \qquad \begin{pmatrix} r \\ p \\ u \\ v \\ x \end{pmatrix}_{s=0} = \begin{pmatrix} 0 \\ p_0 \\ u_0 \\ v \\ M_0 k \end{pmatrix}. \tag{30}$$

For example, substitution of Equation (29) into the memory and prediction equations gives

$$\dot{r} = -\lambda r - \rho p + \lambda u + \rho v, \qquad \theta \dot{p} = \rho(x + nr - p), \qquad \dot{x} = -\lambda x.$$

Together with the evidence equation and $\dot{v} = 0$, these are the rows of A. Each value row follows this same five dimensional system. Its auxiliary coordinates r, x describe the response within the current interval; the persistent state remains M, p, u. The fifth coordinate supplies the decaying contribution of the incoming matrix, and the fourth holds the value constant.

Let $E = e^{\Delta A}$ be the response matrix over the event, using indices $0, \ldots, 4$ in the order (r, p, u, v, x). The retention factors for memory and evidence are $\alpha = e^{-\lambda\Delta}$ and $j = e^{-\rho\Delta}$. Four coefficients (a, b, c, d) describe how the incoming memory read, prediction, evidence, and current value contribute to the write. The coefficients (e, f, g, h) give their contributions to the terminal prediction. These are entries of the same response matrix:

$$\begin{aligned} (a, b, c, d) &= (E_{04}, E_{01}, E_{02}, E_{03}), \\ (e, f, g, h) &= (E_{14}, E_{11}, E_{12}, E_{13}), \qquad\qquad \alpha = e^{-\lambda\Delta}, \quad j = e^{-\rho\Delta}. \end{aligned} \tag{31}$$

With $r_* = r(\Delta)$ denoting the terminal write vector, the exact event response is

$$\begin{aligned} r_* &= aM_0 k + bp_0 + cu_0 + dv, \\ M_+ &= \alpha M_0 + r_* k^\top, \\ p_+ &= eM_0 k + fp_0 + gu_0 + hv, \qquad u_+ = ju_0 + (1 - j)v. \end{aligned} \tag{32}$$

The exponential depends continuously on the key norm, including $n = 0$. This covers keys normalized with a finite denominator constant as well as general nonunit keys. For $k = 0$, the matrix decays, $p_+ = e^{-\rho\Delta/\theta} p_0$, and the evidence follows its displayed response. For $\rho = 0$, the two traces persist and the matrix relaxes toward $u_0 k^\top$, giving $M_+ = \alpha M_0 + (1 - \alpha) u_0 k^\top$. These cases also follow directly from the neural flow.

B.3 EXPLICIT RESPONSE COEFFICIENTS

The matrix exponential above supplies a compact numerical evaluation. To expose its dependence on the circuit parameters, we now evaluate each coefficient through the two coupled memory and

prediction modes. For each value row, the variables r and p in Equation (29) have homogeneous generator

$$\mathsf{A}_{rp} = \begin{pmatrix} -\lambda & -\rho \\ n\rho/\theta & -\rho/\theta \end{pmatrix}. \tag{33}$$

The two eigenvalues of this response generator are

$$\mu_{\pm} = -\frac{\lambda + \rho/\theta}{2} \pm \sqrt{\frac{(\lambda - \rho/\theta)^2}{4} - \frac{n\rho^2}{\theta}}. \tag{34}$$

For distinct eigenvalues, let $E^{rp}(s) = e^{s\mathsf{A}_{rp}}$ denote the homogeneous response over elapsed time s, with row and column indices $1, 2$ corresponding to r, p. Its four entries are

$$\begin{aligned} E^{rp}_{11}(s) &= \frac{(\mu_+ + \rho/\theta)e^{\mu_+ s} - (\mu_- + \rho/\theta)e^{\mu_- s}}{\mu_+ - \mu_-}, \\ E^{rp}_{12}(s) &= -\rho\frac{e^{\mu_+ s} - e^{\mu_- s}}{\mu_+ - \mu_-}, \\ E^{rp}_{21}(s) &= \frac{n\rho}{\theta}\frac{e^{\mu_+ s} - e^{\mu_- s}}{\mu_+ - \mu_-}, \\ E^{rp}_{22}(s) &= \frac{(\mu_+ + \lambda)e^{\mu_+ s} - (\mu_- + \lambda)e^{\mu_- s}}{\mu_+ - \mu_-}. \end{aligned} \tag{35}$$

The forcing terms are already explicit: $u(s) = v + e^{-\rho s}(u_0 - v)$ and $x(s) = e^{-\lambda s}M_0 k$. The response therefore requires only the integral of one exponential against another. For a decay rate $c \geq 0$ and an eigenvalue μ, define

$$\mathcal{I}_c(\mu) = \int_0^{\Delta} e^{\mu(\Delta - s)}e^{-cs}\,\mathrm{d}s = \begin{cases} \dfrac{e^{\mu\Delta} - e^{-c\Delta}}{\mu + c}, & \mu + c \neq 0, \\ \Delta e^{-c\Delta}, & \mu + c = 0. \end{cases} \tag{36}$$

The duration Δ is fixed within this notation. Substituting the forcing into the first row of Equation (35) gives all four write coefficients from Equation (31):

$$\begin{aligned} a &= -\frac{\rho^2}{\theta}\frac{\mathcal{I}_\lambda(\mu_+) - \mathcal{I}_\lambda(\mu_-)}{\mu_+ - \mu_-}, \\ b &= -\rho\frac{e^{\mu_+\Delta} - e^{\mu_-\Delta}}{\mu_+ - \mu_-}, \\ c &= \lambda\frac{(\mu_+ + \rho/\theta)\mathcal{I}_\rho(\mu_+) - (\mu_- + \rho/\theta)\mathcal{I}_\rho(\mu_-)}{\mu_+ - \mu_-}, \\ d &= (\lambda + \rho)\frac{(\mu_+ + \rho/\theta)\mathcal{I}_0(\mu_+) - (\mu_- + \rho/\theta)\mathcal{I}_0(\mu_-)}{\mu_+ - \mu_-} - c. \end{aligned} \tag{37}$$

The second row gives the four continuing prediction coefficients:

$$\begin{aligned} e &= \frac{\rho}{\theta}\frac{(\mu_+ + \lambda)\mathcal{I}_\lambda(\mu_+) - (\mu_- + \lambda)\mathcal{I}_\lambda(\mu_-)}{\mu_+ - \mu_-}, \\ f &= \frac{(\mu_+ + \lambda)e^{\mu_+\Delta} - (\mu_- + \lambda)e^{\mu_-\Delta}}{\mu_+ - \mu_-}, \\ g &= \frac{\lambda n\rho}{\theta}\frac{\mathcal{I}_\rho(\mu_+) - \mathcal{I}_\rho(\mu_-)}{\mu_+ - \mu_-}, \\ h &= \frac{(\lambda + \rho)n\rho}{\theta}\frac{\mathcal{I}_0(\mu_+) - \mathcal{I}_0(\mu_-)}{\mu_+ - \mu_-} - g. \end{aligned} \tag{38}$$

To verify these expressions, variation of constants for the two variables reads

$$\begin{pmatrix} r(\Delta) \\ p(\Delta) \end{pmatrix} = e^{\Delta\mathsf{A}_{rp}}\begin{pmatrix} 0 \\ p_0 \end{pmatrix} + \int_0^{\Delta} e^{(\Delta - s)\mathsf{A}_{rp}}\begin{pmatrix} (\lambda + \rho)v + \lambda e^{-\rho s}(u_0 - v) \\ (\rho/\theta)e^{-\lambda s}M_0 k \end{pmatrix}\mathrm{d}s. \tag{39}$$

Collecting the coefficients of $M_0 k, p_0, u_0, v$ yields the displayed formulas.

Complex conjugate eigenvalues describe damped oscillations; the paired expressions above are real. At a repeated eigenvalue $\mu_+ = \mu_- = \mu$, every divided difference $[f(\mu_+) - f(\mu_-)]/(\mu_+ - \mu_-)$ takes its derivative limit $f'(\mu)$. Equivalently, with I the two dimensional identity matrix, $e^{s\mathsf{A}_{rp}} = e^{\mu s}[I + s(\mathsf{A}_{rp} - \mu I)]$. These limits include critical damping. The five dimensional exponential in Equation (31) evaluates the same response continuously at zero rates, zero keys, and coincident poles.

B.4 The augmented Delta form

Proposition 2 (Exact augmented Delta recurrence)**.** *Append the prediction and evidence columns to the relation matrix to obtain $S = [M, p, u] \in \mathbb{R}^{d_v \times (d_k+2)}$. Let Q_* select the current key and the two trace coordinates. Let C collect their homogeneous response coefficients and let t collect the coefficients of the incoming value. Explicitly,*

$$Q_* = \begin{pmatrix} k & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} a & e & 0 \\ b & f-\alpha & 0 \\ c & g & j-\alpha \end{pmatrix}, \quad t = \begin{pmatrix} d \\ h \\ 1-j \end{pmatrix}. \tag{40}$$

Then the event response of Equation (25) is

$$S_+ = \alpha S + (SQ_*C + vt^\top)Q_*^\top, \qquad o = S_+ \begin{pmatrix} q \\ 0 \\ 0 \end{pmatrix}. \tag{41}$$

Its homogeneous state transition is scalar decay plus a correction of rank at most three. The identity holds for arbitrary keys, including a zero key.

Proof. The three columns of SQ_* are Mk, p, u. Multiplying these by C, adding $vt^\top$, and multiplying by $Q_*^\top$ gives, respectively, the matrix write $r_*k^\top$, the prediction correction $p_+ - \alpha p$, and the evidence correction $u_+ - \alpha u$. Adding αS gives Equation (32). □

The coefficients are fixed response functions of $\lambda, \rho, n, \theta, \Delta$. Their signs and the possible asymmetry of C follow from the interacting prediction and memory dynamics. The three columns of Q_* identify all directions that interact during the event: the current key, prediction, and evidence. Every other direction undergoes the same retention factor α. Equation (41) therefore places the model in the family of gated Delta recurrences with an augmented state. A head adds $2d_v$ persistent scalars to its matrix memory, and its sequential evaluation uses matrix–vector products and outer products. The features used here can be signed, as in the implemented Delta architecture.

B.5 Persistent evidence and canonical Gated DeltaNet

For a held unit key, meaning $k^\top k = 1$, and $\rho > 0$, prediction approaches Mk as $\theta \to 0$. Bounded initial prediction discrepancies relax in a boundary layer of duration $O(\theta/\rho)$, whose integrated contribution to the matrix update vanishes in this limit. To make this limit explicit, let $m(s) = M(s)k$ and let $\varepsilon_p = p - m$ be the prediction lag. Its equation and solution are

$$\begin{aligned} \dot{\varepsilon}_p &= -\frac{\rho}{\theta}\varepsilon_p - \dot{m}, \\ \varepsilon_p(s) &= e^{-\rho s/\theta}\varepsilon_p(0) - \int_0^s e^{-\rho(s-t)/\theta}\dot{m}(t)\,\mathrm{d}t. \end{aligned} \tag{42}$$

For bounded inputs and initial states on a finite interval, the linear response has a bounded $\dot{m}$. Thus $\|\varepsilon_p(s)\| \le e^{-\rho s/\theta}\|\varepsilon_p(0)\| + (\theta/\rho)\sup_{t\le s}\|\dot{m}(t)\|$. Its contribution to the integrated memory write is $O(\theta)$, including the initial transient. The limiting memory equation therefore uses $p = Mk$ throughout the event.

With $\beta = 1 - e^{-\rho\Delta}$ and incoming evidence u_0, the event response is

$$\begin{aligned} u_+ &= (1-\beta)u_0 + \beta v, \\ M_+ &= \alpha M_0(I - \beta kk^\top) + [\beta v + \ell u_0]k^\top, \qquad \ell = (1-\alpha)(1-\beta). \end{aligned} \tag{43}$$

To see this directly, let I denote the identity in key space and use $P_k = kk^\top$, a projector for a unit key. The component $M(I - P_k)$ decays by α. The component $m = Mk$ obeys $\dot{m} = -(\lambda + \rho)m + \lambda u + \rho v$ with $u(s) = v + e^{-\rho s}(u_0 - v)$. Integrating this scalar linear response in each value row gives Equation (43), including its continuous limit at $\lambda = 0$. Explicitly, variation of constants gives

$$\begin{aligned} m(s) &= e^{-(\lambda+\rho)s} m_0 + \int_0^s e^{-(\lambda+\rho)(s-t)} [\lambda u(t) + \rho v] \, \mathrm{d}t \\ &= e^{-(\lambda+\rho)s} m_0 + (1 - e^{-\rho s}) v + e^{-\rho s}(1 - e^{-\lambda s}) u_0. \end{aligned}$$

At $s = \Delta$, $M_+ = \alpha M_0 + (m_+ - \alpha m_0)k^\top$, which gives the stated matrix response.

The canonical Gated DeltaNet update is

$$M_+^{\mathrm{GDN}} = \alpha M_0 (I - \beta k k^\top) + \beta v k^\top. \tag{44}$$

Persistent evidence contributes $\ell u_0 k^\top$ alongside the current value. To express the content and magnitude of this write, define

$$\eta_w = \beta + \ell = 1 - \alpha(1 - \beta), \qquad \bar{v} = \frac{\beta v + \ell u_0}{\eta_w} \quad (\eta_w > 0). \tag{45}$$

The write becomes $\eta_w \bar{v} k^\top$. Its content $\bar{v}$ is a convex combination of current and persistent evidence, and its strength is η_w. At $\eta_w = 0$, the write vanishes. The erase and write magnitudes can differ while both remain determined by the retention and acquisition rates.

Two circuit schedules yield Equation (44). In the first, the evidence coordinate is cleared at the start of each event and prediction is instantaneous. An incomplete reset leaves the term $\ell u_0 k^\top$ determined by its remaining trace. The second schedule uses a pure decay phase, with the anchoring and correction pathways closed, followed by a correction phase. Decay gives $M_A = \alpha M_0$. During correction, the decay and anchoring pathways close and the circuit obeys

$$\dot{M} = \rho(v - Mk)k^\top. \tag{46}$$

For a unit key, $m = Mk$ then obeys $\dot{m} = \rho(v - m)$ while orthogonal matrix components remain fixed, giving $M_+ = M_A(I - \beta k k^\top) + \beta v k^\top$. Here α and β are set by the rates integrated over their respective phases. This schedule requires selective pathway control and instantaneous prediction for the canonical form. Finite prediction retains its additional state. Holding $u = v$ throughout a simultaneous decay and correction event gives the related write strength $1 - \alpha(1 - \beta)$. The trained HACANN uses the continuing evidence and finite prediction dynamics of Equation (25).

The unit key condition has a specific role: it makes $kk^\top$ a projector and makes the prediction relaxation rate equal to ρ. For a general nonzero key, pure correction with instantaneous prediction integrates to

$$M_+ = M_A + \frac{1 - e^{-n\rho\Delta}}{n}(v - M_A k)k^\top, \qquad n = k^\top k > 0. \tag{47}$$

This follows by solving $\frac{\mathrm{d}}{\mathrm{d}t}(Mk) = n\rho(v - Mk)$. Its coefficient tends to $\rho\Delta$ as $n \to 0$, while the outer product vanishes at $k = 0$. The general finite response uses the measured key norm throughout, including the small departure from unit norm caused by the normalization constant in the implemented model.

B.6 Conditional dissipation

Consider two trajectories with the same k, v, λ, ρ and the same u. Denote their memory and prediction differences by δM and δp, and use $\|\cdot\|_F$ for the Frobenius matrix norm and $\|\cdot\|$ for the Euclidean vector norm. Their differences obey $\delta\dot{M} = -\lambda\delta M - \rho\delta p\, k^\top$ and $\theta\delta\dot{p} = \rho(\delta M k - \delta p)$. Therefore

$$\mathcal{E} = \frac{1}{2}\|\delta M\|_F^2 + \frac{\theta}{2}\|\delta p\|^2, \qquad \dot{\mathcal{E}} = -\lambda\|\delta M\|_F^2 - \rho\|\delta p\|^2 \le 0. \tag{48}$$

Indeed, the two mixed terms are $-\rho\,\delta p^\top \delta M k$ and $+\rho\,\delta p^\top \delta M k$, which cancel. Prediction and correction exchange their contributions to the energy while decay and prediction relaxation dissipate

it. The identity holds for time dependent keys and nonnegative gains at any key norm. If $\lambda \geq \lambda_0 > 0$ and $\rho \geq \rho_0 > 0$, it also gives

$$\mathcal{E}(t) \leq e^{-2\min(\lambda_0, \rho_0/\theta)t}\mathcal{E}(0).$$

When a gain closes, the nonincreasing energy permits persistent components. If evidence also differs, the derivative acquires $\lambda\langle \delta M, \delta u\, k^\top\rangle$, with $\delta\dot{u} = -\rho\delta u$. The result therefore gives conditional dissipation of the recurrent memory for a shared evidence trajectory.

For a held key, let s denote an eigenvalue of the homogeneous (Mk, p) response. Its characteristic equation is

$$\theta s^2 + (\theta\lambda + \rho)s + \rho(\lambda + n\rho) = 0. \tag{49}$$

For $n > 0$ and $\rho > 0$, the choice $\theta = 1/(4n)$ is critically damped when $\lambda = 0$. The experiments fix $\theta = 0.25$ before training, corresponding to that reference point for a unit key. Other gain settings can yield damped oscillations while retaining Equation (48).

C Credit dynamics and local plasticity

C.1 Credit during replay

A credit variable is the derivative of the retained task loss with respect to a represented state. Let P, a_p, a_u be the credits on M, p, u, with the same matrix or vector shapes as their corresponding states, and let T be the duration of the retained forward trajectory. During replay, a physical clock s increases while the stored forward address $t = T - s$ decreases. The replay dynamics apply the transpose of the local forward Jacobian to the credit variables evaluated at the actual recorded state:

$$\begin{aligned} P' &= -\lambda P + (\rho/\theta)a_p k^\top, \\ a_p' &= -\rho P k - (\rho/\theta)a_p, \\ a_u' &= \lambda P k - \rho a_u. \end{aligned} \tag{50}$$

The prime denotes positive physical replay time. A terminal credit starts the calculation at the end of the retained window; records then supply the earlier forward states in reverse address order. Here and below, an overbar denotes the derivative of the loss with respect to the indicated quantity. A readout credit $\bar{o}$ at its stored address adds $P \mathrel{+}= \bar{o}q^\top$ and gives $\bar{q} = M^\top\bar{o}$. The adjoint equations follow by collecting the coefficient of each state perturbation in the differential of the forward flow. Appendix E derives the state reconstruction.

Let $b_v, b_k, b_\lambda, b_\rho$ denote the instantaneous contributions to the credits of the value, key, turnover rate, and acquisition rate. During replay these densities are

$$\begin{aligned} b_v &= \rho(Pk + a_u), \\ b_k &= P^\top[\lambda u + \rho(v - p)] + (\rho/\theta)M^\top a_p, \\ b_\lambda &= -\langle P, M - uk^\top\rangle, \\ b_\rho &= (v - p)^\top Pk + \theta^{-1}a_p^\top(Mk - p) + a_u^\top(v - u). \end{aligned} \tag{51}$$

Integrating these densities gives the credit of a held event input. Their terms are local products together with row, column, or head sums. The operation $M^\top a_p$ reads the current or reconstructed matrix of population coordinates. Learned projections between layers use separately stored feedback synapses, described below.

Forward weights remain fixed while a retained window is recorded and replayed, after which its accumulated plasticity is applied. Setting credit beyond the window to zero gives truncated temporal credit with the incoming neural state held fixed during differentiation. Language experiments use 1,024-byte credit windows and carry their forward state between windows. Recorded inputs, checkpoints, address control, buffers, and replay time provide the physical resources for this learning schedule.

C.2 Exact discrete credit for the finite event

This subsection uses unit event duration, as in the task models. For a general interval Δ, the coefficient derivatives act on $\exp(\Delta A)$ and the direct derivatives of $\alpha = e^{-\lambda\Delta}$ and $j = e^{-\rho\Delta}$ acquire a factor Δ.

Integrating the continuous adjoint or differentiating the finite event response gives the same derivative. Let G, P_p, P_u be the incoming credits on M_+, p_+, u_+, after adding any current readout credit to G. Let $m = M_0 k$ be the incoming memory read, let $R = Gk$ be the credit on the write vector r_*, and let $\bar{m} = aR + eP_p$ be the credit on that incoming read. The scalar $\bar{n}$ is the credit contributed by the response coefficients' dependence on the squared key norm $n = k^\top k$. Applying the chain rule to Equation (32) gives

$$\begin{aligned} \bar{M}_0 &= \alpha G + \bar{m}k^\top, & \bar{p}_0 &= bR + fP_p, \\ \bar{u}_0 &= cR + gP_p + jP_u, & \bar{v} &= dR + hP_p + (1-j)P_u, \\ \bar{k} &= G^\top r_* + M_0^\top \bar{m} + 2k\bar{n}. \end{aligned} \tag{52}$$

The eight coefficient credits are

$$(\bar{a}, \bar{b}, \bar{c}, \bar{d}) = (R^\top m, R^\top p_0, R^\top u_0, R^\top v), \tag{53}$$

and the corresponding four inner products with P_p give $(\bar{e}, \bar{f}, \bar{g}, \bar{h})$. Contracting these with coefficient derivatives gives their contributions to $\bar{\lambda}, \bar{\rho}, \bar{n}$. Two direct dependencies add

$$\bar{\lambda}_{\text{direct}} = -\alpha\langle G, M_0\rangle, \qquad \bar{\rho}_{\text{direct}} = jP_u^\top(v - u_0). \tag{54}$$

The coefficient derivatives use the Fréchet derivative of e^A. For a matrix perturbation H of the generator, this derivative is

$$D\exp(A)[H] = \int_0^1 e^{(1-s)A} H e^{sA}\,\mathrm{d}s.$$

Write $A_\lambda = \partial A/\partial\lambda$, $A_\rho = \partial A/\partial\rho$, and $A_n = \partial A/\partial n$. Taking $H = A_\lambda, A_\rho, A_n$ and selecting the same eight entries as in Equation (31) gives the required coefficient derivatives. The nonzero derivatives of the generator are

$$\begin{aligned} A_\lambda:&\quad (0,0) = -1,\ (0,2) = 1,\ (4,4) = -1, \\ A_\rho:&\quad (0,1) = -1,\ (0,3) = 1,\ (1,0) = n/\theta, \\ &\quad (1,1) = -1/\theta,\ (1,4) = 1/\theta,\ (2,2) = -1,\ (2,3) = 1, \\ A_n:&\quad (1,0) = \rho/\theta. \end{aligned} \tag{55}$$

These derivatives depend on the displayed rates and squared key norm. The implementation propagates directional derivatives through a small scaling and squaring calculation. If $B = A/2^s$, Taylor terms obey $T_j = T_{j-1}B/j$ and $\mathrm{d}T_j = (\mathrm{d}T_{j-1}B + T_{j-1}\mathrm{d}B)/j$. Each squaring propagates $\mathrm{d}E \leftarrow (\mathrm{d}E)E + E(\mathrm{d}E)$ using the old E. This evaluates the fixed response function and its derivatives in the digital implementation. The corresponding circuit evolves through the finite neural flow. Independent finite differences verify the resulting derivative calculation.

