DENDRITIC COMPUTATION SHAPES HOW NETWORKS LEARN TO USE CONTEXT

Anonymous authors
Paper under double-blind review

ABSTRACT

Flexible behavior requires learning how to use sensory and contextual information. A recent in vivo study found that activating dendrite-inhibiting interneurons impaired rule relearning while sparing established behavior. How dendritic input organization shapes learned sensorimotor mappings remains unclear. Here we develop a trainable framework of recurrent circuits built from biophysically detailed neurons to manipulate input location and dendritic conductance during learning and execution. In a controlled task, sensory input encodes both direction and rule, while a separate context stream repeats the rule. Apical and basal routing yield similar performance with all inputs present, but apical routing produces less dependence on context after learning. Exchanging sensory weights after training and sharing them during training show that learned sensory parameters contribute to both the expression and formation of this difference. Blocking the apical calcium conductance during training partially narrows the post-withdrawal performance gap even after conductance restoration, whereas acute blockade has much smaller effects on performance without context. Within apical-context circuits, exchanging sensory weights between these training histories also transfers this persistent effect, linking conductance conditions during learning to functionally consequential sensory parameters. A simulated robotic pushing task further shows that context availability during learning affects subsequent closed-loop control under common execution conditions. These findings identify dendritic computation as a source of learning bias, linking cellular input organization and biophysics to learned sensorimotor mappings, with a proof of concept in artificial control.

1 INTRODUCTION

Flexible behavior in changing environments requires learning which information to use and how to combine it (Heald et al., 2021). Understanding this process is central to neuroscience, with implications for generalization in artificial intelligence (Flesch et al., 2022). In context-dependent decision-making, neural circuits learn relationships between sensory features, task rules and appropriate actions (Mante et al., 2013). Yet similar behavioral performance can arise from different underlying neural computations (Pagan et al., 2025). This distinction becomes especially clear when the same task variable is available through multiple input pathways. If a rule encoded within sensory input is also supplied through a separate contextual pathway, removing the additional stream leaves the required task variables available, but may disrupt a solution that has learned to depend on it. This distinction between available information and learned dependence raises a fundamental question: how does the physical organization of neural inputs shape the sensorimotor mappings acquired through experience?

Dendrites provide a cellular basis for investigating this question. In cortical pyramidal neurons, inputs arriving at different locations interact through electrical coupling, active conductances and local nonlinearities (Larkum et al., 1999; Schiller et al., 2000; Poirazi et al., 2003; Polsky et al., 2004). Recent in vivo experiments highlight the importance of examining these interactions during learning. In mouse frontal motor cortex, activating interneurons that inhibit apical dendrites impaired rule relearning while sparing established behavior (Maristany de las Casas et al., 2026; Fig. 1A). This finding motivates separating how dendritic conditions influence the acquisition of behavior from their immediate effects during execution.

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.