Let $a_{\text{raw}}, b_{\text{raw}}$ be the raw retention and acquisition gate activities, respectively. Define $\mathrm{sigmoid}(x) = (1 + e^{-x})^{-1}$ and $\mathrm{softplus}(x) = \log(1 + e^x)$. With $\alpha = \mathrm{sigmoid}(a_{\text{raw}})$ and $\beta = \mathrm{sigmoid}(b_{\text{raw}})$, numerically regular rate variables are

$$\begin{aligned} \lambda &= \mathrm{softplus}(-a_{\text{raw}}), & \partial_{a_{\text{raw}}}\lambda &= -(1-\alpha), \\ \rho &= \mathrm{softplus}(b_{\text{raw}}), & \partial_{b_{\text{raw}}}\rho &= \beta. \end{aligned} \tag{56}$$

For duration Δ, both rates and their derivatives are divided by Δ. These softplus expressions evaluate the rates directly from gate activities and remain regular as either gate approaches its boundary.

C.3 Normalization and interlayer credit

Let z be the raw key vector and let $\epsilon > 0$ be the normalization floor. The implemented key is $k = z/r$, where $r = (z^\top z + \epsilon)^{1/2}$ is its normalization factor. Differentiating this transformation returns credit as

$$\bar{z} = \frac{\bar{k}}{r} - \frac{z(z^\top \bar{k})}{r^3}. \tag{57}$$

The $2k\bar{n}$ term in Equation (52) returns credit through the squared key norm, including the effect of the normalization constant. For a vector of width d, root mean square normalization gives $y = z/r$

with $r = (d^{-1}z^\top z + \epsilon)^{1/2}$. The corresponding credit is $\bar{z} = \bar{y}/r - z(z^\top \bar{y})/(dr^3)$. For a vector input x and a vector gate g, write $\odot$ for componentwise multiplication. The multiplicative gate $y = x \odot g$ gives $\bar{x} = \bar{y} \odot g$ and $\bar{g} = \bar{y} \odot x$. Let $\sigma(z) = (1 + e^{-z})^{-1}$ denote the logistic function. For $\mathrm{SiLU}(z) = z\sigma(z)$, the local derivative is $\sigma(z) + z\sigma(z)[1 - \sigma(z)]$. These peripheral rate operations use local derivatives and shared inner product pools.

Let n and m denote the input and output widths of a learned projection in this subsection. For input row x and output row y, write $y = xW$, with $W \in \mathbb{R}^{n \times m}$ and an independently stored feedback projection $B \in \mathbb{R}^{m \times n}$. The implemented return signal is $\bar{x} = \bar{y}B$. The two directions have local plasticity drives $x_i \bar{y}_a$ and $\bar{y}_a x_i$, using the saved endpoint activity and credit. Each direction updates its own synaptic state and chemical traces. Reciprocal responses, including $B = W^\top$, give the conventional network gradient. Independently evolving pathways transmit approximate credit when their responses differ. The experiments measure the agreement and task performance of the heterogeneous, noisy synapses under their own local updates.

Task supervision reaches the output population as the observed next input or requested answer. Hidden populations receive the resulting credit through the independently stored feedback pathways.

C.4 Learning reciprocity from activity responses

Two independent local probes, a row $\xi \in \mathbb{R}^n$ and a row $\zeta \in \mathbb{R}^m$, test the forward and feedback projections. Their responses are y and z, and δ denotes the discrepancy between the two scalar endpoint comparisons:

$$y = \xi W, \qquad z = \zeta B, \qquad \delta = \zeta y^\top - z\xi^\top. \tag{58}$$

A module pool communicates this scalar response difference. For feedback synapse (a, i), let e_{ai} be its eligibility trace, q_{ai} its instantaneous calibration drive, $\tau_{ai} > 0$ its response time, and $c_{ai} > 0$ its plasticity gain. A common positive learning scale η gives

$$\tau_{ai}\dot{e}_{ai} = q_{ai} - e_{ai}, \qquad \dot{B}_{ai} = c_{ai}e_{ai}, \qquad q_{ai} = \eta\zeta_a\delta\xi_i. \tag{59}$$

Each update uses the two probe activities, their measured response difference, and the synapse's own eligibility trace.

Let q collect the synaptic drives q_{ai} into a matrix, and let I_n, I_m be the identity matrices in the two endpoint spaces. For independent probes satisfying $\mathbb{E}[\xi^\top \xi] = I_n/n$ and $\mathbb{E}[\zeta^\top \zeta] = I_m$,

$$\mathbb{E}[q] = \frac{\eta}{n}(W^\top - B). \tag{60}$$

Indeed, expanding the response difference gives $\delta = \sum_{b,j} \zeta_b(W_{jb} - B_{bj})\xi_j$; independence of the probes contracts the expectation of $\zeta_a\xi_i\delta$ to $(W_{ia} - B_{ai})/n$. For K input probes with summed drive, the factor is $\eta K/n$. Holding W fixed, the averaged dynamics of each feedback synapse are

$$\tau_{ai}\ddot{B}_{ai} + \dot{B}_{ai} = \frac{c_{ai}\eta K}{n}(W_{ia} - B_{ai}). \tag{61}$$

The homogeneous characteristic polynomial has positive coefficients, so its roots have negative real parts. Positive heterogeneity changes the response time and damping while preserving the reciprocal fixed point of the averaged system. Finite probes and noise give a precision and cost tradeoff, and the input width enters the convergence rate. When W changes, its motion drives the tracking error. Tracking a changing forward pathway adds a timescale requirement set by that motion.

The endpoint populations generate independent probes. Calibration stops according to agreement of their measured activity responses. Exact credit is evaluated separately for the numerical gradient comparisons.

C.5 Finite local chemical dynamics

In the language experiments with heterogeneous plasticity, each synapse stores its own weight w, eligibility trace e, and square trace s. Let g be its local task drive, ξ its independent unit-scale noise sample, and σ the noise amplitude. Clipping the observed drive to $[-1, 1]$ gives

$$g_{\mathrm{obs}} = \mathrm{clip}(g(1 + \sigma\xi), -1, 1), \qquad s_+ = 0.99s + 0.01g_{\mathrm{obs}}^2.$$

The square-trace update is the finite response of $\dot{s} = -\log(0.99)(g_{\text{obs}}^2 - s)$ over a unit interval with the observed drive held. The normalized drive is $h = g_{\text{obs}}/(\sqrt{s_+} + 10^{-6})$. During a learning interval, hold h fixed and let $\tau_e > 0$ be the eligibility response time, $c > 0$ the local plasticity gain, $\eta_{\text{lr}} > 0$ the learning rate, and $\kappa_w \geq 0$ the weight turnover rate. The chemical dynamics are

$$\tau_e \dot{e} = h - e, \qquad \dot{w} = -c(\kappa_w w + \eta_{\text{lr}} e). \tag{62}$$

To evaluate these dynamics over duration Δ, define two exponential response integrals for nonnegative rates r_1, r_2:

$$\begin{aligned}
\Phi(r_1, \Delta) &= \int_0^\Delta e^{-r_1(\Delta - t)}\, \mathrm{d}t = \begin{cases} (1 - e^{-r_1\Delta})/r_1, & r_1 \neq 0, \\ \Delta, & r_1 = 0, \end{cases} \\
\Psi(r_1, r_2, \Delta) &= \int_0^\Delta e^{-r_1(\Delta - t)} e^{-r_2 t}\, \mathrm{d}t \\
&= \begin{cases} (e^{-r_2\Delta} - e^{-r_1\Delta})/(r_1 - r_2), & r_1 \neq r_2, \\ \Delta e^{-r_1\Delta}, & r_1 = r_2. \end{cases}
\end{aligned}$$

Let e_0, w_0 be the incoming eligibility and weight. Solving for eligibility first and inserting it into the weight equation gives

$$\begin{aligned}
e_+ &= h + (e_0 - h)e^{-\Delta/\tau_e}, \\
w_+ &= e^{-c\kappa_w\Delta} w_0 - c\eta_{\text{lr}} h \Phi(c\kappa_w, \Delta) \\
&\quad - c\eta_{\text{lr}}(e_0 - h)\Psi(c\kappa_w, 1/\tau_e, \Delta).
\end{aligned} \tag{63}$$

The continuous limits above cover zero weight turnover and equal response rates. Forward and reverse synapses carry independent square traces, eligibility traces, gains, and noise. The main finite plasticity experiments use $c \in [0.8, 1.2]$, $\tau_e \in [0.2, 0.8]$ learning-event durations, $\sigma = 0.02$, and $\kappa_w = 0$. Embeddings, biases, normalization parameters, and temporal filters use their specified local square-trace rule. Chemical learning intervals supply the time units of this model. The relation learning protocol uses the optimizer and noise settings stated in Appendix H.

D Parallel evaluation through causal elimination

Parallel evaluation uses the causal structure of the finite response: a write depends on earlier writes through their overlaps with its key. Let a chunk contain events $t = 1, \ldots, T$ with arbitrary incoming M_0, p_0, u_0. Evaluate the eight response coefficients at event t and append that event index to their symbols. To make their sources visible in the sequence equations, write

$$(c_{m,t}, c_{p,t}, c_{u,t}, c_{v,t}) = (a_t, b_t, c_t, d_t), \qquad (d_{m,t}, d_{p,t}, d_{u,t}, d_{v,t}) = (e_t, f_t, g_t, h_t).$$

Let w_t be the write vector and $m_t = M_{t-1}k_t$ the incoming memory read at event t. Equation (32) becomes

$$\begin{aligned}
M_t &= \alpha_t M_{t-1} + w_t k_t^\top, & m_t &= M_{t-1}k_t, \\
w_t &= c_{m,t} m_t + c_{p,t} p_{t-1} + c_{u,t} u_{t-1} + c_{v,t} v_t, \\
p_t &= d_{m,t} m_t + d_{p,t} p_{t-1} + d_{u,t} u_{t-1} + d_{v,t} v_t, \\
u_t &= j_t u_{t-1} + (1 - j_t) v_t.
\end{aligned} \tag{64}$$

All coefficients retain their dependence on the actual key norm. The independent affine recurrence for u is evaluated first. Store each vector sequence as a $T \times d_v$ matrix, with one transposed vector in each row. Define the overlap matrix H of past writes with later keys, and the incoming-memory contribution Z, by

$$\begin{aligned}
H_{tj} &= \mathbf{1}_{j<t} \left(\prod_{i=j+1}^{t-1} \alpha_i \right) k_j^\top k_t, \\
Z_t &= \left(\prod_{i=1}^{t-1} \alpha_i \right) M_0 k_t.
\end{aligned} \tag{65}$$

Empty products equal one. Unrolling the matrix recurrence gives $m = Z + Hw$, with H strictly lower triangular.

Let D_x denote the diagonal matrix of a scalar coefficient sequence x; for example, D_{d_p} has entries $d_{p,t}$ on its diagonal. The matrix L carries the effect of a past prediction input to a later prediction state. Let b^p contain the contribution of the initial prediction, and let f^p, f^w collect the direct evidence and value terms. These quantities are

$$\begin{aligned} L_{tj} &= \mathbf{1}_{j<t} \prod_{i=j+1}^{t-1} d_{p,i}, & b_t^p &= \left(\prod_{i=1}^{t-1} d_{p,i} \right) p_0, \\ f_t^p &= d_{u,t} u_{t-1} + d_{v,t} v_t, & f_t^w &= c_{u,t} u_{t-1} + c_{v,t} v_t. \end{aligned} \tag{66}$$

Let p_{before} collect the incoming predictions p_{t-1}, and define K and b as the matrix and offset relating writes to memory reads. Eliminating the prediction trace gives

$$\begin{aligned} p_{\text{before}} &= b^p + L(D_{d_m} m + f^p), \\ w &= Km + b, \qquad K = D_{c_m} + D_{c_p} L D_{d_m}, \\ b &= D_{c_p}(b^p + L f^p) + f^w. \end{aligned} \tag{67}$$

Hence

$$(I - KH)w = KZ + b. \tag{68}$$

The product KH is strictly lower triangular, so the diagonal is exactly one. Eliminating the two persistent traces leaves a solve of dimension T with d_v right hand sides.

The structured product $S = L D_{d_m} H$ can be computed through the row recurrence

$$S_1 = 0, \qquad S_t = d_{p,t-1} S_{t-1} + d_{m,t-1} H_{t-1}. \tag{69}$$

An associative affine scan on rows of length T constructs S in $O(T^2)$ work and $O(\log T)$ scan stages. The subsequent triangular solve is a separate operation. Products and affine scans also handle zero or negative prediction coefficients, since the construction avoids division by these coefficients.

Let p^{free} be the prediction sequence driven only by the incoming memory and the direct evidence and value terms. It supplies the right hand side R_c of a triangular system with coefficient matrix A_c:

$$\begin{aligned} p^{\text{free}} &= b^p + L(D_{d_m} Z + f^p), \\ A_c &= I - D_{c_m} H - D_{c_p} S, \\ R_c &= D_{c_m} Z + D_{c_p} p^{\text{free}} + f^w, \qquad A_c w = R_c. \end{aligned} \tag{70}$$

After solving, $m = Z + Hw$ and another affine scan recovers p. The readouts are

$$o_t = \left(\prod_{i=1}^{t} \alpha_i \right) M_0 q_t + \sum_{j \le t} \left(\prod_{i=j+1}^{t} \alpha_i \right) w_j (k_j^\top q_t). \tag{71}$$

The terminal matrix has the analogous weighted outer product sum; the terminal vector traces pass to the next chunk.

The arithmetic per chunk is $O(T^2 d_k + T d_k d_v + T^2 d_v)$, together with $O(T^2 + T d_v)$ trace work and small coefficient calculations. This is the same block asymptotic order as dense Delta chunk arithmetic. The construction uses additional scalar arrays, so practical constants and data movement remain relevant. For incoming credit $\bar{w}$, solve $A_c^\top Y = \bar{w}$ for the auxiliary credit matrix Y. Differentiation then gives $\bar{R}_c = Y$ and $\bar{A}_c = -Y w^\top$, followed by reverse scans and products. The chunk implementation verifies equivalence to sequential forward evaluation. Task training uses the sequential event response and its explicit adjoint.

E Reconstructing past states from their updates

The matrix recurrence also provides a causal representation of its history. In a replay segment of length C, store its incoming matrix M_0 and event records $(w_j, k_j, \alpha_j, p_{j-1}, u_{j-1})$ for $j = 1, \ldots, C$.

Let A_t be the accumulated retention from the segment start and a_{tj} the retention between write j and state t:

$$A_t = \prod_{i=1}^{t} \alpha_i, \qquad a_{tj} = \prod_{i=j+1}^{t} \alpha_i. \tag{72}$$

Repeated substitution gives

$$M_t = A_t M_0 + \sum_{j \le t} a_{tj} w_j k_j^\top. \tag{73}$$

Multiplication forms every attenuation coefficient, so this representation also covers zero retention factors. Finite decay rates give $\alpha > 0$. The stored writes are those produced during the forward trajectory from the inputs available at their event addresses.

For a key-space vector $x \in \mathbb{R}^{d_k}$ and a value-space vector $y \in \mathbb{R}^{d_v}$, the required earlier matrix actions are

$$\begin{aligned} M_t x &= A_t M_0 x + \sum_{j \le t} a_{tj} w_j (k_j^\top x), \\ M_t^\top y &= A_t M_0^\top y + \sum_{j \le t} a_{tj} k_j (w_j^\top y). \end{aligned} \tag{74}$$

These expressions evaluate the required matrix actions directly from their records. Decay credit additionally requires $\langle G, M_t \rangle$, where $G \in \mathbb{R}^{d_v \times d_k}$ is the current memory credit and brackets denote the Frobenius inner product. Store the scalar pairings c_0 with the incoming matrix and c_j with each recorded write:

$$c_0 = \langle G, M_0 \rangle, \qquad c_j = w_j^\top G k_j. \tag{75}$$

A credit update with scalar retention a and outer product $yx^\top$, namely $G \leftarrow aG + yx^\top$, then induces

$$c_0 \leftarrow a c_0 + y^\top M_0 x, \qquad c_j \leftarrow a c_j + (w_j^\top y)(k_j^\top x), \tag{76}$$

and the required inner product is

$$\langle G, M_t \rangle = A_t c_0 + \sum_{j \le t} a_{tj} c_j. \tag{77}$$

These identities follow by bilinearity. The event adjoint uses these operations: a readout injection $G \leftarrow G + \bar{o}_t q_t^\top$, followed by $G \leftarrow \alpha_t G + \bar{m}_t k_t^\top$.

Initializing the pairings for an arbitrary incoming credit matrix costs $O(C d_v d_k)$. Subsequent record reads and pairing updates across the segment cost $O(C^2(d_v + d_k))$. Thus the total is

$$O\big(C d_v d_k + C^2(d_v + d_k)\big), \tag{78}$$

with persistent replay state $O(d_v d_k + C(3d_v + d_k))$ plus scalar gates and pairings. The current credit matrix, common network activations, and temporary workspaces are still present. A longer segment trades fewer matrix checkpoints for more vector-record work. The record implementation evaluates the complete recurrent adjoint and is checked against event differentiation. Reported training costs refer to the executed event implementation and its allocated state.

F Finite key formation and transmission through layers

F.1 A key that forms over a finite interval

A physical key can continue to form while it updates the relation. Let κ be its unnormalized activity, let $\kappa_\star$ be the target held by the input stage, and let h track its squared amplitude. The positive constants τ_k and τ_h specify their response times. A positive floor ϵ_{norm} bounds the normalization gain $g(t)$. The key and its normalization pool then obey

$$\tau_k \dot{\kappa} = \kappa_\star - \kappa, \qquad \tau_h \dot{h} = \|\kappa\|^2 - h, \qquad k(t) = g(t)\kappa(t), \quad g(t) = (h(t) + \epsilon_{\mathrm{norm}})^{-1/2}. \tag{79}$$

The following reduction holds the target $\kappa_\star$, value v, and the two rates fixed during the event. Let κ_0 be the incoming raw key. With a common key time constant, its trajectory lies in the span of κ_0

and $\kappa_\star$. Collect these directions in the basis matrix B_k, let $s(t)$ contain their exponential mixing weights, let $c(t)$ include the normalization gain, and let G_k be their Gram matrix:

$$B_k = [\kappa_0, \kappa_\star], \qquad s(t) = \begin{pmatrix} e^{-t/\tau_k} \\ 1 - e^{-t/\tau_k} \end{pmatrix}, \qquad c(t) = g(t)s(t), \quad G_k = B_k^\top B_k, \tag{80}$$

Then $k(t) = B_k c(t)$. Let $a(t)$ be memory retention, let $W_k(t)$ collect the writes along the two basis directions, and let $X(t)$ contain the retained incoming reads in those directions. The memory can be written as

$$M(t) = a(t)M_0 + W_k(t)B_k^\top, \quad a(t) = e^{-\lambda t}, \quad X(t) = a(t)M_0 B_k. \tag{81}$$

Substitution in Equation (25) gives

$$\begin{aligned} \dot{W}_k &= -\lambda W_k + [\lambda u + \rho(v - p)]c^\top, \\ \theta\dot{p} &= \rho[Xc + W_k G_k c - p], \\ \dot{u} &= \rho(v - u), \qquad \dot{X} = -\lambda X, \qquad\qquad W_k(0) = 0. \end{aligned} \tag{82}$$

The pool drive is $s^\top G_k s$. In each value row, write w_1, w_2 for the entries of W_k and x_1, x_2 for the entries of X. The seven variables $(w_1, w_2, p, u, x_1, x_2, v)$ follow a linear equation with time dependent coefficients set by $c(t)$. Integrating the common response of these seven variables and the scalar pool recovers all value rows. The full matrix is accessed to form $M_0 B_k$ and the terminal $W_k B_k^\top$.

The augmented transition differs from aI within the span of the two key directions and the two trace directions, giving a correction of rank at most four. Equal or dependent columns of B_k remain valid because the reduction uses products with G_k. The circuit itself follows Equations (25) and (79). The basis B_k, Gram matrix G_k, coefficients W_k, and shared seven dimensional response provide an efficient numerical solution of those physical dynamics.

F.2 A RANK BUDGET FOR HETEROGENEOUS KEY TIME CONSTANTS

Let $a_0 > 0$ be a reference key relaxation rate, δ_i the deviation of coordinate i, and ε the maximum relative heterogeneity. Suppose coordinate i has decay rate $a_i = a_0 + \delta_i$, with $|\delta_i| \le \varepsilon a_0$ and $0 \le \varepsilon < 1$. Its activity is $\kappa_i(t) = \kappa_{\star,i} + d_i e^{-(a_0+\delta_i)t}$, where $d = \kappa_0 - \kappa_\star$ is the initial displacement from the target. For a positive integer r, retain r terms of the expansion of $e^{-\delta_i t}$. With $\odot$ denoting componentwise multiplication and δ^j the vector of componentwise powers, the resulting approximation is

$$\widehat{\kappa}(t) = \kappa_\star + e^{-a_0 t} \sum_{j=0}^{r-1} \frac{(-t)^j}{j!} (\delta^j \odot d). \tag{83}$$

The key basis has dimension at most $r+1$, the response dimension is $2(r+1)+3$, and the augmented event correction has rank at most $r + 3$. The exponential Taylor remainder gives

$$\|\kappa(t) - \widehat{\kappa}(t)\| \le e^{-(1-\varepsilon)a_0 t} \frac{(\varepsilon a_0 t)^r}{r!} \|d\|. \tag{84}$$

The scalar factor $e^{-(1-\varepsilon)a_0 t} t^r$ reaches its maximum at $t = r/[(1-\varepsilon)a_0]$. Substitution gives

$$\sup_{t \ge 0} \frac{\|\kappa(t) - \widehat{\kappa}(t)\|}{\|d\|} \le \left(\frac{\varepsilon}{1-\varepsilon}\right)^r \frac{r^r e^{-r}}{r!}. \tag{85}$$

For $d = 0$ the error is zero and the ratio is unnecessary. At $\varepsilon = 0.2, r = 4$, the bound is 7.6316×10^{-4}, using five key directions and an event correction of rank at most seven. This measures the unnormalized key error relative to the initial-to-target displacement; the effect on the relation follows from the forced error dynamics below.

To include the pool, let K_0 bound the raw-key amplitude and E_0 bound its approximation error: $\|\kappa\| \le K_0$ and $\sup_t \|\kappa - \widehat{\kappa}\| \le E_0$. Let $\widehat{h}$ and $\widehat{k}$ be the pool and normalized key driven by $\widehat{\kappa}$. Assume equal initial pool states and a common positive lower bound $h_{\min}$ on $h + \epsilon_{\text{norm}}$ and $\widehat{h} + \epsilon_{\text{norm}}$. The difference of the pool drives obeys $|\|\kappa\|^2 - \|\widehat{\kappa}\|^2| \le E_0(2K_0 + E_0)$. The pool

convolves this difference with a positive exponential whose integral is at most one. The derivative bound $|(x^{-1/2})'| \le 1/(2h_{\min}^{3/2})$ then gives

$$|h - \widehat{h}| \le E_0(2K_0 + E_0),$$
$$\|k - \widehat{k}\| \le \frac{E_0}{\sqrt{h_{\min}}} + \frac{(K_0 + E_0)E_0(2K_0 + E_0)}{2h_{\min}^{3/2}}. \tag{86}$$

This bound applies over the specified normalization gain range. Increasing heterogeneity changes the number of basis directions needed for a prescribed accuracy.

The resulting key error acts as a physical input to the relation system. Let $\widehat{M}, \widehat{p}$ be the states driven by $\widehat{k}$, and define $\Delta k = k - \widehat{k}$, $\delta M = M - \widehat{M}$, and $\delta p = p - \widehat{p}$. For identical u, v, λ, ρ, their differences satisfy

$$\delta\dot{M} = -\lambda\delta M - \rho\delta p\, k^\top + [\lambda u + \rho(v - \widehat{p})]\Delta k^\top,$$
$$\theta\delta\dot{p} = \rho(\delta M k - \delta p) + \rho\widehat{M}\Delta k. \tag{87}$$

Thus the energy in Equation (48) has derivative

$$\dot{\mathcal{E}} = -\lambda\|\delta M\|_F^2 - \rho\|\delta p\|^2 + \langle \delta M, [\lambda u + \rho(v - \widehat{p})]\Delta k^\top\rangle + \rho\delta p^\top \widehat{M}\Delta k. \tag{88}$$

The approximation error is the residual of the moving key closure. Its accumulation depends on state magnitudes, gates, and duration. This expression provides a direct comparison with the fully integrated flow. Language training uses the held key event model.

F.3 Layered processing with finite latency

Each layer can operate in a bounded local time slot. An input buffer holds its current event while an input stage forms target activities over a fixed duration. The relation stage evolves over its assigned interval and passes the resulting readout to the next layer. This schedule composes finite local responses through the hierarchy. The chosen peripheral rate model determines the remaining input and readout response errors.

Number tokens and layers from one and physical time slots from zero. With equal slot durations, layer ℓ processes logical token t in slot $t + \ell - 2$. It already retains its own state from token $t - 1$, and the preceding layer has just supplied token t. After the pipeline fills, the final layer can produce one output per slot. The first output has a delay proportional to the depth; the sustained rate is set by the slowest local stage. Extra input, readout, and communication stages contribute real delay and buffers. For depth L and layer width d_h, two sets of input and output buffers require $O(Ld_h)$ activity storage; credit history is additional.

Learning over a retained window can hold synapses fixed, collect records, reconstruct forward as needed, traverse addresses for credit, and commit the accumulated local plasticity at the window boundary. A forward window requires $O(T + L)$ slots under this schedule, with replay and reconstruction adding their explicit passes. Continuous external input can be accommodated with buffers or by alternating collection and learning intervals. This schedule specifies the causal timing and storage required by the proposed hierarchy.

G Architecture, representations, and synaptic learning

G.1 How the hierarchy processes an input

Each layer learns how an input should address, update, and read an associative memory. The key addresses the relation to be updated, the value supplies its new content, and the query asks the memory for a response. A layer forms these three activities from its input, lets its memory respond, and passes the retrieved activity to the next layer. Repeating this operation builds a hierarchy in which later memories act on features learned by earlier layers. Language prediction, semantic association, and visual classification share this processing block. Their input encodings, output classes, and times of supervision are specified in Appendix H. The recurrent populations follow Equation (25); their integrated response to one input is Equation (32).

Activity scale and transfer functions. Let t index input events, l index layers, and d be the layer width. We write the incoming row vector as x_t, suppressing the layer index until its output is passed onward. Root mean square normalization controls its amplitude. We denote this normalized vector by $\mathrm{RMS}(x)$. The sigmoid function σ and the sigmoid linear unit, SiLU, are applied componentwise. For a vector x with d components and a scalar argument s, these functions are

$$\mathrm{RMS}(x) = \frac{x}{\sqrt{d^{-1}\sum_{i=1}^{d} x_i^2 + 10^{-5}}}, \qquad \sigma(s) = \frac{1}{1+e^{-s}}, \qquad \mathrm{SiLU}(s) = s\sigma(s). \tag{89}$$

Write $h_t = \mathrm{RMS}(x_t)$ for the normalized input. Dense projections act on row vectors. Within an associative head, keys, values, queries, and traces are regarded as columns when multiplied by the memory matrix.

Forming keys, values, and queries. The key specifies the relation to update, the value supplies its content, and the query specifies the content to retrieve. For each of these three pathways, indexed by $A \in \{Q, K, V\}$, a learned matrix W_A first produces a projected activity $z_t^A = h_t W_A$. A causal filter then combines its current and two preceding values. Let c_j^A be the learned vector of filter coefficients at delay j, and let $\odot$ denote componentwise multiplication. Writing $\widetilde{z}_t^A$ for the filtered activity gives

$$\widetilde{z}_t^A = \sum_{j=0}^{2} c_j^A \odot z_{t-j}^A. \tag{90}$$

This short filter supplies local temporal context before an input reaches the associative memory. Its history resets at the same boundaries as that memory.

Each layer contains H parallel associative memories, called heads. Each head receives $K = d/H$ components of the filtered activity and maintains its own memory. For one head, write its query, key, and value as q_t, k_t, and v_t, respectively, and let $\|\cdot\|$ denote the Euclidean norm. Suppressing the head index on the filtered activity, we obtain

$$\begin{aligned} q_t &= \frac{\mathrm{SiLU}(\widetilde{z}_t^Q)}{\sqrt{\|\mathrm{SiLU}(\widetilde{z}_t^Q)\|^2 + 10^{-6}}}, \qquad k_t = \frac{\mathrm{SiLU}(\widetilde{z}_t^K)}{\sqrt{\|\mathrm{SiLU}(\widetilde{z}_t^K)\|^2 + 10^{-6}}}, \\ v_t &= \mathrm{SiLU}(\widetilde{z}_t^V). \end{aligned} \tag{91}$$

Each vector has K components, and the memory M has $K \times K$ entries. Normalization bounds query and key norms while leaving value amplitudes free. The finite response uses the actual squared key norm $k_t^\top k_t$, including the normalization constant.

Setting retention and acquisition. Two further projections determine how much the current input changes each head. Let W_α, W_β be their learned matrices and b_α, b_β their biases. The retention gate α_t and acquisition gate β_t define the nonnegative rates λ_t and ρ_t:

$$\alpha_t = \sigma(h_t W_\alpha + b_\alpha), \qquad \beta_t = \sigma(h_t W_\beta + b_\beta), \qquad \lambda_t = -\log \alpha_t, \qquad \rho_t = -\log(1-\beta_t). \tag{92}$$

These expressions give one pair of rates per head. Each input lasts one model time unit, during which its key, value, query, and rates are held constant. A HACANN head carries its memory matrix M, predicted content p, and accumulated evidence u into this interval. Their response is the finite solution of Equation (25). Thus prediction and evidence retain the effects of earlier inputs as the current relation is updated. The parameter θ sets the prediction response time relative to that of evidence and is 0.25 in all task experiments. Let M_t denote the memory at the end of the event and o_t its retrieved output. Retrieval is $o_t = M_t q_t$. The Gated DeltaNet comparison uses its canonical recurrence with the same feature projections, gates, filters, and surrounding layer.

Passing memory output to the next layer. The retrieved activity is normalized within each head. A learned K-component gain n_l is shared across the heads of layer l, and a projection W_R supplies an input-dependent output gate. Let ν index heads and let concat join them into a row vector. The combined head activity is a_t, the first residual output is r_t, and its normalized activity is f_t. With

learned projection matrices W_O, W_1, W_2, W_3, the layer computes

$$a_t = \mathrm{concat}_{\nu=1}^{H} \left[n_l \odot \mathrm{RMS}(o_t^\nu) \odot \mathrm{SiLU}\big((h_t W_R)^\nu\big) \right], \tag{93}$$

$$r_t = x_t + a_t W_O, \qquad f_t = \mathrm{RMS}(r_t), \tag{94}$$

$$x_t^{l+1} = r_t + \left[\mathrm{SiLU}(f_t W_1) \odot (f_t W_2) \right] W_3. \tag{95}$$

The first residual connection combines the incoming activity with its retrieved memory. The second applies a gated nonlinear transformation, usually called SwiGLU, with intermediate width d_f. Its output becomes the input to the next associative layer. These projections have no additional biases. After the final layer, RMS normalization and an affine output projection produce logits, whose softmax gives the response probabilities. Language input embeddings and the output projection are learned separately.

G.2 A RATE CIRCUIT FOR THE COMPLETE PROCESSING BLOCK

The hierarchy uses two kinds of population dynamics. Adaptive recurrent populations retain the associative state between inputs. Faster populations form and transform the signals that address this state. The equations above specify the stationary responses of these faster populations. Their circuit realization uses synaptic summation, opponent coding, shared gain control, causal traces, and multiplicative modulation. The same operations serve the input projection, the memory interface, the nonlinear block, and the output.

Signed activities and synaptic projections. Let a signed activity vector x be represented by two nonnegative rate vectors x^+ and x^-, with $x = x^+ - x^-$. Similarly decompose a learned matrix as $W = W^+ - W^-$, where W^+ and W^- contain its positive and negative parts. Nonnegative output drives I^+ and I^- are

$$I^+ = x^+ W^+ + x^- W^-, \qquad I^- = x^+ W^- + x^- W^+. \tag{96}$$

Their difference is xW. Excitatory and inhibitory relays provide the corresponding signed contributions while each physical output pathway keeps its sign. Biases are tonic drives represented in the same way. If the two receiving populations have activities y^+ and y^- and response time $\tau_f > 0$, their dynamics are

$$\tau_f \dot{y}^+ = -y^+ + I^+, \qquad \tau_f \dot{y}^- = -y^- + I^-. \tag{97}$$

Nonnegative initial rates remain nonnegative. With a fixed input, the stationary decoded response is $y^+ - y^- = xW$. Its deviation from this response falls by $e^{-\Delta/\tau_f}$ over duration Δ. The dense projections in all layers use this construction. Their numerical implementation evaluates the stationary response; Appendix F treats finite formation and propagation of inputs.

Normalization and transfer functions. For an activity vector with d components, let n be a shared pool of squared activity, $\varepsilon > 0$ a tonic floor, and $\tau_n > 0$ its response time. A normalization population obeys

$$\tau_n \dot{n} = \varepsilon + d^{-1} \sum_{i=1}^{d} x_i^2 - n, \qquad y_i = x_i n^{-1/2}. \tag{98}$$

Its stationary response is the RMS normalization used above. Shared inhibitory gain control motivates this population operation (Heeger, 1992; Carandini & Heeger, 2012; Wilson et al., 2012). Query and key normalization use the unaveraged squared norm and their stated floor. The square, reciprocal root, and sigmoid specify effective rate transfer functions on the operating range; the positive floor bounds the normalization gain near zero activity.

The sigmoid linear unit first obtains the bounded rate $\sigma(s)$ and uses it to modulate the signed signal s. The nonlinear block combines two projected activities by the same modulation. Conductance-dependent gain modulation (Chance et al., 2002) and models of dendritic gating (Yang et al., 2016) motivate such interactions. More generally, for signed scalar inputs $x = x^+ - x^-$ and $z = z^+ - z^-$, the positive and negative product drives are $x^+ z^+ + x^- z^-$ and $x^+ z^- + x^- z^+$, respectively. Their difference gives xz. Thus the signed products used by the memory force, output gate, and nonlinear block share an opponent realization.

Rates and temporal filters. Let a_α and a_β be the preactivations of the two gates. The physical rates can be formed directly by the nonnegative transfer laws

$$\lambda = \log(1 + e^{-a_\alpha}), \qquad \rho = \log(1 + e^{a_\beta}). \tag{99}$$

These are the rates corresponding to the sigmoid gates defined above. Each head shares its two rates across memory, prediction, and evidence populations. The three-tap filter uses the current projected activity and two retained samples. Gated traces copy these samples at event boundaries, then hold them while the next event is processed. Its learned coefficients are synaptic gains on these three pathways. This realization uses the event timing already assumed by the finite response. Continuous formation of the same drive is described in Appendix F.

Input and output. A language token activates one component of the input alphabet. Its embedding is the synaptic response of this indicator activity through the embedding matrix. Semantic messages use the same operation after tokenization. Visual inputs activate signed populations proportional to local pixel values and position coordinates. The READ event is an additional learned input pattern. Residual connections add direct and transformed input pathways at their receiving populations.

After the final layer, let ℓ_i denote the logit for output i, and let c be any common reference level. Positive output activities $r_i^{\text{out}} = \exp(\ell_i - c)$ undergo shared normalization, giving probabilities $\pi_i = r_i^{\text{out}} / \sum_j r_j^{\text{out}}$. The common reference cancels. The observed target supplies a corresponding indicator activity, and its difference from π supplies the output teaching signal for cross entropy. Language targets are the next observed bytes; semantic and visual targets are the instructed actions and class labels.

Timing and numerical realization. For each event, the input pathways form the key, value, query, and rates. These drives are held while the associative populations respond. Retrieval uses the memory at the end of that response, after which the output gate, residual pathways, and nonlinear block form the next layer's input. This ordering is repeated through the hierarchy. In the projected implementation, fast transfer functions are evaluated at their stationary responses and the associative flow is integrated analytically. In the spatial experiment, every associative and trace coordinate is replaced by its four spatial fields, with the surrounding rate operations evaluated as above. The 4,456,448-variable count refers to those spatial fields. The complete rate construction also includes the input, transformation, normalization, and readout populations described in this section.

The rate model specifies the required responses and pathways. Its normalization, gain functions, event timing, and access to adaptation traces are biological modeling assumptions. The exact circuit correspondence concerns the translation projection and its stated timing regime; spatial deviations and finite input formation are quantified in Appendices A, F, and H.1.

G.3 Information flow, dimensions, and parameter counts

Table 1 follows one event through the language model. The batch dimension indexes independently processed streams. It does not change the operations within a stream. For semantic association, a 147-component message replaces the byte embedding; for vision, a local pixel field and its position replace it. The shared layer equations retain the same order. The semantic encoder itself uses the four-layer language processing block with its own input alphabet and intent readout.

Table 1: Information flow through one event. Dimensions refer to one stream in the four-layer language model. Each layer has four heads of width 32.

Stage	Activity dimensions	Operation and outgoing information
Input	256 indicators to 128	Synaptic embedding of the observed byte.
Normalize and project	Three vectors of 128	Form query, key, and value pathways.
Temporal filtering	Three vectors of 128	Combine current activity with two retained samples per pathway.
Address memory	Four sets of 32	Normalize keys and queries; form values and two rates per head.
Associative response	$4 \times (32 \times 32 + 32 + 32)$	Update memory, prediction, and evidence for one event; retrieve each head's answer.
Combine heads	Four vectors of 32 to 128	Normalize and gate each answer; concatenate and project it.
First residual	128	Add the retrieved response to the incoming activity.
Nonlinear block	$128 \to 352 \to 128$	Normalize; form two branches of width 352; modulate, project, and add the second residual.
Next layer	128	Repeat the block through four layers.
Output	128 to 256	Normalize, form logits, and normalize positive output rates into next-byte probabilities.

Table 2 accounts for every learned forward parameter in the language comparison. The two recurrent cores share these parameters. Their different memory states are activities carried between inputs. Independently calibrated credit adds one return matrix for every dense forward projection, including the final classifier. The 41 return matrices contain 905,216 parameters. Embeddings, filter coefficients, head gains, and biases have their own local learning rules, without a second matrix representing their transpose.

Table 2: Complete language parameter count. The five square projections in each layer are query, key, value, output gate, and output mixing.

Parameter family	Per layer	Four layers / whole model
Five 128×128 projections	81,920	327,680
Three 128×352 nonlinear projections	135,168	540,672
Two 128×4 rate projections	1,024	4,096
Two four-component gate biases	8	32
Three three-tap filters of width 128	1,152	4,608
Shared head gain of width 32	32	128
All layer parameters	219,304	877,216
Input embedding, 256×128		32,768
Output projection, 128×256		32,768
Output bias, 256 components		256
All forward parameters		**943,008**
Independent feedback matrices		905,216

G.4 State, supervision, and temporal credit

Two timescales of change. Two kinds of change have different roles. Neural activity carries the relations acquired during the current input sequence. Synaptic learning changes how inputs address those relations and how their contents guide the task response. Synapses are updated after a training window or batch. Language training minimizes mean next-byte cross entropy. Semantic encoder

training classifies the intent at the end of a request; association training predicts an instructed action at each query; visual training classifies the image after all its local fields have been presented. Each supervised response supplies credit to the earlier activities that contributed to it. Here, credit denotes the teaching signal used to change a parameter or to assign an earlier activity its influence on the loss. For an analytic derivative this signal is the sensitivity of loss to the activity or parameter; learned feedback approximates that signal through a separate return pathway.

Language state and filter histories continue across 1,024-event training windows and reset at document boundaries. Temporal credit spans a complete window and stops at its boundary. For semantic messages, the encoder resets between messages, while association state continues through every rule, query, and revision within an episode. For vision, state resets between images. All parameter families are plastic within their training stage, subject to the frozen encoder and branch interventions in Appendix H.

Stored state and replay. Each layer stores two earlier projected activities for each of Q, K, V, giving $6d$ filter-history scalars. In the four-layer language model, these histories contain 3,072 scalars. Its memory matrices contain 16,384 coordinates. HACANN adds 512 prediction and 512 evidence coordinates, giving 17,408 associative coordinates, a 6.25% increase over the matrix state. Including the common filter histories, the carried state contains 20,480 scalars for HACANN and 19,456 for Gated DeltaNet, a 5.26% increase. The spatial realization of the associative coordinates is specified in Appendix H.1.

Temporal credit uses the input sequence and the activities produced during the forward calculation. The implementation stores the corresponding endpoint activities and traverses them in reverse event order. Memory and filter credit are propagated over the stated window or episode; dense interlayer credit travels through the pathway defined below. This replay storage, the synaptic parameters, and the plasticity traces are additional to the state retained between inputs. Appendix C gives the continuous and discrete credit equations, including normalization, nonlinearities, and the finite recurrent response.

G.5 Synaptic development and learning

Developing forward and feedback pathways. Each dense forward projection is initialized from correlations between independently drawn Gaussian activities at its endpoints. Let d_{in} and d_{out} be its input and output widths and $N = \max(d_{\text{in}}, d_{\text{out}})$ the number of initialization samples. The matrices $X \in \mathbb{R}^{N \times d_{\text{in}}}$ and $Y \in \mathbb{R}^{N \times d_{\text{out}}}$ have independent standard normal entries. The forward weight matrix W and separately stored feedback matrix B begin at

$$W = \frac{X^\top Y}{\sqrt{N d_{\text{in}}}}, \qquad B = 0. \tag{100}$$

The forward array has full rectangular rank almost surely. Its entry variance $1/d_{\text{in}}$ sets the initial response scale. The return pathway develops through activity calibration. Language, semantic, and visual training use this initialization. Input embeddings have independent Gaussian entries with standard deviation 0.02. Temporal filters initially pass only the current activity: their immediate coefficients are one and both delayed coefficients are zero. Head gains begin at one, retention biases at $\log 99$, and other biases at zero. All then participate in learning.

Credit through the return pathway. Let x and y be the sending and receiving activities of a dense projection, related by $y = xW$. Denote their credit signals by $\bar{x}$ and $\bar{y}$. The feedback pathway produces $\bar{x} = \bar{y}B$ at the sending population. Products of endpoint activity and credit supply local plasticity drives to the forward and feedback synapses, as derived in Appendix C. Each direction retains its own synaptic variables. The independent-credit condition uses the learned matrix B. In the exact-credit condition, dense return signals use $W^\top$ and the recurrent and nonlinear operations use their analytic derivatives. Crossing these two credit conditions with the two recurrent cores separates the effects of memory dynamics and feedback learning in the language experiment.

Local normalization of the plasticity drive. Let w be a scalar parameter, G its accumulated learning drive, s its running square trace, and η the learning rate. A subscript $+$ denotes the value after an update. The clipping operation bounds each component between its stated lower and upper

limits. Denoting the clipped drive by g, the update for embeddings, biases, head gains, and temporal filters is

$$g = \mathrm{clip}(G, -1, 1), \qquad s_+ = 0.99s + 0.01g^2, \qquad w_+ = w - \eta \frac{g}{\sqrt{s_+} + 10^{-6}}. \tag{101}$$

Dense forward and feedback synapses use the same square-trace normalization followed by the finite chemical response below.

Finite chemical response. For each dense synapse, let ε be an independent standard Gaussian draw. The noisy drive is $g = \mathrm{clip}(G(1 + 0.02\varepsilon), -1, 1)$. It updates its own square trace and holds the normalized drive $h = g/(\sqrt{s_+} + 10^{-6})$ for one learning event. Let e be its eligibility trace, τ_e the eligibility time constant, c the reaction gain, and κ_w the turnover rate. Its weight evolves through

$$\tau_e \dot{e} = h - e, \qquad \dot{w} = -c(\kappa_w w + \eta e). \tag{102}$$

Reaction gains and time constants are drawn independently for every forward and feedback synapse, uniformly from $[0.8, 1.2]$ and $[0.2, 0.8]$, respectively, and then held fixed. Time is measured in learning-event units. The analytic response over one event updates both weight and eligibility; its expression, including coincident decay rates and zero turnover, is given in Equation (63). Language and semantic learning use $\kappa_w = 0$. Vision uses the turnover rates in Appendix H.4.

Calibrating the feedback response. Calibration teaches a feedback pathway to return the activity relationships established by the forward pathway. With forward synapses held fixed, it presents independent activity probes at the two endpoints and compares their forward and return responses. The discrepancy drives feedback plasticity through Equation (58). Initial calibration develops the return pathway. Later calibration tracks changes in the learned forward pathway. The visual experiment uses the same operation to repair damaged feedback.

For a projection with input width d_{in}, let ξ and ζ be its input and output activity probes. The input probe has components $\xi_i \in \{-1, 1\}/\sqrt{d_{\mathrm{in}}}$ and the output probe has components $\zeta_j \in \{-1, 1\}$. Their scalar forward and return responses, denoted f and b, are

$$f = \xi W \zeta^\top, \qquad b = \zeta B \xi^\top. \tag{103}$$

For a fixed held-out probe set, angle brackets denote the average over probes. Let E_B be the root mean square difference between the forward and return responses, divided by the root mean square forward response:

$$E_B = \frac{\sqrt{\langle (f - b)^2 \rangle}}{\max(\sqrt{\langle f^2 \rangle}, 10^{-12})}. \tag{104}$$

Calibration runs for the prescribed number of rounds. Its final response is accepted when $E_B \leq 0.05$; nonfinite responses or the stated divergence threshold stop a run. Calibration has its own eligibility trace and random stream. Appendix H specifies probe counts, durations, and calibration intervals for each task.

H Experimental methods and complete results

The experiments connect the circuit construction to four observable capacities: maintaining predictive memory in spatial neural populations, learning from natural text, revising a relation selected by meaning, and recovering the ability to learn after feedback is disrupted. Each experiment introduces its task and comparison before specifying the implementation and results. The spatial study is detailed in Appendix H.1, language prediction in Appendix H.2, semantic revision in Appendix H.3, and visual learning in Appendix H.4.

Shared model and learning conditions. HACANN retains its prediction and evidence states between inputs. The parameter θ is the ratio of prediction to evidence response times; all task experiments use $\theta = 0.25$. A credit signal describes how a parameter or earlier activity contributes to the task loss. In the *independent-credit* condition, this signal travels between layers through separately stored feedback projections developed by activity calibration. In the *exact-credit* condition, it follows the analytic derivatives, using the transposed forward matrix at each dense projection.

Appendix G specifies the shared processing block, temporal credit, and synaptic plasticity. The language experiment compares both HACANN and canonical Gated DeltaNet (GDN) under independent and exact feedback credit. The spatial experiment continues learning from HACANN language models trained with independent feedback. Semantic and visual learning also use the independent pathway. Temporal credit spans complete 1,024-byte windows for language, complete association episodes, or complete images, as specified below.

One forward event and one learning update. An event first forms the input representation, then passes through the four operations in each layer: address the memory, let the association respond, retrieve an answer, and transform that answer for the next layer. A target supplies an error at the output events indicated in Table 3. Within an event sequence, population coordinates continue to change as information arrives. During training, reverse replay of the specified interval assigns credit to its activities; local plasticity then changes the learned projections. Evaluation holds these slow parameters fixed while the population memory continues to respond. This ordering separates the information used to make a prediction from the target used to teach it.

Table 3: Event, state, and teaching boundaries in the four studies. A reset concerns the transient memory and filters; learned synapses persist.

Study	Incoming event	State boundary	Output supervision and temporal credit
E1	Observed text byte	Spatial memory continues across windows	Next byte at every event; projected credit along the physical trajectory over 1,024 targets.
E2	Observed text byte	Continue within an article; reset at its end	Next byte at every event; credit over the current 1,024-target window.
E3	Rule, request, or revision message	Reset the encoder per message and association memory per episode	Action code at query events; association credit over the complete episode.
E4	Local pixel field, then READ	Reset before each image	Class at READ; credit over all 65 image events.

Replication and reported quantities. Language training uses seeds 1001, 1002, and 1003; the other studies use seeds 1001 and 1002. Intervention branches share the starting model for their seed and receive the same subsequent task exposure. We report individual seed values and paired differences. Means and standard deviations describe variation across trained seeds. Comparisons within a trained model retain the pairing of its articles, episodes, or images. Conditional resampling uses these complete units and therefore describes variation in the responses of the fixed models. Training exposure includes repeated byte and image presentations.

Execution. The experiments use two NVIDIA A100 GPUs, with one task per GPU. E1 integrates the spatial fields in double precision; language, semantic, and visual training use single precision for the projected recurrence. The training environment uses JAX and JAXlib 0.7.2, BrainPy 2.8.2 where used, and CUDA 12. The accompanying environment record lists the dependencies. Reported run times include computation, data decoding, calibration, checkpointing, and evaluation for the corresponding completed runs.

H.1 Spatial fields, population coordinates, and language learning

From a learned memory to a spatial circuit. The projection theory gives a physical interpretation of each memory variable: it is a displacement represented jointly by activity, adaptation, and filtered drive. We test this interpretation at the scale of a trained language model. One branch integrates the spatial populations directly; a paired branch integrates their projected coordinates. Both then continue learning from the same text. Agreement in prediction and learning tests whether the spatial

circuit preserves the computation derived from it. A second comparison perturbs the populations to determine which part of their state controls the remembered information.

The two starting models are the HACANN language models with independent feedback selected on the complete validation set in Appendix H.2. We instantiate spatial fields for every entry of the memory matrix M, prediction vector p, and evidence vector u in all four layers and four heads. Each of the 17,408 coordinates has 64 sites. Each site carries activity U, adaptation V, and two nonnegative drive populations G_+, G_-, whose difference provides the signed actuator drive. The complete associative state therefore contains 4,456,448 population variables per stream. The spatial implementation retains their forward weights in double precision and calibrates the feedback pathway through endpoint activity.

Spatial discretization and integration. Every coordinate is represented on the same periodic interval of length 12, with spacing $12/64 = 0.1875$. Adaptation strength is $m = 0.1$. The Gaussian recurrent kernel has width $a = 0.5$, inhibition $\kappa = 2/(a\sqrt{2\pi})$, and amplitude $J_0 = 3\sqrt{2}(1 + m)$. The activity, adaptation, and drive response times are $\tau_U = 1$, $\tau_V = 4$, and $\tau_g = 0.2$, respectively. The effective translation constant is $\gamma = \tau_U - m\tau_V = 0.6$. The two actuator pathways receive recurrent profiles shifted by $\delta = 0.09375$ in opposite directions. Spatial convolution is evaluated by the discrete Fourier transform, including the grid-spacing factor. The shifted profiles use the corresponding Fourier phase factors. Numerical convolution outputs are rectified at zero.

The Gaussian initial activity has unit peak and adaptation $V = mU$; both actuator populations begin at zero. Let Q denote the activity center, weighted by the cube of rectified activity, and let Z denote the joint coordinate that also includes adaptation and drive. The represented model coordinate is $16Z$, so a unit displacement in the model occupies $1/16$ of a spatial unit. During an event, keys, values, and rates are held for one model time unit. Fourth-order Runge–Kutta integration takes 32 substeps per event. The force is recomputed from the current physical field readout at each substep. The activity center and joint coordinate are calculated by the grid quadratures of Equations (11) and (13). These choices determine both the spatial trajectory and the coordinate used by the remaining network.

Matched continuation. The spatial field and event recurrence branches start with identical forward weights, feedback weights, chemical variables, and random states. Both learn from the same 32 complete windows, each containing 1,024 training targets, with neural state continuing between windows. At every input, the field branch integrates all spatial populations and reads their joint coordinates to obtain its predictions. Learning uses the translation dynamics evaluated along this physical trajectory. Credit spans the whole window. Its calculation retains the influence of a coordinate shift and sets derivatives of the remaining spatial deformation to zero. This specifies the approximation in the learning calculation; the forward calculation evolves the complete spatial fields. Both branches update all 67 forward and 41 feedback arrays. The continuation ends before the next 100-window calibration interval, so the feedback calibrated at entry supplies learning throughout these 32 windows.

Language evaluation and learning rate. Before and after continuation, each branch predicts all 8,715 targets of the first validation article. At the final endpoint, the seed 1001 branches also predict the next two complete articles, with 14,387 and 4,663 targets. Each implementation has 53,910 scored target observations across these evaluations, including repeated before/after target addresses. We report bits per byte (bpb), the average negative log probability of the correct next byte expressed in bits. For N target bytes, this is the sum of their negative log likelihoods in nats divided by $N \log 2$. Lower loss means that the model assigns more probability to the observed text.

The learning rate is $0.0003/64$, selected using development data before the continuations. Both trajectories end after 32 windows. Table 4 gives the complete first-article results. Spatial learning reduces loss by 0.010279 and 0.012492 bpb in the two seeds; its improvement agrees with the matched event implementation within 0.00008 bpb. The additional articles also give closely matched predictions after learning (Table 5). The spatial populations thus support both language prediction and the measured improvement from further exposure to text.

Table 4: Complete first-article performance at the selected step size. Every row contains 8,715 paired targets; change is after minus before.

Seed	Execution	Before (bpb)	After (bpb)	Change
1001	Spatial fields	2.355252	2.344973	-0.010279
1001	Event recurrence	2.355112	2.344901	-0.010211
1002	Spatial fields	2.367226	2.354734	-0.012492
1002	Event recurrence	2.367146	2.354732	-0.012414

Table 5: Every additional complete-article endpoint at seed 1001. Article numbers follow the fixed validation order. These are endpoint measurements after continuation.

Evaluation	Article	Targets	Field (bpb)	Event (bpb)
Final	2	14,387	2.157904	2.157890
Final	3	4,663	2.155442	2.155441

Operating range of the populations. Across the 64 field training windows, the largest visible-position fraction of the specified chart bound is 0.07993. The minimum actuator activity is 6.34×10^{-25}, and the minimum cubic mass defining the center is 1.00909. Actuator activity and cubic mass remain positive, and the visible positions occupy a small fraction of the coordinate range during learning. The deformation residual reaches 0.02411; the residual-corrected identity in the field equations has maximum absolute numerical error 1.48×10^{-16}. The deformation residual measures the part of population motion omitted by the translation approximation. The much smaller identity error measures the accuracy of the equation that includes this residual. Together with the matched prediction losses, these quantities describe both the spatial approximation and its consequence for the task. The numerical records include window extrema and synaptic changes.

Separating visible position from stored memory. The theory assigns memory to the joint coordinate Z, which includes adaptation and drive as well as the visible activity position Q. The current activity pattern can therefore move while the remembered coordinate remains fixed. Conversely, an intervention can change the remembered coordinate before a visible displacement develops. To test these two predictions, we apply currents to all 4,096 matrix coordinates in the four heads of the first layer of one fixed seed 1001 checkpoint. The intervention follows the first 1,024 inputs of the first training article in file order. Its direction is a fixed Rademacher draw with seed 90001. The prediction and evidence populations, and every other layer, undergo the matched zero-current interval. Four current designs each have dose 0 and dose 0.03. The four zero-dose labels share one sham trajectory. Each nonzero pulse has energy 0.073728, matched to within 1.89×10^{-16} relative error. The pulse lasts 0.02 model time units and is integrated in 16 Runge–Kutta steps with the relation gains closed. Denote the currents entering activity and adaptation by I_U and I_V. Pulse energy is the time and spatial integral of $I_U^2 + I_V^2$, summed over the selected coordinates. Current amplitudes are normalized to dose squared at each coordinate throughout the pulse. Small positive and negative physical probes determine the two-population response used to select the pulse direction.

Two current designs separate the coordinates: one moves visible position while preserving the joint coordinate, and the other moves the joint coordinate while preserving visible position. Two controls address U alone or V alone. We measure coordinate displacement by its Euclidean (L_2) norm across the selected populations. We also measure the Kullback–Leibler (KL) divergence from the perturbed model's next-byte probability distribution to the sham model's distribution, averaged over the following 64 targets and expressed in nats. This quantity measures how much the pulse changes the prediction. At the end of the physical pulse, we reconstruct the coordinates from the spatial fields and use these coordinates for the subsequent prediction trajectory. The pulse therefore acts on the population state, and the following behavioral readout measures the memory it leaves. The residual identity is evaluated after the external current ends.

Moving Q by 2.61×10^{-2} while holding the Z shift to 1.44×10^{-7} changes predictive KL by 3.76×10^{-12} nats. The complementary pulse moves Z by 6.16×10^{-2} while holding the Q shift

to 2.08×10^{-10}, and produces KL 8.38×10^{-4} nats. Single-population controls also change Z and subsequent predictions (Table 6). The prediction changes follow the joint coordinate. Adaptation and drive therefore contribute to the memory available to the computation even before their pending motion is expressed in the visible activity pattern.

Table 6: All eight recorded coordinate conditions. Zero-dose rows are four labels for one shared sham. Shifts use the selected-coordinate L_2 norm; predictive KL is measured in nats.

Current design	Dose	Q shift	Z shift	Predictive KL
Move Q, preserve Z	0.00	0	0	0
Move Q, preserve Z	0.03	2.61×10^{-2}	1.44×10^{-7}	3.76×10^{-12}
Move Z, preserve Q	0.00	0	0	0
Move Z, preserve Q	0.03	2.08×10^{-10}	6.16×10^{-2}	8.38×10^{-4}
U only	0.00	0	0	0
U only	0.03	3.71×10^{-2}	6.17×10^{-2}	8.42×10^{-4}
V only	0.00	0	0	0
V only	0.03	9.25×10^{-5}	6.17×10^{-2}	7.02×10^{-4}

H.1.1 Language prediction under correlated population current

Sensitivity to population fluctuations. Current fluctuations perturb the physical state that carries the learned memory. To measure their effect on computation, we apply spatially and temporally correlated current throughout a complete article and evaluate the resulting language predictions. All conditions use the same seed 1001 checkpoint with slow parameters frozen. Each trajectory predicts all 8,715 targets of the first validation article. Neural and forcing states continue across its 545 evaluation windows. Seven trajectories give 61,005 scored target observations on the same target addresses.

The population field has 64 spatial sites per coordinate and 64 integration substeps per event. Its length is 12, the width parameter is $a = 0.5$, and the dynamical constants are $\tau_U = 1$, $\tau_V = 4$, $m = 0.1$, and $\tau_g = 0.2$. One software coordinate unit corresponds to $1/16$ of a field coordinate unit, and an event lasts one model time unit. The current varies smoothly across nearby neurons and nearby times. Its correlation time is $\tau_c = 0.1$, its correlation length is $\ell = 0.5$, and it contains 16 nonzero spatial frequencies. It enters the U equation as an additive current before division by τ_U. Each coordinate has independent stationary Ornstein–Uhlenbeck coefficients for one constant mode and sine/cosine pairs. Let x, x' denote positions on the periodic interval, $q_j = 2\pi j/12$ the spatial frequency of mode j, and $w_j = \exp[-(q_j\ell)^2/2]$ its spectral weight. Let $C(x, x')$ denote the covariance between positions x and x', normalized to one at coincident positions. It is

$$C(x, x') = \frac{1 + 2\sum_{j=1}^{16} w_j \cos[q_j(x - x')]}{1 + 2\sum_{j=1}^{16} w_j}. \tag{105}$$

For current intensity D, the pointwise variance is D/τ_c, and the temporal covariance between times t, t' is proportional to $e^{-|t-t'|/\tau_c}$. Coefficients are sampled using the exact Ornstein–Uhlenbeck transition at each forcing knot, and the current between knots is interpolated linearly. We use intensities $D \in \{0, 10^{-8}, 10^{-6}\}$, with forcing seeds 93001 and 93002 at each nonzero intensity and a matched zero-current trajectory.

Five principal trajectories use 64 linear-interpolation intervals per event. For seed 93001 at each nonzero intensity, a second trajectory uses 128 intervals from the same underlying 128-interval forcing path. This paired comparison measures sensitivity to the finite interpolation of the prescribed current. The checkpoint and article remain fixed across all conditions. Forcing seeds describe current realizations within this one trained model.

Results and interpretation. At $D = 10^{-8}$, the two principal trajectories add 0.005979 and 0.005050 bpb to the zero-current article loss. At $D = 10^{-6}$, they add 0.527332 and 0.518468 bpb (Table 7). The smaller current produces a modest prediction penalty, while the larger current substantially disturbs the learned computation. Doubling forcing interpolation resolution changes

mean article loss by 0.0001365 and 0.0001502 bpb, respectively. Targetwise changes are larger and partly cancel in the article average; both absolute and signed summaries are retained in Table 8. These measurements relate the imposed population fluctuations to the change in language loss at the stated intensities and finite forcing resolutions.

Table 7: All complete current trajectories. Each row uses the same 8,715 targets; the difference is relative to the matched zero-current article loss. Time is GPU-job elapsed time.

D	Forcing seed	Intervals	bpb	Δ bpb	Time (min)
0	93001	64	2.355252	+0.000000	19.19
1.00×10^{-8}	93001	64	2.361232	+0.005979	22.98
1.00×10^{-8}	93002	64	2.360302	+0.005050	23.23
1.00×10^{-6}	93001	64	2.882584	+0.527332	22.98
1.00×10^{-6}	93002	64	2.873721	+0.518468	22.90
1.00×10^{-8}	93001	128	2.361368	+0.006116	23.40
1.00×10^{-6}	93001	128	2.882735	+0.527482	23.15

Table 8: Both full-article forcing-resolution comparisons at seed 93001. The signed difference is 128 minus 64 interpolation intervals. Mean and maximum absolute differences use all 8,715 paired target likelihoods in bits.

D	Signed mean	Mean absolute	Maximum absolute
1.00×10^{-8}	+0.0001365	0.0043976	0.2029027
1.00×10^{-6}	+0.0001502	0.0546078	1.2301054

H.2 Language prediction and the use of document context

Comparing the memory dynamics and the learning pathway. Natural language prediction requires the memory to retain information from earlier text and use it to anticipate what follows. We evaluate this ability on WikiText-2. The model reads UTF-8 bytes and predicts the next byte at every position, yielding a 256-symbol prediction task on the original text. The comparison changes two components independently: the recurrent core, HACANN or canonical Gated DeltaNet (GDN), and the return pathway used for learning, independent or exact credit. Comparing cores under the same credit condition tests the memory dynamics. Comparing credit conditions within a core tests whether calibrated feedback supports learning. All four conditions share the surrounding language architecture. All four conditions use four layers of width 128, four heads, feedforward width 352, and a 256-byte input and output alphabet. Each model has 943,008 forward parameters; the independent pathway stores a further 905,216 feedback parameters. The processing block, optimization settings, and training exposure are shared. HACANN carries finite prediction and evidence states, while GDN uses its canonical recurrence. Independent credit travels through separately stored feedback synapses; exact credit uses the corresponding exact derivatives.

Data, state boundaries, and selection. The raw WikiText-2 corpus (Merity et al., 2017) is partitioned at its top-level article headings. Every source row is preserved, and a line feed is appended only when the row lacks one. Text is encoded as UTF-8 bytes. The first byte of each article supplies context; every subsequent byte is a prediction target. The training split contains 629 articles and 10,927,167 targets per complete traversal. Each batch lane advances through complete articles, continuing across training windows. Article order is shuffled with the run's seed at the start and at each new traversal.

Each model learns next-byte prediction for 768 updates, with batch size 64 and complete 1,024-byte credit windows. This gives 50,331,648 target-byte presentations per run and 603,979,776 across twelve runs. These counts include repeated passages. State continues within each natural document and resets at its boundary; temporal credit is truncated at complete window boundaries.

Validation at updates 256, 512, and 768 scores all 60 validation articles, containing 1,145,487 targets. The earliest minimum validation loss selects the checkpoint; update 768 is selected in all twelve runs. Every selected model then scores all 62 test articles and 1,289,061 targets. Training and checkpoint selection use their respective training and validation splits. We aggregate negative log likelihood over targets before converting to bpb, so each byte has equal weight regardless of article length.

Optimization and feedback calibration. The learning rate is 3×10^{-4}. The local square-trace accumulator has decay 0.99, denominator constant 10^{-6}, and elementwise clipping to $[-1, 1]$. Dense reciprocal projections use the finite chemical dynamics in Appendix G, with separate feedback, eligibility, chemical, and random states.

Independent feedback starts at zero and receives 7,500 initial calibration rounds, with 128 probes per round, probe gain 0.1, refresh duration 0.02, eight chemical substeps, and eligibility time constant 0.5. Every 250 rounds, 512 held-out probes measure the normalized response error in Equation (104). The final response is accepted at threshold 0.05. A response ratio of 100 is the divergence stopping threshold. Tracking calibration uses 500 rounds every 128 training updates. The exact-credit arms provide the matched reference and incur no calibration time.

Complete test results. Table 9 gives the test loss of every trained model. HACANN with independent feedback reaches mean 2.24548 bpb, compared with 2.25075 bpb for GDN with exact credit. HACANN with exact credit and GDN with independent feedback reach 2.24979 and 2.24928 bpb, respectively. Across three seeds, the four means lie within 0.00527 bpb. The two recurrent cores learn comparably predictive representations under this matched model size and training exposure. Their similar outcomes under both credit conditions also show that the calibrated return pathway supports learning in each core. Table 9 gives the variation across training seeds; article-level results retain the shared test passages.

Table 9: Complete final WikiText-2 bits per byte. Lower values indicate better prediction. The independent unit is a trained seed; the final column gives the arithmetic mean and sample standard deviation across the three seeds.

Model / credit	Seed 1001	Seed 1002	Seed 1003	Mean $\pm$ SD
HACANN / independent	2.24307	2.27000	2.22338	2.24548 ± 0.02340
HACANN / exact	2.24386	2.27417	2.23133	2.24979 ± 0.02203
GDN / independent	2.29725	2.23455	2.21604	2.24928 ± 0.04256
GDN / exact	2.28985	2.23936	2.22304	2.25075 ± 0.03483

Dependence on document context. A recurrent memory can carry information across the whole document. We measure which earlier inputs affect these trained models by holding the scored passage fixed and changing the history used to construct its incoming state. This tests the use of context separately from the overall test loss. We take the first 20 eligible anchors in fixed article order, beginning at byte position 4,097 in each article and advancing by 4,096 bytes while 64 target positions remain. The anchors span six test articles. Every condition scores the same 64 targets after each anchor, so HACANN-independent and GDN-exact receive the same 1,280 targets in each seed. We restrict the available history to the nearest 64, 256, 1,024, or 4,096 bytes and compare with full document history. A separate intervention replaces the earlier natural-text prefix while preserving the nearest 1,024 inputs. Every anchor has an eligible donor from a different article, with length between one half and twice the recipient length and enough bytes for the required prefix. The closest eligible article length determines the donor, with article identifier resolving ties.

Restricting HACANN-independent to 64 bytes raises mean loss by 0.019477 bpb; retaining 256 bytes reduces the penalty to 0.002458 bpb. GDN-exact gives corresponding penalties of 0.026092 and 0.002469 bpb. With 1,024 or 4,096 bytes, every absolute seed-level effect is below 0.00015 bpb in both models. Replacing the remote prefix changes loss by less than 0.00029 bpb in absolute value. At these anchors, both trained models draw the measured benefit of context mainly from recent text.

Memory removal and reconstruction from input. Clearing a state component immediately before the scored targets tests its contribution to the next predictions. Clearing the memory matrix M raises mean loss by 0.461047 bpb in HACANN-independent and 0.496928 bpb in GDN-exact. Clearing HACANN prediction, evidence, or short filter histories separately changes mean loss by 0.003068, 0.002639, or 0.004724 bpb. The memory matrix makes the largest measured contribution at these anchors. Prediction and evidence participate in forming and updating that matrix as the input arrives.

To test recovery, we clear each component before replaying the nearest 1,024 inputs. The resulting loss changes are much smaller: recent input can reconstruct the state needed for these predictions. Parameters remain fixed throughout both interventions. These interventions identify two functions of the dynamics: retaining an already formed predictive state and rebuilding it as relevant input arrives. The large immediate effect of matrix clearing establishes its role in prediction; recovery through replay shows that its useful contents can be formed again from recent text. Table 10 gives every condition for each training seed.

Table 10: Natural-history interventions: loss change from full history (bpb). Each condition uses the same 1,280 targets per trained model. Positive values indicate worse prediction. The prediction and evidence states belong to HACANN.

Condition	Seed 1001	Seed 1002	Seed 1003
HACANN with independent feedback			
64-byte history	+0.016557	+0.012746	+0.029128
256-byte history	+0.001767	+0.003106	+0.002503
1,024-byte history	-0.000071	+0.000007	-0.000085
4,096-byte history	-0.000027	+0.000049	+0.000016
Replace remote prefix	-0.000054	-0.000164	+0.000286
Clear M immediately	+0.449182	+0.503762	+0.430197
Clear p immediately	+0.006889	+0.002998	-0.000684
Clear u immediately	+0.000895	+0.000458	+0.006565
Clear filters immediately	+0.007694	+0.005600	+0.000877
Clear M, replay 1,024	-0.000005	-0.000056	-0.000132
Clear p, replay 1,024	-0.000023	-0.000068	+0.000043
Clear u, replay 1,024	-0.000017	-0.000056	+0.000039
Clear filters, replay 1,024	+0.000121	-0.000010	+0.000043
Gated DeltaNet with exact credit			
64-byte history	+0.017146	+0.025095	+0.036034
256-byte history	+0.001237	-0.003098	+0.009268
1,024-byte history	+0.000069	+0.000148	+0.000004
4,096-byte history	-0.000007	+0.000009	-0.000031
Replace remote prefix	+0.000012	+0.000128	-0.000110
Clear M immediately	+0.494477	+0.515605	+0.480701
Clear filters immediately	+0.002781	+0.008760	-0.000088
Clear M, replay 1,024	+0.000050	+0.000112	+0.000065
Clear filters, replay 1,024	+0.000007	-0.000012	-0.000062

Computational cost. Average completed-run times are 26.43 minutes for HACANN-independent, 22.90 minutes for HACANN-exact, 21.89 minutes for GDN-independent, and 18.69 minutes for GDN-exact. Calibration contributes 2.63 minutes per independent run. Peak GPU allocations are 17,105, 17,099, 16,991, and 16,989 MiB, respectively. Per-seed training, calibration, validation, and other costs are supplied in the result tables. These measurements include the finite-response computation, explicit temporal credit, and the stated 1,024-byte training configuration.

H.3 Semantic grounding, temporary binding, and selective revision

Revising a relation selected by meaning. A familiar request can require a new response when an instruction changes. The meaning of the request stays the same; its current relation to an action

changes. This task brings together the two operations in the proposed memory: meaning retrieves an existing association, and new evidence revises the association it addresses. BANKING77 supplies natural customer requests from 77 banking intents. During each episode, temporary instructions assign a small set of these intents to action codes and later revise some assignments. A learned encoder represents request meaning, while the associative circuit maintains the current instructions. HACANN and GDN receive the same semantic representations within each training seed.

Real requests and the semantic encoder. We use all 77 BANKING77 classes (Casanueva et al., 2020), with 8,972 training requests and 987 development requests under the fixed grouped split. The 3,080 official test requests supply the held-out request pool. A byte-level byte-pair encoding tokenizer has 2,048 tokens. It is trained on the training requests, with minimum token frequency two. Each seed trains its own HACANN semantic encoder using independent feedback. The encoder has four layers, width 128, four heads, feedforward width 352, and maximum segment length 128.

The semantic encoder learns to classify all 77 intents. Semantic learning uses 2,000 updates at batch size 32 and learning rate 3×10^{-4}, giving 64,000 example presentations per seed. Complete development accuracy is measured at updates 500, 1,000, 1,500, and 2,000. The highest accuracy selects the model, with the earlier checkpoint resolving an exact tie. This selects update 1,500 for seed 1001 and update 2,000 for seed 1002, with 826/987 (83.69%) and 801/987 (81.16%) correct development classifications. These classification scores describe the 77-way semantic training objective. Association learning receives the continuous hidden activity and learns to compare and bind requests within each episode. Initial calibration uses 7,500 rounds and 128 probes; tracking uses 500 rounds every 100 updates. Probe gain 0.1, refresh duration 0.02, eight substeps, eligibility time constant 0.5, held-out checks every 250 rounds with 512 probes, response acceptance 0.05, and divergence response ratio 100 are shared with the language calibration protocol.

What the association circuit receives. The semantic encoder processes the sentence in every rule, revision, and query. Its final normalized activity has 128 continuous components and is scaled by $128^{-1/2}$. This activity supplies a learned similarity structure through which differently worded requests can address the same relation. Rules and revisions also provide an observed 16-dimensional action code. Three indicators identify the message type. For a query, the action input is zero and the model predicts its current action code. The circuit receives the continuous sentence activity and the message-type indicator; intent labels are used to construct and score the episodes.

The resulting 147 components pass through a learned projection to width 128, then a two-layer, four-head association core with feedforward width 256 and a 16-way output. The encoder resets for every message. The association state continues across messages and resets at the episode boundary. Keys, values, queries, gates, and output weights are learned. For each seed, cached 128-dimensional activities reproduce its fixed encoder's response, giving the HACANN and GDN cores identical semantic inputs. Intent labels determine episode construction and the correct evaluation action. The association circuit receives the continuous encoder activity, message type, and instructed action code described above.

Episode structure and the need for selective revision. A main episode introduces four rules, each assigning an action code to a distinct intent, and then asks one real query for each intent. A new instruction revises one rule, followed by the same four queries. A second instruction revises a different rule, followed by all four queries again. The sequence is therefore: four rules, four questions, one revision, four questions, a second revision, and four questions. The episode contains 18 association events and twelve responses: four initial answers, two answers testing the current revision, and six answers testing protection of other relations. One protected answer concerns a relation revised earlier in the episode. The four query requests retain their wording across phases, while their presentation order is shuffled in each phase. The appropriate answer therefore changes only when its rule changes. Model parameters remain fixed within an episode; neural state carries the temporary assignments and their revisions.

We distinguish generalization to a new combination of familiar relations from generalization to a new individual relation. Each intent has ten codes allowed during training, three reserved for development, and three reserved for final evaluation. In the *unseen mappings* suite, individual intent–code pairs are allowed during training, but a fixed hash rule partitions complete supervised mappings in proportions 80%, 10%, and 10% for training, development, and test. This rule applies after every

revision as well as to the initial map. The *unseen bindings* suite uses the disjoint intent–code pairs assigned to each split; pairwise disjointness also separates these maps from the training maps. All sixteen outputs are available at every query. Mapping construction accepts the first proposal that satisfies the mapping constraints, with at most 20,000 search nodes per proposal and 32 problem proposals. The proposal fixes its intents, request texts, and revision targets before searching for action codes. The first valid assignment is accepted. The included generator specifies the hash salts and action pools, so each episode is determined by its seed and split constraints.

Table 11: The complete main semantic episode. The same four query requests reappear after each revision, with their presentation order reshuffled. The second revision changes a different relation.

Events	Information arriving	Answers measured
1–4	Establish four intent–code relations	Instructions supply the current assignments.
5–8	Query all four intents	Four initial answers.
9	Revise one assignment	Instruction supplies its new action code.
10–13	Query all four intents	One revised answer and three protected answers.
14	Revise a second assignment	Instruction supplies its new action code.
15–18	Query all four intents	One revised answer and three protected answers, including the earlier revision.

The episode contains eighteen messages and twelve supervised answers. An instruction makes its action code available as input. A query presents the request with the query indicator; the required code supplies its training target or evaluation label. The continuous semantic activity, action-code indicator, and message indicator together form the 147-component association input described above. Thus the query tests the relation retained from earlier instructions. The four initial, two revised, and six protected answers account for every scored event in the main episode.

Association learning. Each core trains for 2,048 updates at batch size 64 and learning rate 3×10^{-4}, totaling 131,072 episodes. The first 256 updates introduce two relations and one revision; the remaining 1,792 use four relations and two revisions. Credit spans the complete episode, with one chemical commit after the complete batch. Initial calibration uses 7,500 rounds and subsequent calibration uses 1,500 rounds every 128 updates. Probe counts, gains, duration, time constant, and acceptance settings follow those above; calibration reaction gain is one. The final update is the evaluation endpoint.

Within each seed, HACANN and GDN share the trained encoder, semantic activity, association initialization, and episode stream. The two seeds have separately trained encoders and cores. Evaluation holds slow parameters, feedback, chemical state, and calibration fixed.

Evaluation and metrics. Each model is evaluated on 512 main episodes, divided equally between unseen mappings and unseen bindings. Consecutive episode seeds begin at 96,110,000 and 96,120,000 for the two suites, respectively. The same episodes are used for both cores and both trained seeds. Each suite therefore contains 1,024 initial answers, 512 current-revision answers, 1,536 protected answers, and 256 answers protecting an earlier revision. Complete-episode success requires all twelve answers to be correct. Table 12 reports each core and seed.

HACANN mean answer accuracy is 94.43% for unseen mappings and 95.12% for unseen bindings. Accuracy on the current revision is 93.55% and 95.02%, while protection of other answers is 94.66% and 95.15%. Complete-episode success is 72.85% and 75.98%. The paired GDN cores achieve comparable response accuracy. Reading these measures together shows how the memory uses meaning: a request selects the relation whose answer changes, while the remaining relations continue to guide their respective answers.

Table 12: Every main semantic outcome (percent). Initial, revision and protection refer to answer accuracy; earlier refers to protection of a previously revised relation; whole requires every answer in an episode.

Core	Seed	Initial	Revision	Protection	Earlier	Whole
Unseen complete mappings						
HACANN	1001	95.12	93.55	95.12	92.58	73.83
HACANN	1002	93.95	93.55	94.21	92.97	71.88
GDN	1001	95.90	93.75	96.09	93.36	78.12
GDN	1002	94.14	93.95	93.88	92.19	71.48
Unseen intent–code edges						
HACANN	1001	95.02	93.95	95.25	95.31	74.61
HACANN	1002	95.21	96.09	95.05	96.48	77.34
GDN	1001	95.31	94.53	95.51	94.92	78.12
GDN	1002	94.73	95.70	93.55	95.70	75.00

Behavioral reference policies. Revision and protection make different demands on a memory. A policy that always returns the last instructed code achieves 100% current-revision accuracy, 0% protection, and 25% overall accuracy. A policy given oracle intent access that retains the initial map achieves 75% overall accuracy and misses every current revision. Both have zero complete-episode success. These response patterns explain why we measure revision and protection alongside overall accuracy. A static ridge readout fitted from continuous semantic activity to the actual training query supervision reaches 10.61% and 9.41% overall accuracy on unseen mappings, and 0.42% and 0.13% on unseen bindings in seeds 1001 and 1002. This reference fits sixteen action scores to the 128 semantic features and an intercept using all actual training-query targets. Ridge regularization has strength 0.001 on the mean squared objective, with an unpenalized intercept and double-precision solution. Its fixed semantic-to-action map receives no current episode rules. The association cores combine semantic knowledge with the temporary instructions supplied in the episode.

Paired changes to codes, rules, and meaning. Paired counterfactuals test which parts of an instruction control the answer. On the first 64 main episodes in each suite, we permute public action codes, change one revision code, remove revisions, or change the meaning of a query. Each intervention preserves the other public content. Some query positions therefore require a changed answer, while others retain their original answer. Table 13 reports accuracy on both subsets. A changed target is any query whose required action differs after the intervention. A protected target belongs to the original episode's protection set and retains its required action after the intervention. The subsets therefore address different questions and together need not cover every query.

HACANN follows a changed revision code on 96.08–97.89% of affected answers and retains 93.48–97.17% accuracy on eligible protected answers. Changing query meaning gives 89.06% to 95.31% accuracy across suites and seeds. Selecting a different relation through meaning is therefore more demanding in these comparisons than several direct changes to action codes. The paired tables also report the fraction for which both the original and changed answers are correct.

Table 13: Every counterfactual cell in both generalization suites. Each pair of percentages is seed 1001 / seed 1002; n_C, n_P are the changed and unchanged protected answer counts per seed. Counts refer to eligible responses within each trained model.

Core	Intervention	n_C	Changed (%)	n_P	Protected (%)
Unseen complete mappings					
HACANN	Permute codes	713	94.11 / 94.81	30	93.33 / 100.00
HACANN	Change revision	95	96.84 / 97.89	353	93.48 / 97.17
HACANN	Remove revision	192	93.23 / 96.35	320	95.62 / 97.81
HACANN	Change meaning	64	89.06 / 95.31	349	94.27 / 96.85
GDN	Permute codes	713	94.81 / 94.11	30	93.33 / 100.00
GDN	Change revision	95	91.58 / 95.79	353	96.60 / 94.90
GDN	Remove revision	192	94.79 / 95.83	320	96.88 / 94.69
GDN	Change meaning	64	90.62 / 93.75	349	95.70 / 95.13
Unseen intent–code edges					
HACANN	Permute codes	717	93.58 / 94.70	27	96.30 / 100.00
HACANN	Change revision	102	96.08 / 97.06	346	93.93 / 94.51
HACANN	Remove revision	192	96.35 / 96.88	320	92.81 / 93.44
HACANN	Change meaning	64	92.19 / 90.62	353	92.07 / 93.77
GDN	Permute codes	717	93.58 / 92.61	27	100.00 / 100.00
GDN	Change revision	102	96.08 / 97.06	346	94.22 / 91.04
GDN	Remove revision	192	97.40 / 97.92	320	92.81 / 91.88
GDN	Change meaning	64	92.19 / 95.31	353	94.05 / 90.65

Transformation under code permutation. A permutation of the action codes should produce the corresponding permutation of the model's answers. We measure this relation as prediction equivariance. Under unseen mappings, HACANN gives 97.92% and 97.14%, and GDN gives 97.66% and 96.61%. Under unseen bindings, HACANN gives 94.40% and 95.70%, and GDN gives 95.57% and 94.27%. Each value uses 768 paired answers. Equivariance measures whether the answer follows the transformation; answer accuracy measures whether it agrees with the requested action. A consistently transformed answer can still be incorrect, so both measures are reported.

Memory load. Increasing the number of relations tests how the learned update operates as more answers are maintained and revised. The fixed models receive 128 episodes with two relations and one revision, and 128 with eight relations and four revisions, each divided equally between the two generalization suites. These are termed localization and pressure conditions in the result tables. The smaller episodes ask four questions, and the larger episodes ask forty. Consecutive episode seeds begin at 96,210,000 and 96,220,000 for the smaller condition and at 96,310,000 and 96,320,000 for the larger condition, in suite order. The models receive these loads with their trained parameters fixed. In the eight-relation condition, HACANN answer accuracy averages 91.78% for unseen mappings and 89.28% for unseen bindings; complete-episode success is 36.72% and 21.09%. Here, success requires forty correct answers, compared with twelve in the main task. The drop in complete-episode success combines the increased relation load with the greater number of opportunities for a semantic or associative error. Table 14 gives both seeds for every load condition.

Table 14: Accuracy as the number of relations changes (percent). Each row uses 64 episodes. The two-relation condition has one revision and four answers per episode; the eight-relation condition has four revisions and forty answers. Whole requires every answer to be correct.

Core	Seed	Initial	Revision	Protection	Whole
Two relations, unseen mappings					
HACANN	1001	97.66	95.31	96.88	92.19
HACANN	1002	95.31	93.75	98.44	89.06
GDN	1001	94.53	93.75	95.31	85.94
GDN	1002	96.88	95.31	100.00	93.75
Two relations, unseen bindings					
HACANN	1001	96.88	90.62	98.44	89.06
HACANN	1002	97.66	95.31	100.00	93.75
GDN	1001	96.88	90.62	100.00	89.06
GDN	1002	95.31	92.19	98.44	87.50
Eight relations, unseen mappings					
HACANN	1001	92.58	88.67	91.35	34.38
HACANN	1002	91.41	91.80	92.52	39.06
GDN	1001	92.19	89.06	92.19	43.75
GDN	1002	92.19	92.58	92.75	43.75
Eight relations, unseen bindings					
HACANN	1001	89.26	90.62	87.95	17.19
HACANN	1002	90.43	92.97	89.56	25.00
GDN	1001	89.26	89.84	88.34	23.44
GDN	1002	90.43	90.62	89.62	23.44

Paired responses and variation across episodes. The original and counterfactual evaluations yield 17,920 responses per fixed model and 71,680 across the four models. The 296-row paired table provides complete-episode bootstrap intervals and leave-one-content-out sensitivity for core comparisons. Each resample keeps all queries in an episode together. The intervals therefore describe variation across episode contents for the fixed trained pair. Four cells have no eligible target and remain undefined. Both training seeds are reported individually throughout.

H.4 Visual learning and restoration of feedback

Feedback as a pathway for learning. A model's current predictions and its capacity to learn from their errors depend on different pathways. Forward activity produces the predictions; feedback carries teaching signals to earlier representations. We test this distinction after a hierarchy has learned eight visual classes. The forward pathway is held fixed while feedback connections are rearranged. A second branch receives the same disruption followed by calibration with unlabelled activity. The branches then learn all ten classes from identical image exposure. Their subsequent learning tests the functional role of the feedback correspondence. A branch that updates only the output classifier measures the contribution of learning throughout the hierarchy.

Images and class partition. The task uses all ten CIFAR-10 classes (Krizhevsky, 2009). A fixed split within each class divides the 50,000 official training images into 45,000 training and 5,000 development images, with 4,500 and 500 per class. The 10,000-image official test has 1,000 images per class. Pixel-content hashes are disjoint across all three splits. A class-name hash rule assigns airplane and dog as novel classes, leaving eight old classes. Prediction always uses ten outputs, so learning the novel classes requires distinguishing them from the old animals and vehicles within the same classifier. Development data come from the official training set, and both seeds are evaluated at fixed epochs.

From pixels to a sequence of local fields. The input is the original 32×32 RGB image, transformed by $2(\mathrm{RGB}/255) - 1$. Training uses reflection padding by four pixels, a seeded 32×32 crop,

and a horizontal flip with probability one half. Development and test use a single original view. The model learns its input features directly from these pixels.

During old-class learning, an 8×8 grid of nonoverlapping 4×4 local fields covers the image. During mixed learning, each field expands to 6×6 at stride four using reflection padding of the same image. The packed 48×48 array contains these 64 overlapping local fields. Two fixed position coordinates accompany each field, and a learned READ activity follows the 64 fields, giving 65 events per image. Four HACANN layers have width 128, four heads, feedforward width 352, and $\theta = 0.25$. Recurrent memory and temporal filters reset between images; synapses and local chemical states continue through learning.

Credit and chemical dynamics. Supervision at the ten-way readout supplies credit for all 65 image events. This credit reaches earlier representations through independently stored feedback projections. Dense forward and feedback synapses have separate square and eligibility traces, gains in $[0.8, 1.2]$, time constants in $[0.2, 0.8]$, and multiplicative drive noise 0.02. Their local chemical dynamics include turnover $\kappa_w = 0.1\eta$ during old learning and 0.3η during mixed learning, where η is the current learning rate. Other parameter families use the ordinary local square-trace update. Task credit supplies the learning drive; endpoint activity supplies the calibration drive.

Old-class learning. Seeds 1001 and 1002 start independently. Each receives 40 epochs over 36,000 old-class images, totaling 1,440,000 presentations and 22,520 updates at batch size 64, including each final 32-image batch. An update-wise cosine schedule reduces the learning rate from 3×10^{-4} to 3×10^{-5} over 30 epochs, then restarts at 10^{-4} and reduces it to 3×10^{-5} over ten epochs. Optimizer, feedback, chemical, and random states continue through the schedule restart. Initial activity calibration uses 7,500 rounds; tracking uses 500 rounds every 128 updates.

Matched feedback interventions. The complete old model for each seed supplies four branches: intact feedback, damaged feedback, repaired feedback, and readout-only learning. When the input fields expand, their learned central contacts are copied and the added outer contacts start at zero. All branches consequently retain the same predictions at the start of mixed learning. Feedback damage and repair also leave those predictions unchanged: both act while forward synapses are fixed. Differences that develop under later image exposure therefore measure the effect of the learning pathway.

Damage permutes the rows and columns of each of the 42 feedback matrices, using permutations with no fixed points. The training seed and projection identity determine each permutation; the damaged and repaired branches receive identical permutations. This operation preserves each matrix's collection of entries, norm, and singular values while changing which endpoints it connects. Forward parameters and task chemistry remain fixed during injury. The repair branch then receives 7,500 rounds of unlabelled activity calibration, with 128 probes per round and forward parameters fixed. Repair uses the normalized held-out response error of Equation (104), with 512 probes per projection and acceptance threshold 0.05. Probe gain, refresh duration, substeps, and calibration time constant are 0.1, 0.02, eight, and 0.5, respectively. Calibration reaction gain is one, held-out responses are checked every 250 rounds, and the divergence response ratio is 100. Readout-only learning freezes all hidden forward, feedback, and chemical variables, stored in 583 arrays, and adapts the classifier with its own chemical state.

Shared mixed learning. All branches receive the same image order, pixels, and initial task random state. Each trains for 32 epochs over all 45,000 training images, totaling 1,440,000 presentations and 22,528 updates, including eight-image final batches. The first two epochs warm the learning rate linearly from 2×10^{-5} to 3×10^{-4}; the next 30 use cosine decay to 10^{-5}. Augmentation draws use epoch indices 0–31. Mixed learning proceeds without periodic feedback calibration. Full-learning branches adapt forward and feedback synapses. Their comparison with readout-only learning therefore includes both learning the additional local input contacts and adapting representations deeper in the hierarchy.

Evaluation. Development accuracy is measured at entry and every five epochs during old-class learning, then at entry and every epoch during mixed learning. The resulting 282 measurements cover both seeds and every branch. Fixed checkpoints are old epoch 40 and mixed epoch 32. After

training, the ten endpoints are evaluated on the 10,000 complete test images. Overall, novel-class, and old-class accuracy use 10,000, 2,000, and 8,000 images, respectively; every prediction remains ten-way. Activity comparisons use matched probes from the shared entrance model with expanded input fields. Retention is measured relative to the old-stage test predictions.

Table 15: Final CIFAR-10 test accuracy (percent). Denominators are 10,000 overall, 2,000 novel and 8,000 old images. Every endpoint uses ten-way prediction.

Seed	Branch	All	Novel	Old
1001	Shared old	63.410	0.000	79.263
1001	Intact	80.580	77.800	81.275
1001	Repaired	80.730	78.350	81.325
1001	Damaged	72.960	75.400	72.350
1001	Readout only	69.060	49.100	74.050
1002	Shared old	63.360	0.000	79.200
1002	Intact	80.920	79.500	81.275
1002	Repaired	80.890	78.550	81.475
1002	Damaged	72.090	71.350	72.275
1002	Readout only	69.090	48.550	74.225

Restored responses and restored learning. Damage raises the median normalized response discrepancy across the 42 feedback projections to 1.404 and 1.411 in the two seeds. Unlabelled repair reduces these medians to 2.512×10^{-4} and 2.502×10^{-4}. The largest post-repair discrepancies are 0.01213 and 0.01210. Every projection is measured with the same held-out probe construction and its discrepancy is normalized by its forward-response RMS. Figure 5 shows how the feedback responses recover their correspondence with the forward responses within each trained circuit.

After the same mixed learning, repaired models reach 80.73% and 80.89% overall accuracy, close to intact models at 80.58% and 80.92%. Damaged models reach 72.96% and 72.09%. Repair thus increases overall accuracy by 7.77 and 8.80 percentage points. On the same test images it corrects 1,362 and 1,407 errors and introduces 585 and 527 errors, respectively. The net improvement is present for both novel classes (2.95 and 7.20 points) and old classes (8.975 and 9.200 points). Repaired-minus-intact differences are $+0.15$ and -0.03 points.

The intact hierarchy exceeds readout-only learning on novel classes by 28.70 and 30.95 points. Its old-class accuracy also rises from approximately 79.2% at the old endpoint to 81.275% in both seeds; repaired models reach 81.325% and 81.475%. Learning throughout the hierarchy therefore helps acquire new visual knowledge and refine previously learned representations. The repaired circuit recovers this ability under the same subsequent task exposure, linking correspondence of the feedback response to its function in learning.

The course of learning. All four branches begin with identical development accuracy: 63.80% and 63.74% in the two seeds. After one mixed-class epoch, damaged models reach 35.52% and 32.58%, while repaired models reach 65.44% and 66.96%. The feedback intervention thus changes the course of learning from a shared initial prediction. Figure S7f summarizes the paired benefit over the full learning interval. For each seed and class group, we integrate repaired-minus-damaged development accuracy by the trapezoidal rule over integer epochs 0–32 and divide by 32. In percentage points, this quantity gives the average accuracy benefit during learning.

Resources and paired measurements. The ten training stages consume 9.8530 GPU hours, and official test evaluation consumes 0.0168 GPU hours. These costs include calibration and the overhead within each run. The results include complete per-class counts, old-class retention, paired image outcomes, learning curves, and all five paired branch comparisons. Seed means and observed ranges describe the two independently trained starting models; image-level comparisons describe their paired responses. These comparisons link feedback correspondence to a behavioral capacity. A hierarchy can preserve its predictions at the time of disruption while its subsequent learning deteriorates. Calibration restores the return responses and the ability to learn useful representations.

The comparison with a fixed hidden hierarchy further shows how learning at the sensory input and deeper layers contributes to acquiring new visual classes.

H.5 Measured quantities and figure construction

Each quantitative panel is computed from the observations at its stated experimental unit. Figure 2a uses the paired first-article validation measurements before and after continuation: two seeds, two implementations, and two endpoints give eight observations. Lines connect endpoints within an implementation and seed. Panel b reads the four pulse conditions at their common final time, giving one displacement of Q and one of Z for each condition. Panel c retains every prediction divergence for the same 64 subsequent targets in each condition. Its 256 points describe responses of one trained checkpoint to four interventions. Horizontal position within a condition only separates overlapping marks. Short horizontal lines give arithmetic means across targets; both displacement and divergence use logarithmic axes.

Figure 3a shows all twelve complete-test outcomes and the mean within each core and feedback condition. Panels b and c use the same three seeds for each of the two intervention arms. The six combinations of core and removed state give eighteen observations in panel b. The four retained-history lengths give twenty-four observations in panel c. Each observation first aggregates the loss change over the same twenty positions and 1,280 target bytes. Means then average these seed outcomes equally. Connecting lines in panel c link the four measured history conditions; the horizontal axis is logarithmic and the loss axis is linear.

Figure 4b places each policy's revision accuracy against its preservation accuracy. Two trained seeds and two transfer suites give four points per policy, hence twenty paired outcomes across the five policies. The inset enlarges the region occupied by the eight learned-memory outcomes. Coincident reference outcomes keep their measured positions. Panel c shows all twenty-four complete-episode outcomes across two cores, two seeds, two suites, and three loads. Each horizontal span connects the two observed seeds for a core and suite. These spans describe the observed seed range. The shaded row identifies the four-relation training condition.

Figure 5b retains each feedback pathway's response discrepancy at the three intervention stages; its emphasized trajectories are medians across the 42 pathways within a seed. Development trajectories in panel c follow the recorded epoch endpoints. Panel d reports each seed's complete test accuracy. Throughout the figures, connected observations preserve their paired seed, checkpoint, or condition identity. The supplied plotting programs and source tables specify the observations and transformations for every panel.

I Neural interpretation and experimental predictions

HACANN asks how a circuit that maintains a value can become part of a system that maintains knowledge of relations. Recurrent excitation and inhibition sustain a localized activity pattern. Adaptation and incoming drive determine how that pattern moves. Connections between populations organize these movements so that a cue retrieves associated content and directs its revision. The resulting memory has both a physical state and a computational meaning. This appendix develops that connection and its experimental consequences. Appendix A derives the population coordinate; Appendices B–E derive its associative dynamics and learning rules.

I.1 From maintaining a variable to maintaining a relation

A continuous attractor stores a value by allowing a population pattern to occupy a continuous range of positions. Recurrent interactions restore the shape of the pattern after a small disturbance. Translation changes its position while preserving the recurrent balance. An input along this direction leaves a displacement that can remain after the input ends. The same circuit can therefore retain a value and integrate successive changes to it (Amari, 1977; Seung, 1996; Zhang, 1996; Wu et al., 2008; Fung et al., 2010).

Navigation circuits provide concrete examples of this organization. An experimentally imposed activity bump can persist in the fly heading network, and offset pathways convey movement signals that shift the represented heading (Kim et al., 2017; Turner-Evans et al., 2017; 2020). Mammalian

head direction and grid cell populations exhibit ring and toroidal structure across behavioral conditions (Chaudhuri et al., 2019; Gardner et al., 2022). These observations connect the geometry of a represented variable to the collective dynamics of a neural population.

HACANN extends this organization to the representation of a relation. Let M denote a matrix mapping a cue to associated content. Its entry M_{ij} specifies how cue component j contributes to content component i. One adaptive population maintains each entry. Reading many coordinates together gives the answer associated with a cue; moving those coordinates changes the answer. Thus a persistent population state can act as an adjustable association coefficient. The recurrent connections within each population provide persistence. Connections between populations make the retained coefficients work together as a mapping.

This division also separates two timescales of memory. Within an episode, motion of the attractor coordinates changes the currently stored associations. Across episodes, synaptic learning shapes the representations through which experiences address those associations. The former changes what a cue predicts. The latter changes which experiences use the same cue and which content their predictions contain. A single population mechanism can support both the stored relation and the prediction and evidence signals used to revise it.

These roles distinguish the geometry that supports memory from the contents currently remembered. Each population has a family of possible persistent states supplied by its recurrent connections. An experience changes where the population lies in that family. Across many populations, the resulting coordinates determine the current associative mapping. The attractor family can therefore remain available while the relations expressed by its occupied states change from one context to the next. This is the sense in which the same circuit can participate in different computations.

Sensory associations with a stable attractor can themselves change through experience. Visual input to the fly heading circuit is remapped by altered sensorimotor experience, and correlated visual and compass activity can establish a heading representation in a new scene (Fisher et al., 2019; Kim et al., 2019). HACANN develops a corresponding computational distinction between the persistence supplied by recurrence and the meaning supplied by learned coupling.

The proposal that CANNs are canonical modules (Wu et al., 2016) acquires a specific computational content here. The repeated circuit preserves a coordinate and moves it in response to a directed input. A set of these circuits preserves an association; a hierarchy composes associations. This is a claim about a reusable operation, in the sense that normalization can recur across neural systems with different represented variables (Carandini & Heeger, 2012). Its biological test is whether populations carrying different task variables share the predicted modes of persistence and motion, and whether their connections organize those modes into retrieval and correction.

I.2 Circuit operations and their biological realization

The construction uses a population rate description. In Equation (4), the convolution $J*r$ is recurrent excitation, and the shared denominator in the firing law describes divisive inhibition. Normalization provides a common account of response gain in sensory circuits (Heeger, 1992); activation of cortical inhibitory populations can alter that gain in vivo (Wilson et al., 2012). The activity variable U, adaptation variable V, and filtered drive g evolve with their respective response times. The recurrent kernel is translation invariant over the coordinate range used for memory, and its shape modes are assumed to relax around the supported bump family.

The pathway that moves a bump uses two copies of the recurrent projection shifted in opposite directions. An imbalance between them generates a directed displacement. Splitting this signed drive into two nonnegative populations permits each output pathway to retain a fixed excitatory or inhibitory sign. Offset connectivity in the fly heading system supplies a physiological precedent for moving an internally maintained representation (Turner-Evans et al., 2017; 2020). In HACANN, the shifts and their normalization specify the relation between the input force and the resulting coordinate velocity.

Associative coupling adds a second operation: one population signal modulates the gain of another. Conductance changes can modulate the gain of pyramidal neurons (Chance et al., 2002), and models of dendritic disinhibition explain selective gating of input pathways (Yang et al., 2016). The model represents such modulation by products of nonnegative rates over a bounded operating range. Oppo-

nent populations carry signed variables, while fixed output signs implement the required excitatory and inhibitory contributions. Readout uses weighted population sums, local rate transformations, and normalization. These are the effective circuit operations from which the population construction is assembled.

The input to a memory population shows how the operations combine. Let $Z_{ij} = M_{ij}$ be its maintained coordinate; let k_j be cue component j; and let v_i, p_i, and u_i be the incoming content, current prediction, and evidence trace for content component i. The rates ρ and λ regulate acquisition and turnover, respectively. The required force is

$$F_{ij} = -\lambda Z_{ij} + \big[\lambda u_i + \rho(v_i - p_i)\big] k_j. \tag{106}$$

The coordinate supplies its own turnover term. The bracketed signal is shared across a value row, and the cue component is shared down a key column. Each population receives only the signals needed to change its contribution to the prediction. Summation across columns then reads the updated content.

The two rates have distinct neural meanings. Increasing ρ accelerates the response of evidence, prediction, and the correction driven by their discrepancy. Increasing λ moves memory toward the association carried by the evidence trace. With no evidence, this term produces decay; with continuing evidence, it also writes that evidence into memory. Their integrated effects give the retention and correction coefficients in Appendix B. A gate therefore controls how much of a neural response unfolds during the presentation of an input.

The adaptation signal must also be available to the memory readout. When an adaptive inhibitory population carries V, its output can reach both the recurrent field and the decoder. For intracellular adaptation, a separate causal trace supplies the corresponding accessible signal. Appendix A derives both realizations. Their physiological signatures include the availability of this trace, the signs and offsets of the motion pathways, and the gain dependence of the resulting displacement.

I.3 Reading a memory while activity is changing

The current center of an activity bump does not contain the entire state of an adaptive population. Drive already received can still be acting, and adaptation can still reflect an earlier position. Even after an external force ends, these processes can continue to move the activity. The memory available during this transient includes the displacement that the present state will produce as it relaxes.

The translation projection identifies this memory. Let ψ_0 be the normalized sensitivity to a displacement around a stationary bump. For perturbations δU, δV, and δg, define their projected components by $q_U = \langle \psi_0, \delta U \rangle$, $q_V = \langle \psi_0, \delta V \rangle$, and $q_g = \langle \psi_0, \delta g \rangle$, where brackets denote spatial integration. Let τ_U, τ_V, and τ_g be the activity, adaptation, and drive time constants; let m be adaptation strength; and set $\gamma = \tau_U - m\tau_V > 0$. The change in the joint memory coordinate is

$$\delta Z = \frac{\tau_U}{\gamma} q_U - \frac{\tau_V}{\gamma} q_V + \tau_g q_g. \tag{107}$$

For unforced linear relaxation, this quantity equals the position change remaining after the other motion components have decayed. The coefficients follow from the circuit time constants. Reading Z therefore accounts for motion already implied by the neural state, including motion that is still absent from the visible activity center.

This result gives adaptation a role in the content of a transient memory. The displacement of adaptation relative to activity determines part of the remaining motion. Filtered drive contributes the displacement due to input whose effect is still unfolding. Adaptive continuous attractors use these same interactions for tracking and anticipative motion (Mi et al., 2014; Li et al., 2025). Here they determine the variable available to associative computation. Equation (5) extends the readout to the current population profile, and Appendix A gives its residual when that profile deforms.

Persistent information carried by changing neural activity has been studied in recurrent circuit models and observed in prefrontal population recordings (Druckmann & Chklovskii, 2012; Murray et al., 2017). Models of selective integration distinguish the direction of persistent motion from the population readout that extracts it (Mante et al., 2013; Pagan et al., 2025). HACANN specifies that readout through the activity, adaptation, and drive dynamics. Prospective variables in Latent Equilibrium address the related problem of reading a computation before slow neural responses

have settled (Haider et al., 2021). The attractor projection supplies an explicit expression for this quantity in the present circuit.

A physiological test would estimate the three response times and the adaptation strength from independent perturbations. These measurements fix the coefficients of the joint readout. On held-out trajectories, it should predict the eventual retained position and subsequent retrieval. Its difference from the activity center should diminish as adaptation catches up and drive relaxes. This time dependence tests the mechanism with a constrained decoder.

Causal perturbations offer a second test. A change in activity can be offset by adaptation or drive, preserving Z while moving the visible center Q. Another perturbation can preserve Q while changing Z. Subsequent retrieval should follow the displacement of Z. E1 implements these interventions with equal-energy currents in the full spatial system and reads their consequences over the following 64 targets. The comparison makes the physical carrier of predictive memory experimentally accessible.

The broader idea that memory is distributed across interacting neural states also appears in models of synaptic working memory (Mongillo et al., 2008). HACANN gives this idea a particular form: recurrent activity supports the persistent coordinate, and the state of adaptation and drive determines how that coordinate is read during motion. Simultaneous measurements of these variables would test the relation between an observed transient and the memory it leaves behind.

I.4 Prediction error changes a remembered relation

An associative memory supports a simple cycle of inference and revision. A cue evokes the content that past experience associates with it. Incoming content can agree with that prediction or reveal a discrepancy. The cue then directs the discrepancy back to the association that generated the answer. The memory after this response contains a revised expectation for that cue.

This organization gives the Delta rule its neural interpretation. In the instantaneous prediction regime, the predicted content is Mk and the discrepancy is $v - Mk$. The prediction-error contribution to an update vanishes when the answer already matches the incoming content. Its magnitude grows with the part of the content left unexplained by the current association. The cue determines where this discrepancy is written. With finite prediction time, the population variable p follows Mk, and the circuit uses the developing discrepancy $v - p$.

Predictive coding supplies a related account of how predictions and residual errors organize neural computation. In the model of Rao & Ballard (1999), feedback conveys predictions of lower-level activity and feedforward pathways convey their residuals. In HACANN, an associative prediction determines the residual, and that residual revises the relation held in population state. The connection concerns a specific operation: experience changes memory according to what the current memory leaves unexplained. The cue specifies which expectation produced that discrepancy and where to return it.

The residual also contains information about what was unexpected. Each of its components specifies a content feature and a direction of correction. A scalar novelty signal can regulate the correction rate; the vector discrepancy determines the change in the predicted content. Models with several inhibitory cell types show how excitation and inhibition can support populations selective for different signed prediction errors (Hertäg & Clopath, 2022). Recordings in mouse auditory cortex identify neurons selective for particular violations of learned sound expectations (Audette & Schneider, 2023). HACANN assigns such feature information a further computational role through its product with the cue. Equal-magnitude errors in different content directions should drive different memory changes, while changing the cue should redirect the associations affected by the same error.

The correction has a transparent consequence when the cue and incoming content are held fixed. In the instantaneous prediction regime with zero turnover, let $e = v - Mk$ be the content discrepancy. Differentiating this expression using the memory dynamics gives

$$\dot{e} = -\rho\|k\|^2 e, \qquad \frac{\mathrm{d}}{\mathrm{d}t}\frac{\|e\|^2}{2} = -\rho\|k\|^2\|e\|^2. \tag{108}$$

The retrieved answer approaches the supplied content at a rate set by acquisition and cue strength. The error supplies its own stopping condition: as the answer improves, the force that changes it

decreases. Finite prediction time, evidence persistence, and turnover give the temporal responses derived in Appendix B.

This distinction between fast association and learned representation explains how an experience can have an immediate, selective effect. Changing the content associated with a cue changes the memory consulted on the next query, with fixed feature projections. Across many episodes, synaptic learning shapes those projections so that the same correction applies to the experiences that should share it. Prediction and representation learning thus operate on different aspects of the same relation.

I.5 Cue geometry determines what a correction changes

The effect of a new experience extends to other queries according to their overlap with its cue. Let q be a query, $o(q) = Mq$ the answer before an update, $\Delta o(q)$ its change, and β the strength of an isolated Delta correction. Then

$$\Delta o(q) = \beta(v - Mk)(k^\top q). \tag{109}$$

The content discrepancy determines the direction of change in the answer. The overlap $k^\top q$ determines how much of that change reaches another query. An orthogonal query preserves its answer under this correction. A query aligned with the cue receives the full correction. The complete HACANN response also includes retention and continuing traces, as derived in Appendix B.

The rule follows from the way memory produced the answer. Entry M_{ij} contributes to answer component i in proportion to cue component k_j. Returning the error through k_j changes that entry according to the sensitivity of the answer to it. For instantaneous prediction, this is the negative gradient of the squared discrepancy $\frac{1}{2}\|v - Mk\|^2$ with respect to the matrix. Each gradient component is realized as the motion of a maintained population coordinate.

There is also a precise sense in which this update changes only what the present correction requires. Take a unit key and let $d = \beta(v - Mk)$ be the desired change in its answer. Every matrix change A satisfying $Ak = d$ can be written as $A = dk^\top + B$, where $Bk = 0$. Write $\|\cdot\|_F$ for the Frobenius norm of a matrix. The two terms are orthogonal in its associated inner product, giving

$$\|A\|_F^2 = \|d\|^2 + \|B\|_F^2. \tag{110}$$

The Delta correction $dk^\top$ therefore has the smallest Frobenius norm among changes that produce the specified answer. Additional movement in B changes other cue directions without helping the current correction. This statement concerns the associative correction at fixed representations. Retention and continuing evidence contribute their own specified changes in the complete recurrence.

The same geometry determines generalization and interference. Two expressions benefit from overlapping cues when their answers should be revised together. They benefit from distinct cue directions when their assignments should change independently. Population geometry has been linked to abstraction and generalization in hippocampus and prefrontal cortex (Bernardi et al., 2020). HACANN assigns that geometry a dynamical role: it determines which future memories a new experience changes. E3 tests this role with different natural-language expressions of customer intentions and changes to their assigned actions.

This interpretation also clarifies the connection to cognitive maps. A relational map specifies how one represented situation gives access to another item or outcome. Models such as the Tolman–Eichenbaum Machine relate shared structure across experiences to spatial and relational generalization (Whittington et al., 2020); a related account connects Transformer computation to hippocampal representations (Whittington et al., 2022). In HACANN, the map is the collection of cue-to-content relations carried by the populations. Their joint coordinates describe its current contents, and learned cue geometry determines how it is revised. Learning a representation therefore includes learning which experiences should remain linked when circumstances change. This interpretation gives the map an operational meaning: it is specified by the answers it retrieves and by how those answers change after a new experience.

The meaning of an individual coordinate depends on its role in this relation. For an orthogonal change of key basis represented by matrix R, define transformed keys, queries, and memory by

$$k' = Rk, \qquad q' = Rq, \qquad M' = MR^\top. \tag{111}$$

Retrieval, the Delta rule, and Equation (6) are preserved, with unchanged prediction, evidence, and value. The computational relation thus survives a change in the coordinate labels. Its physical implementation still requires each population to remain within its operating range, so coordinate scale also determines the attainable precision and cost.

Equation (109) suggests a quantitative test. Cue overlap can be varied while holding the value discrepancy and update duration fixed. The resulting change in a second answer should follow that overlap. Holding overlap fixed while changing prediction time or input duration measures the additional effect of neural timing. A pathway intervention that changes the cue used for writing while preserving the cue used for retrieval should redirect the correction. Representations estimated independently of the tested changes provide the quantities needed for these predictions.

I.6 Response times determine the history carried into an update

Prediction and evidence take time to form. During one input interval, prediction approaches the answer retrieved from memory and evidence approaches the incoming value. At the next interval, each can retain part of its previous response. The integrated recurrence describes how the current association, unfinished prediction, and recent evidence jointly determine the next memory.

For a held key k, only the memory component $y = Mk$ interacts with the prediction population. Let θ be its response time relative to evidence, and let s denote a temporal growth or decay rate of the homogeneous subsystem. Its characteristic equation is

$$\theta s^2 + (\theta\lambda + \rho)s + \rho(\lambda + \rho \|k\|^2) = 0. \tag{112}$$

Evidence adds a relaxation rate $-\rho$. At zero turnover and a unit key, $\theta = 1/4$ gives a repeated decay rate -2ρ. This is the reference value used in the trained models. Changing the prediction response time changes the temporal course of correction while the cue continues to determine its direction.

Continuing evidence introduces a measurable dependence on history. Consider two states with the same entering matrix but different evidence traces, whose difference is Δu. In the instantaneous prediction limit, let $\alpha = e^{-\lambda\Delta}$ and $\beta = 1 - e^{-\rho\Delta}$ be retention and acquisition over a duration Δ, for a unit key. The resulting difference in their next memories is

$$\Delta M_+ = (1 - \alpha)(1 - \beta)\Delta u\, k^\top. \tag{113}$$

Prior evidence matters more when turnover is stronger and less when current acquisition is nearly complete. Thus a trace perturbation should interact with input duration and with the two gains in a specified way. E2 measures the importance of the trace at the learned operating point; controlled manipulation of these quantities would test the full relation.

Finite response also explains the extra state in the recurrence. Along the current cue, memory interacts with one prediction vector and one evidence vector. Appending these two vectors to the memory matrix gives an augmented transition whose correction has rank at most three. Its coefficients are fixed by the response equations. Clearing the incoming evidence and taking instantaneous prediction recovers canonical Gated DeltaNet. A separately gated decay interval followed by a correction interval supplies the other construction in Appendix B. The experiments retain the continuing traces and their finite response.

The gates admit a further temporal prediction. Multiplying both λ and ρ by a common positive factor, while dividing the presentation interval by that factor, preserves the final projected state for held representations. Varying either rate alone changes the balance between turnover, correction, and evidence persistence. This relates an input-dependent gate to the speed of an underlying response and gives a direct gain–duration comparison. At the level of the full field, the result also depends on how well the population follows its projected dynamics.

Modulatory control of when experience changes a circuit has a physiological precedent in the fly heading system. Dopamine release during orienting movements alters the later influence of visual cues (Fisher et al., 2022). HACANN gives such control explicit computational roles: acquisition and turnover gains regulate the ongoing association, while plasticity gains regulate the slower changes in its representation.

I.7 PERSISTENCE, PRECISION, AND STABILITY

Restoring the shape of a population pattern and preserving its represented value are different dynamical requirements. A perturbation toward a deformed shape can relax back to the attractor family. A perturbation along that family changes the retained value. Projection methods determine how a spatial input contributes to each direction (Wu et al., 2008; Fung et al., 2010). The relation between neural variability and reported position in working memory has supported a diffusing bump account of memory precision (Wimmer et al., 2014). In HACANN, this precision concerns the joint coordinate carried by activity, adaptation, and drive.

Let Σ be the covariance matrix of the three projected perturbations (q_U, q_V, q_g) in Equation (107), and let c collect their readout coefficients. Their contribution to readout variance is

$$\mathrm{Var}(\delta Z) = c^\top \Sigma c, \qquad c = \left(\frac{\tau_U}{\gamma}, -\frac{\tau_V}{\gamma}, \tau_g\right)^\top. \tag{114}$$

The vector c contains the readout coefficients fixed by the response times. The effect of adaptation noise depends on its covariance with activity. As γ decreases, the readout becomes more sensitive to errors in their weighted difference. A test can vary adaptation strength, estimate the changed response times, and measure the resulting memory precision. This links persistence to the dynamics through which a population approaches its remembered state.

At the level of associative coupling, retrieval and correction also have a stability consequence. Consider two trajectories sharing inputs, gains, and evidence, with memory difference δM and prediction difference δp. Define their quadratic difference energy by $\mathcal{E} = \frac{1}{2}\|\delta M\|_F^2 + \frac{\theta}{2}\|\delta p\|^2$. Then

$$\dot{\mathcal{E}} = -\lambda \|\delta M\|_F^2 - \rho \|\delta p\|^2 \le 0. \tag{115}$$

The cue carries the memory response to prediction and returns the prediction discrepancy to memory. These reciprocal contributions cancel in the energy derivative. The same coupling that specifies which relation changes also prevents growth of this difference energy.

This gives a causal prediction about pathway matching. If retrieval uses cue k_r and writing uses cue k_w, the energy derivative acquires the term $\rho\, \delta p^\top \delta M (k_r - k_w)$. Its sign depends on the mismatch and current state. Altering the correspondence between retrieval and writing pathways should therefore change both the direction of revision and the transient stability. Differences in evidence add a decaying forcing, as derived in Appendix B; comparisons of complete learned hierarchies also involve their changing representations and gains.

The continuous family of stationary states follows from the recurrent symmetry specified by the model. Structural deviations can produce a slowly evolving manifold that supports memory over finite intervals (Ságodi et al., 2024). The field experiments perturb currents and measure departures from the motion projection with a shared kernel. Manipulating the recurrent kernel would additionally measure drift due to structural asymmetry and its effect on retention.

Population size and coordinate range determine the cost of precision. E1 represents a software coordinate by a physical displacement sixteen times smaller. This keeps trajectories within the selected spatial chart and amplifies physical readout noise by sixteen when expressed in software units. The observed noise response consequently refers to this implementation scale. Population size, chart scale, and recurrent kernel can each change the relation between precision and resource use.

The number of reliably stored associations depends on key dimension, overlap, forgetting, and value scale. E3's eight-relation condition increases both memory load and the number of correct answers required for complete episode success. Varying these factors independently would connect storage capacity to the geometry and physical precision of the maintained coordinates.

I.8 LEARNING A HIERARCHY THROUGH RECIPROCAL ACTIVITY

A hierarchy uses the answer from one associative memory to help form the cues and content of another. The same local persistence and correction mechanism can therefore participate in successively different computations. Synaptic learning organizes these transformations and their gates.

Context-dependent integration in prefrontal cortex provides a related example of recurrent dynamics operating on learned representations (Mante et al., 2013). HACANN specifies both the memory retained within each layer and the signals used to learn its connections.

Forward projections determine representations. Separate feedback projections convey the effect of an output error toward preceding populations. Their correspondence is learned by comparing responses to activity applied at the endpoints. Let W be a forward projection and B its feedback projection. Let ξ and ζ be independent zero-mean row-vector probes, with covariances $\Sigma_\xi = \mathbb{E}[\xi^\top \xi]$ and $\Sigma_\zeta = \mathbb{E}[\zeta^\top \zeta]$. Define their scalar response discrepancy as $\delta = \zeta(W^\top - B)\xi^\top$. The mean calibration drive is

$$\mathbb{E}[\zeta^\top \delta \xi] = \Sigma_\zeta (W^\top - B)\Sigma_\xi. \tag{116}$$

Isotropic probes give a restoring direction toward reciprocal action. Restricted probe covariance calibrates the directions sampled by the activity. The diversity of endpoint activity thus determines which feedback functions can be learned or repaired.

This rule makes a causal prediction about learning. Feedback damage can preserve the current forward computation while disrupting subsequent acquisition. Calibration with fixed forward synapses can restore the ability to learn. E4 tests this sequence: damage feedback, repair it with unlabeled probes, and resume learning from matched entering states. Varying the subspace sampled by probes at fixed total probe energy would further test whether recovery follows the error directions represented during repair.

Activity-based reciprocal learning (Akrout et al., 2019), feedback alignment (Lillicrap et al., 2016), and spike-based alignment (Gierlich et al., 2026) provide related accounts of communication between forward and feedback pathways. Dendritic models separate forward drive from teaching signals (Guerguiev et al., 2017; Sacramento et al., 2018), and eligibility traces retain local activity relevant to later plasticity (Bellec et al., 2020). HACANN uses endpoint activity, a shared discrepancy signal, and finite eligibility responses to calibrate the feedback pathways of its hierarchy.

Learning from a delayed outcome also requires access to the preceding events. The implementation retains the input sequence and forward state information, then traverses the recorded interval in reverse event order to assign temporal credit. Appendix E describes how replay with checkpoints can reconstruct the required states and gives its memory and computation requirements. Reverse sequence replay has been observed in hippocampal populations (Foster & Wilson, 2006). A circuit realization of the present learning algorithm additionally requires the address sequence, reconstructed activities, and error signals specified in that appendix.

Other dynamical accounts of temporal credit use prospective states and dendritic response dynamics to approximate an adjoint during ongoing computation (Ellenberger et al., 2026; Brandt et al., 2026). HACANN uses calibrated reciprocal pathways and replay of a recorded interval. Its temporal credit range is consequently determined by the retained history, replay duration, and response of the learning circuit. These quantities supply observable costs and timescales for the proposed learning mechanism.

I.9 Relation to other accounts of neural and artificial memory

Fast-weight models describe rapidly changing associations that retrieve recent experience (Ba et al., 2016). Linear attention makes this retrieval a recurrent matrix computation (Katharopoulos et al., 2020), and the Delta rule changes an association according to its prediction error (Schlag et al., 2021). Parallel evaluation has made these recurrent memories practical for language modelling (Yang et al., 2024); Gated DeltaNet adds control of retention (Yang et al., 2025). Qwen3-Next combines Gated DeltaNet with full attention (Qwen Team, 2025). Kimi Linear uses Kimi Delta Attention, with finer control of forgetting, together with Multi-Head Latent Attention (Kimi Team, 2025). These architectures establish an application for revisable associative memories in large language models.

HACANN derives this computational family from maintained population coordinates. Finite prediction and evidence add two continuing vectors, whose interaction with memory fixes the coefficients of the integrated update. Gated DeltaNet-2 separately controls erasure and writing (Hatamizadeh et al., 2026); HACANN derives its additional state from the finite response of the neural circuit. The

canonical Gated DeltaNet rule is obtained in the regimes specified in Appendix B. The language experiments evaluate the continuing finite response and its learned hierarchy.

Attention has several established connections to associative neural computation. Modern Hopfield networks derive attention retrieval from an energy function (Ramsauer et al., 2021), and sparse distributed memory provides an account of similarity-based access (Bricken & Pehlevan, 2021). Neuron–astrocyte models implement attention through interactions at tripartite synapses (Kozachkov et al., 2023). Astrocyte gating of associative memory gives a related dynamical realization (Vivet & Arenas, 2026). Transient Hebbian potentiation can implement the selective weighting of presynaptic inputs (Ellwood, 2024). Controlled Dynamics Attractor Transformer combines attention, Hopfield energies, and attractor-inspired excitation and inhibition for iterative inference (Zhang et al., 2026). Continuous fast-weight models formulate changing associations through differential equations (Irie et al., 2022). The present construction connects associative correction to the collective translation of adaptive neural fields and uses that relation to identify the physical state that carries predictive memory.

Spiking language models explore temporal coding and learning at a different level of neural description. SpikeGPT trains a recurrent spiking architecture with surrogate gradients (Zhu et al., 2024), while SpikeLLM adapts pretrained language models through saliency-based spiking and quantization (Xing et al., 2025). HACANN develops the population rate mechanism of memory and revision. A spiking realization and scaling studies would connect this mechanism to additional neural and computational constraints.

Continuous attractors can also generate internal trajectories through asymmetric recurrent interactions (Spalla et al., 2021). In HACANN, successive cues and discrepancies move a collection of coordinates so that later queries encounter a revised relation. The local circuit supplies persistence, its coupling determines the revision, and learned projections determine how that relation participates in a hierarchy. This connects the dynamics of a neural population to the organization of memory used by a language model.

J SUPPLEMENTARY FIGURES

Individual outcomes and supporting measurements for the four experiments follow.

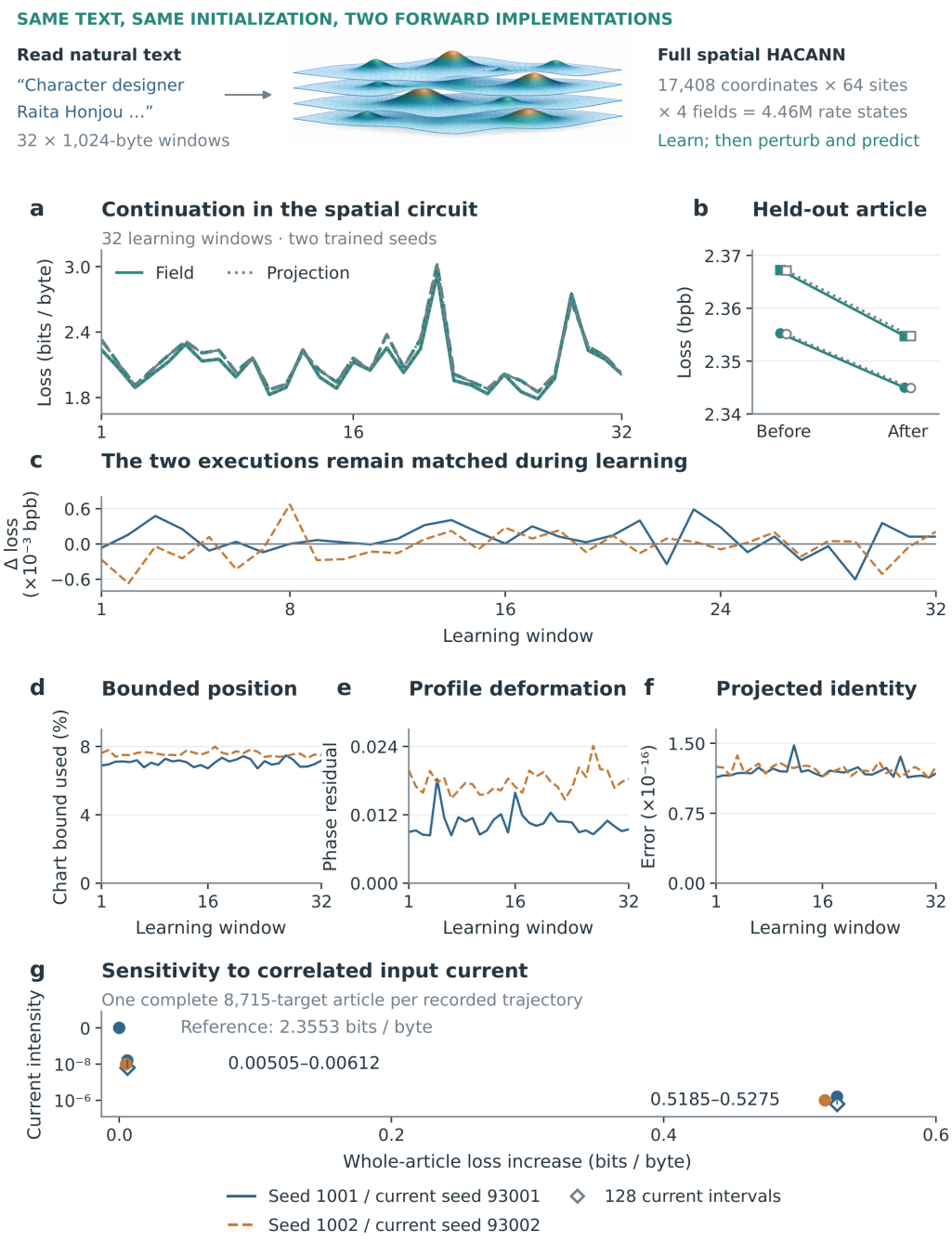

Figure S1: **Learning and the operating range of the complete spatial circuit.** (a) All 32 matched continuation windows per trained seed. (b) The same held-out article before and after learning, scoring 8,715 targets. (c) Field minus projected loss at every window. (d–f) Maximum chart occupancy, deformation residual, and residual-corrected identity error. (g) Every article outcome under correlated current, relative to zero current; labels give observed ranges. Circles and diamonds distinguish 64- and 128-interval realizations. Panels (a–f) retain two trained seeds. Panel (g) uses one checkpoint, two forcing seeds, and a paired finer realization for forcing seed 93001.

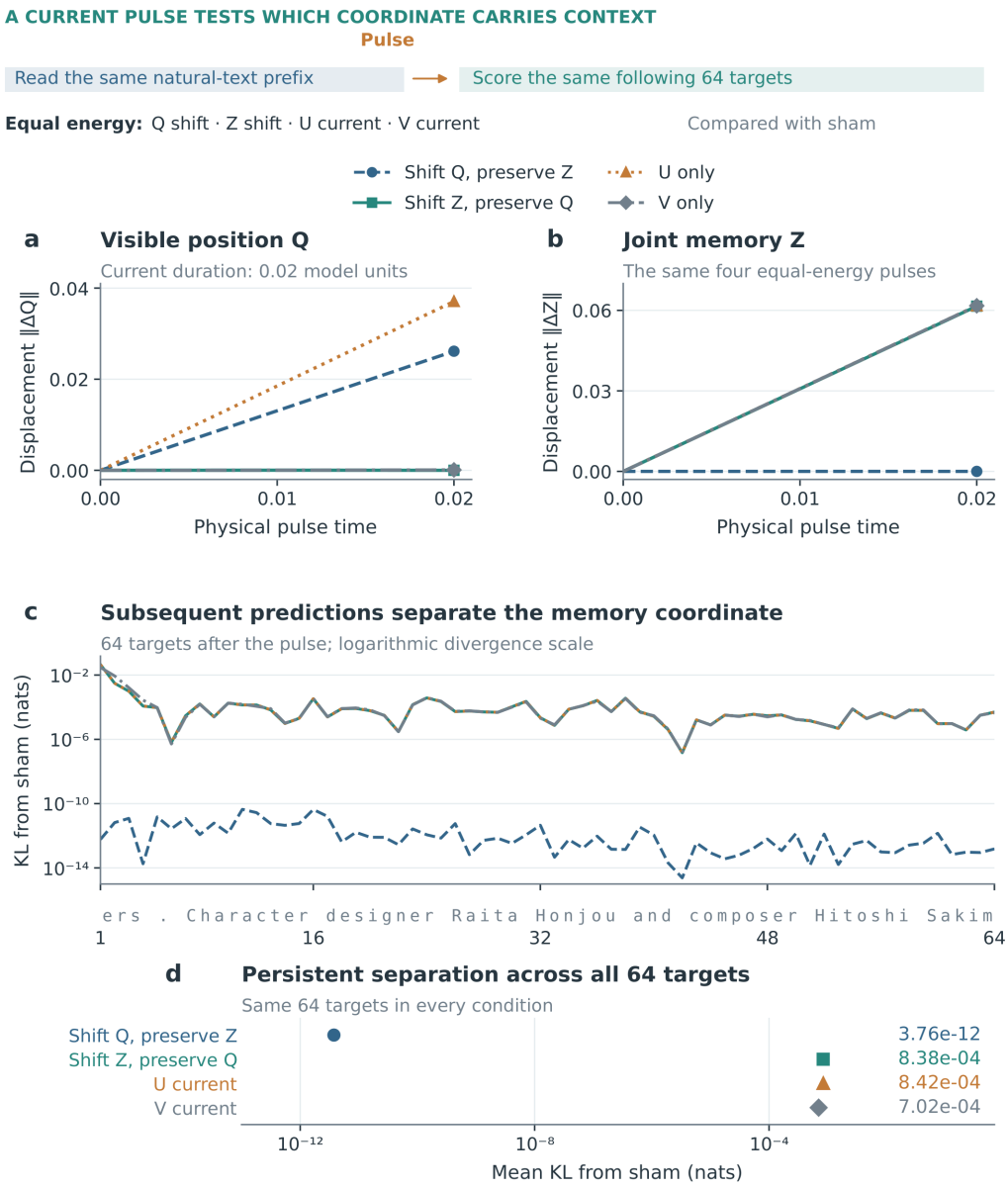

Figure S2: **A selective physical perturbation distinguishes visible position from predictive memory.** (a,b) The complete 17-point pulse trajectories of position and joint memory displacement, measured relative to the equal-time sham. Four current designs share the same nonzero input energy; symbols mark pulse endpoints. (c) All 64 subsequent predictive divergences, shown in target order with the corresponding natural text. Moving position while preserving joint memory leaves the prediction sequence near the sham; the other three designs change joint memory and the predictions. (d) Arithmetic mean divergence over the same 64 targets, with numerical values shown for each condition. All displacements and divergences are measured against the same equal-time sham trajectory. All measurements use one trained checkpoint, boundary, and direction; target bytes are successive observations within a trajectory.

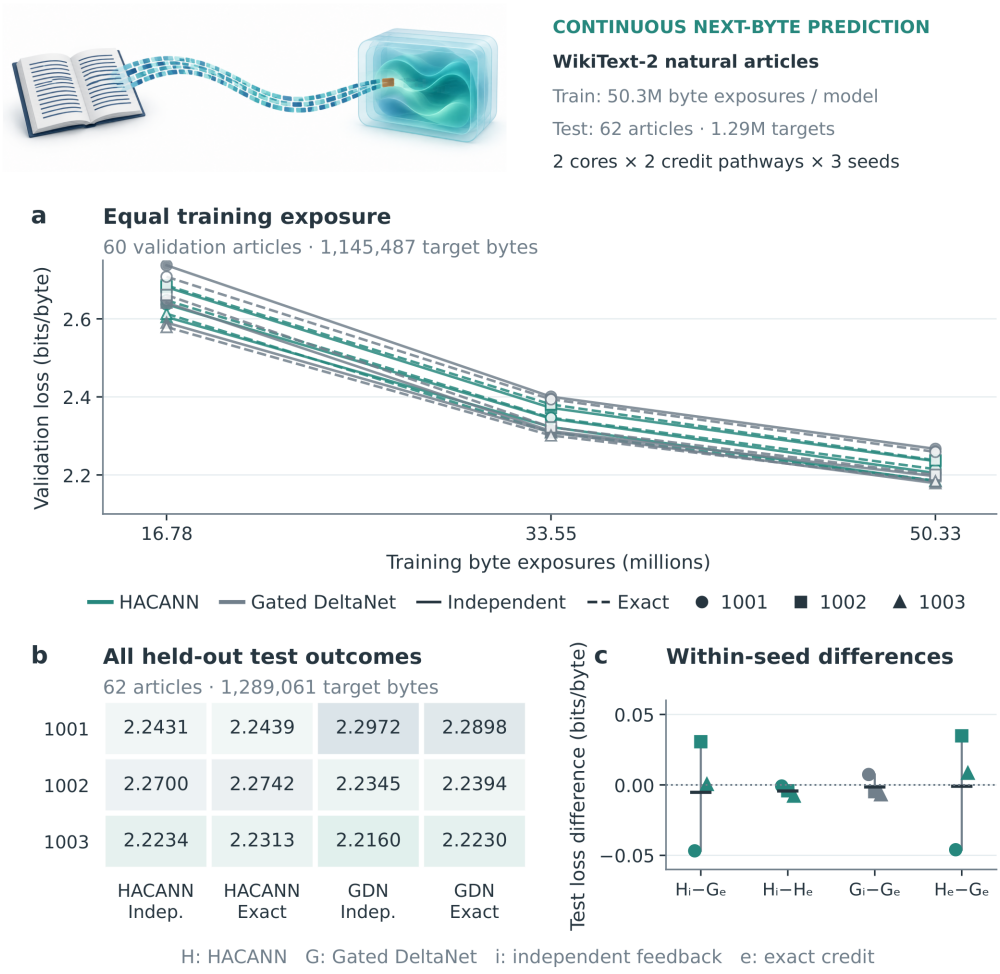

Figure S3: **Matched language learning across all twelve trained models.** (a) Every validation observation from the two cores, two credit pathways, and three seeds. Solid and dashed curves denote independent and exact credit; marker shapes identify seeds. All observations score 1,145,487 targets across 60 validation articles. Training exposures include repeated passages. (b) The complete test loss matrix in bits per byte. Every selected checkpoint scores all 62 test articles and 1,289,061 targets. (c) All paired seed differences for the four specified comparisons. Vertical segments span the observed seed differences; horizontal ticks indicate their mean. H and G denote HACANN and Gated DeltaNet; subscripts i and e denote independent and exact credit. These are descriptive outcomes from three trained seeds.

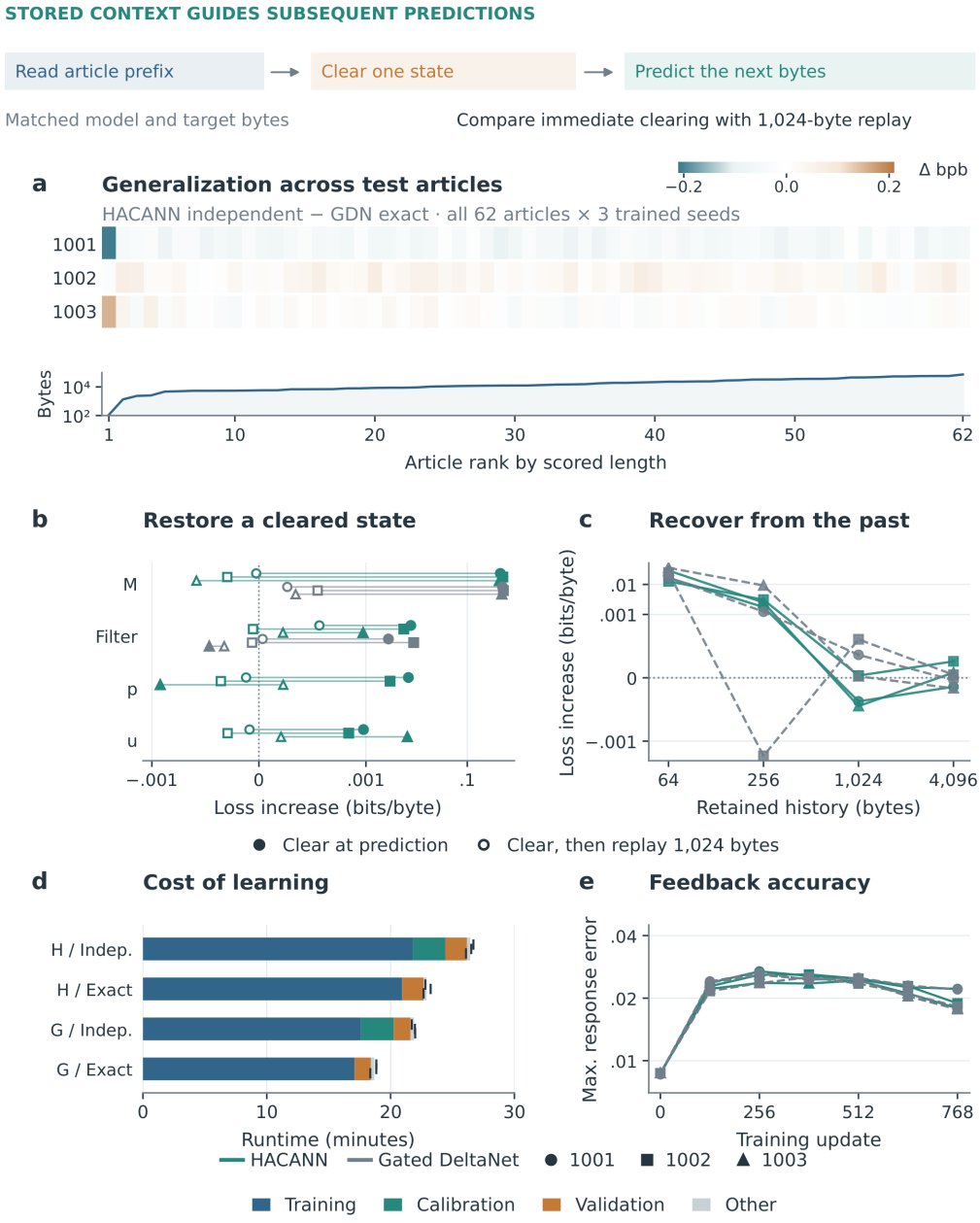

Figure S4: **Article variation, memory reconstruction, and the cost of language learning.** (a) All 186 article-level loss differences between HACANN with independent feedback and Gated DeltaNet with exact credit. Each column is one test article, ordered by its scored byte count; rows identify trained seeds. The lower trace gives article lengths. Negative differences favor HACANN. (b) Filled markers show immediate state clearing; paired open markers show clearing followed by 1,024 bytes of natural input before the same targets. Prediction recovers after replay. Prediction p and evidence u apply to HACANN. (c) All retained-suffix outcomes. The loss axes in (b,c) are symmetric logarithmic, linear within 10^{-4} bits per byte of zero. (d) Mean runtime components in all four factorial conditions, with every seed's total marked in black. H and G denote HACANN and Gated DeltaNet. (e) Every recorded feedback calibration endpoint in the independent-credit models. Marker shape identifies the trained seed throughout. Article and target measurements describe variation within these models.

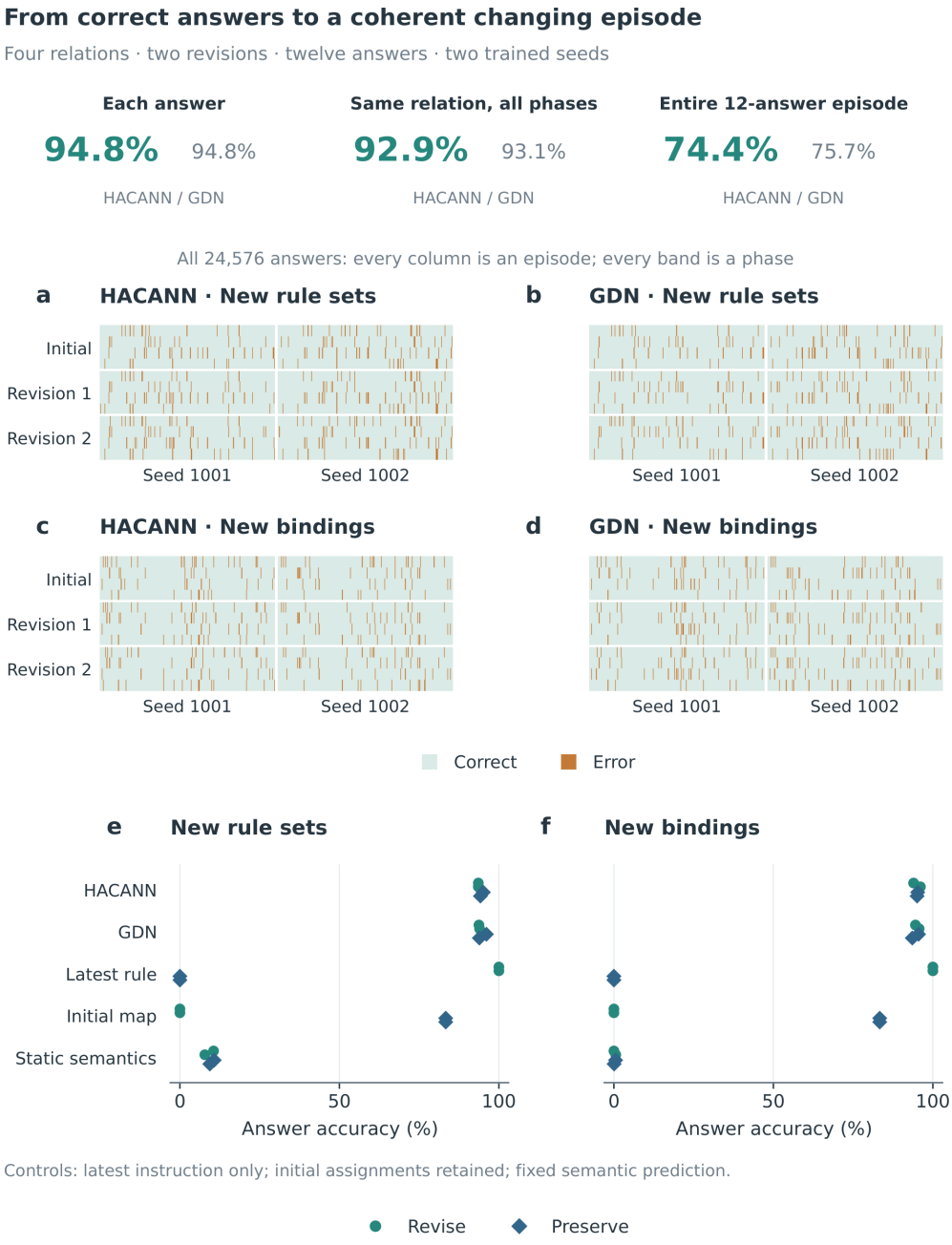

Figure S5: **Maintaining a coherent changing episode.** The opening summaries distinguish pooled per-answer accuracy, accuracy for a relation throughout all three phases, and complete-episode success. (a–d) All 24,576 main-task responses: each column is an episode and each four-row band is a phase. Relations retain their row positions through both revisions; trained seeds occupy separate halves. (e,f) Revision and preservation outcomes for the two models and three diagnostic policies, separated by transfer suite. Lines join observed seed values. The summaries pool both trained seeds and transfer suites.

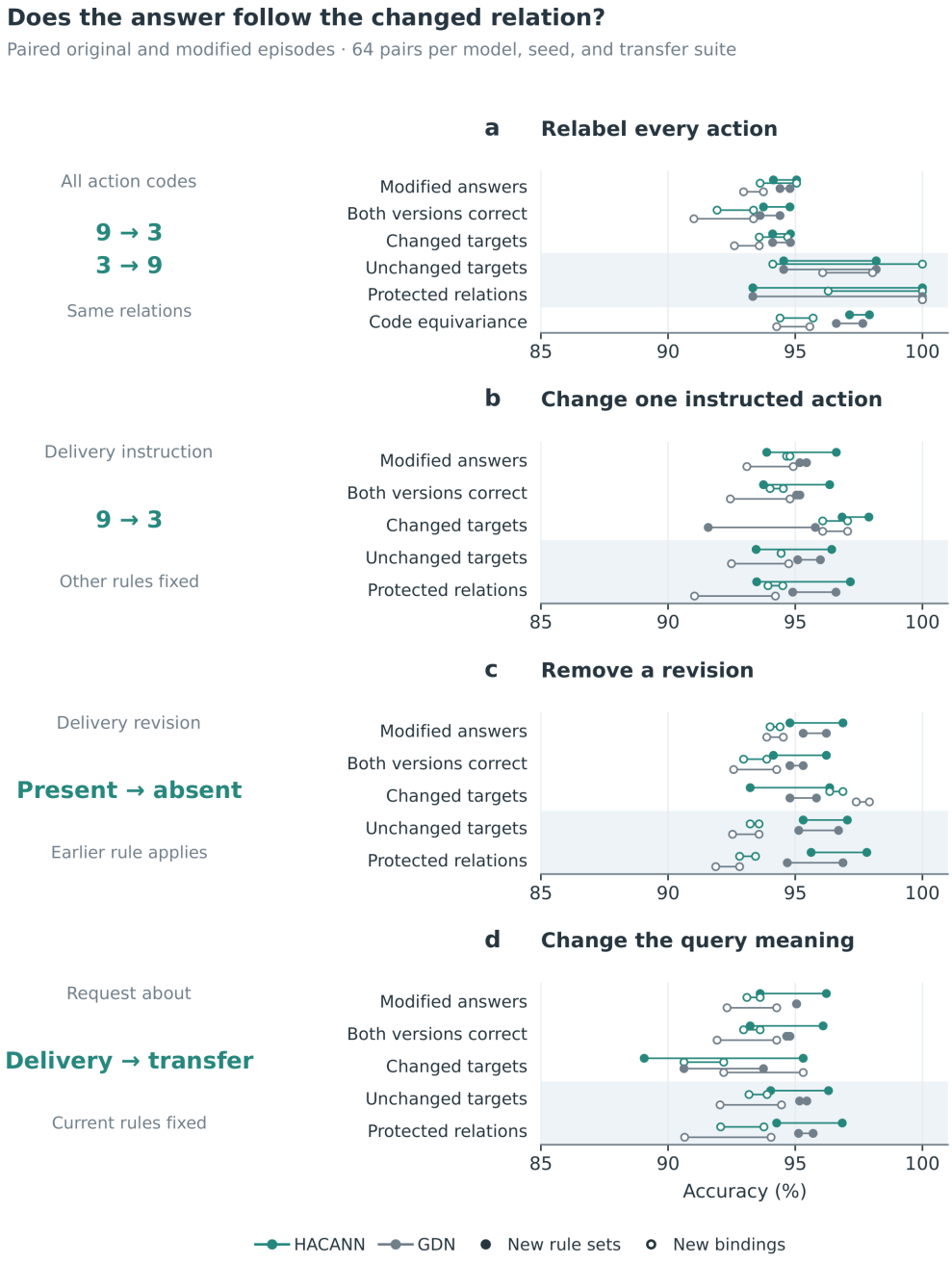

Figure S6: **Responses to controlled changes in the relation.** The examples at left introduce (a) action-code relabeling, (b) a changed instructed action, (c) a removed revision, and (d) a query with changed meaning. Plots retain all 168 recorded outcomes. Connectors join two trained seeds within a model and transfer suite; marker fill denotes the suite. Shaded rows identify preservation measures. Each evaluation uses 64 paired episodes. Metrics without eligible targets have no row. All panels share the same accuracy scale; Table 13 gives eligible target counts.

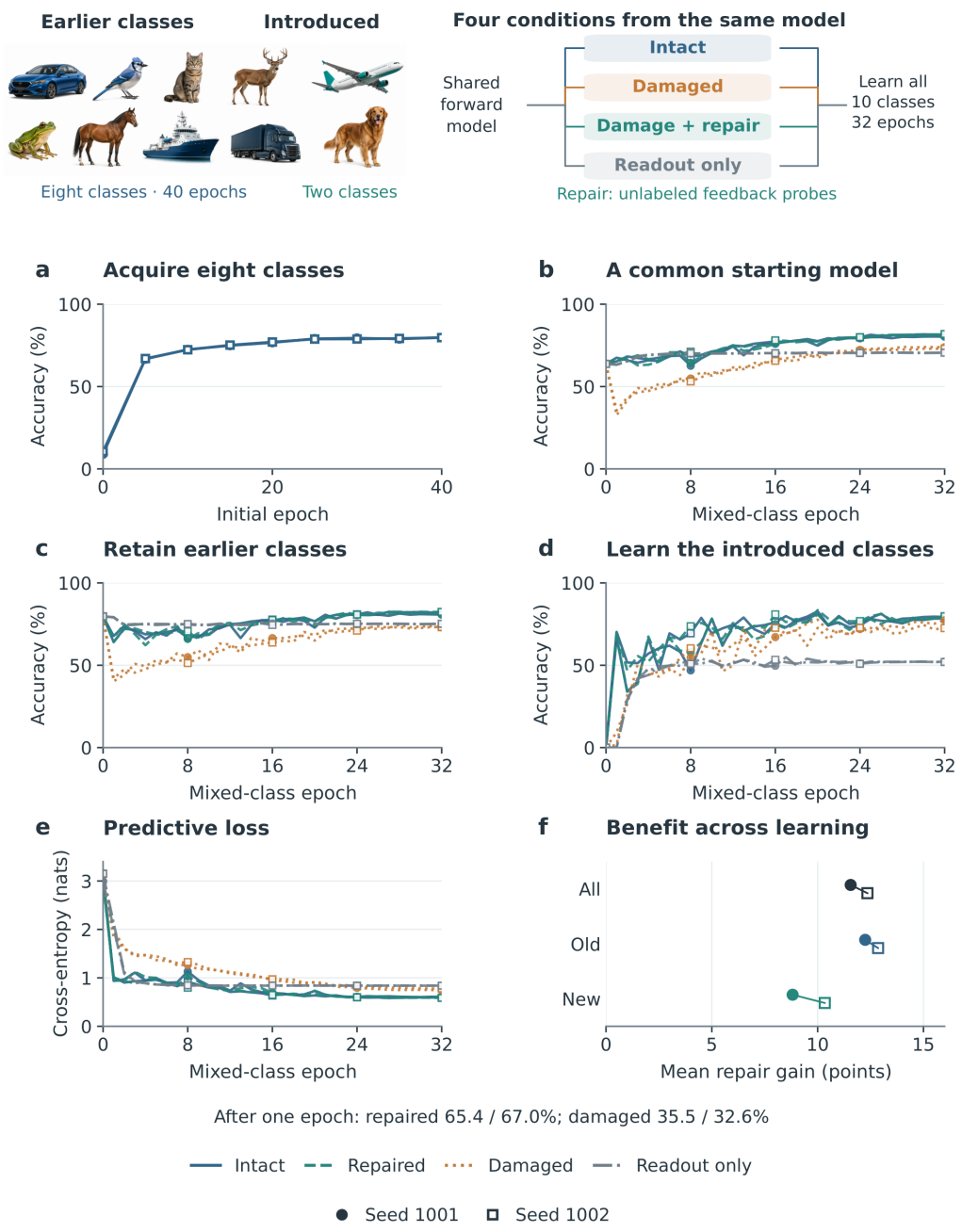

Figure S7: **The course of learning after feedback repair.** The illustrated protocol starts from one eight-class model before matched exposure to all classes. (a) Initial development accuracy. (b) Accuracy during mixed-class learning. (c,d) Earlier-class retention and introduced-class acquisition. (e) Predictive cross-entropy. All 18 initial-learning and 264 mixed-learning observations are shown. (f) Within-seed repaired-minus-damaged accuracy averaged over learning: trapezoidal integration over epochs 0–32, divided by 32, in percentage points. Circles and squares identify the two trained seeds.

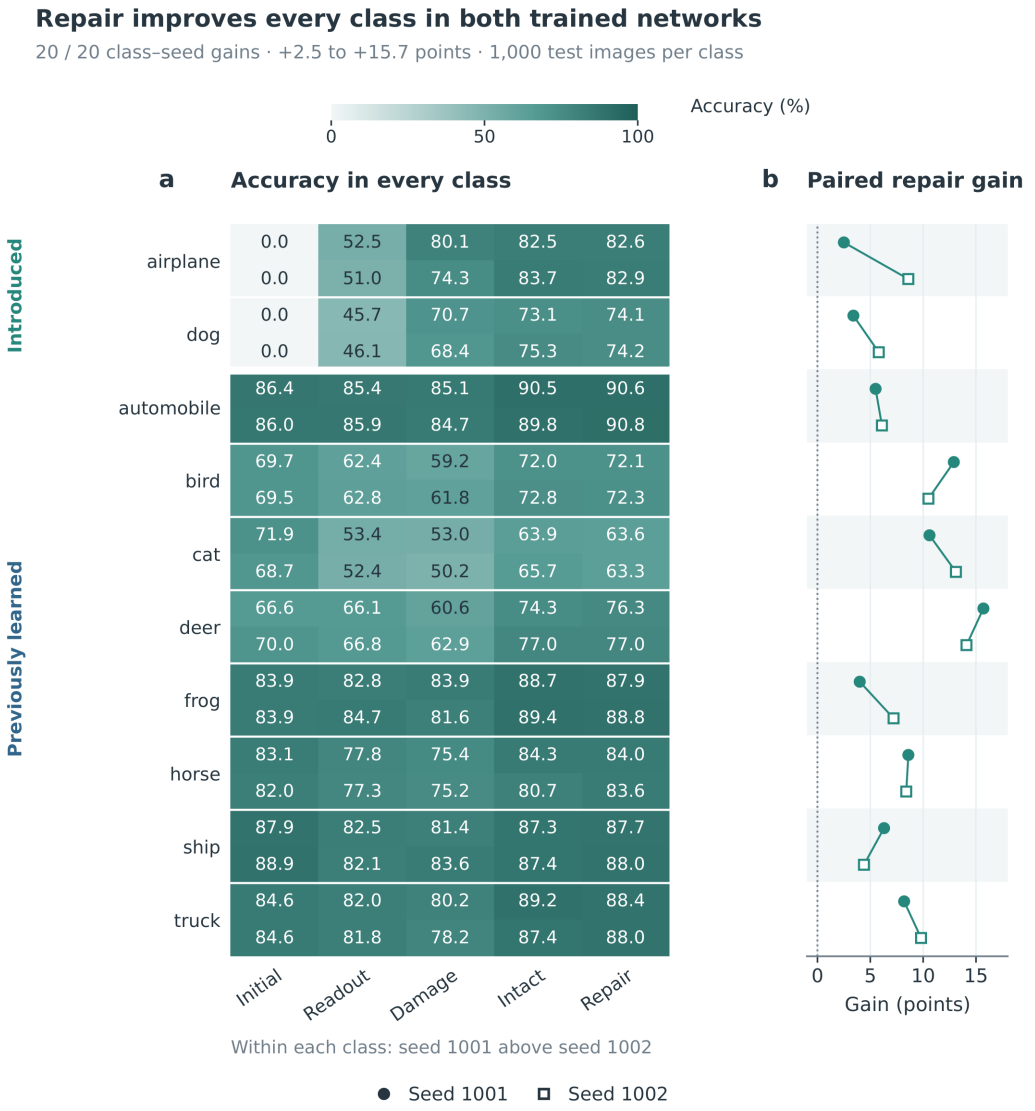

Figure S8: **Repair improves every class in both trained networks.** (a) All 100 class-by-branch-by-seed accuracies, each computed from 1,000 held-out images. Introduced classes appear above the eight earlier classes. Within each class, seed 1001 appears above seed 1002. (b) All 20 paired repaired-minus-damaged accuracy gains, aligned with their corresponding rows. Every gain is positive, spanning 2.5–15.7 percentage points. The comparison covers both acquisition of new classes and retention of earlier knowledge.

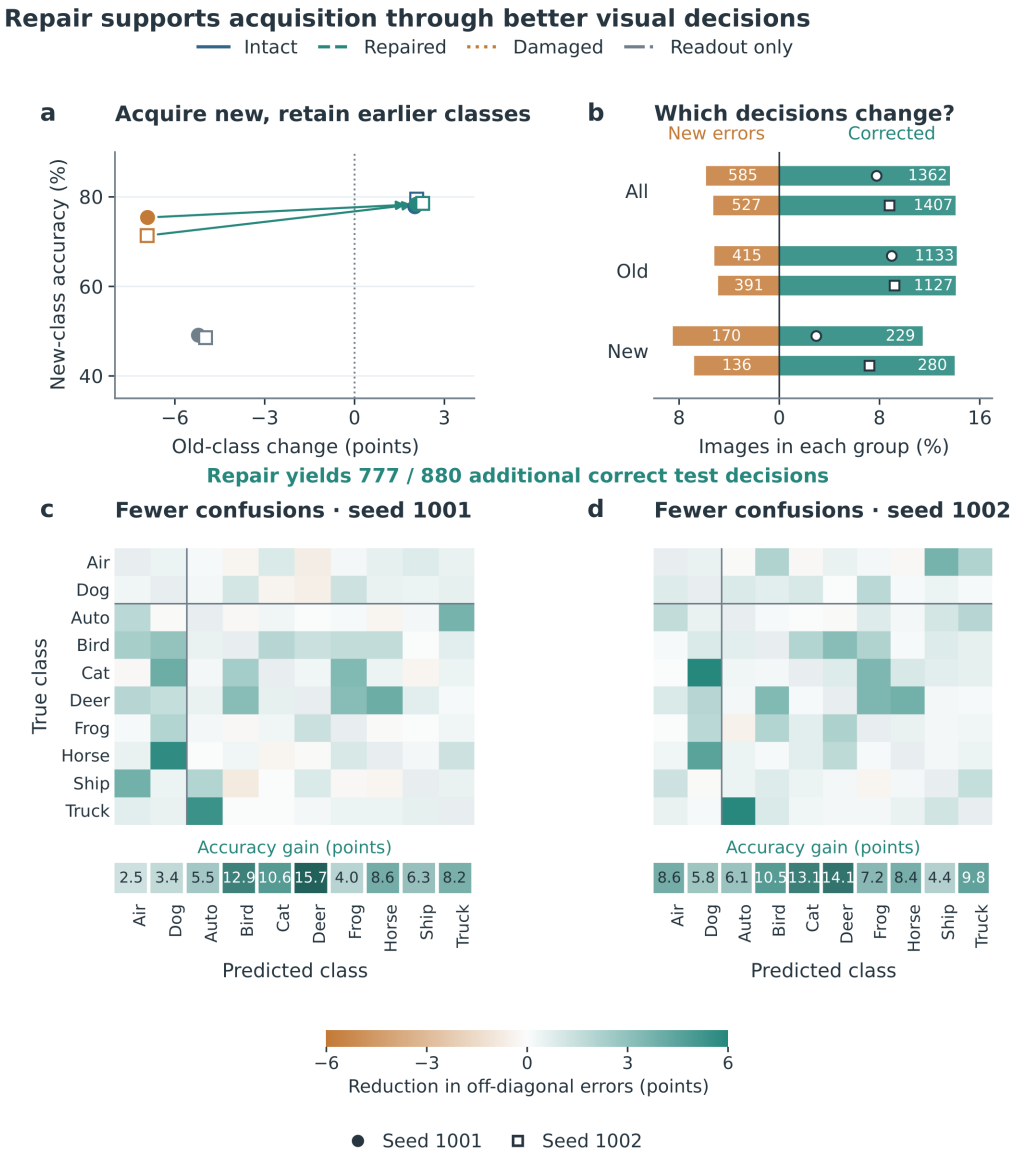

Figure S9: **How feedback repair changes visual decisions.** (a) New-class accuracy against old-class accuracy change from the initial checkpoint; arrows connect damaged and repaired outcomes within a seed. (b) Images corrected or made newly incorrect, with exact counts and net gains. Group denominators are 10,000, 8,000, and 2,000 images; upper and lower rows identify seeds 1001 and 1002. (c,d) Off-diagonal confusion reduction, computed as damaged minus repaired. Teal denotes fewer mistakes. Separate strips show diagonal accuracy gains, computed as repaired minus damaged. Entries are percentage-point changes within each true class; all 200 entries are retained.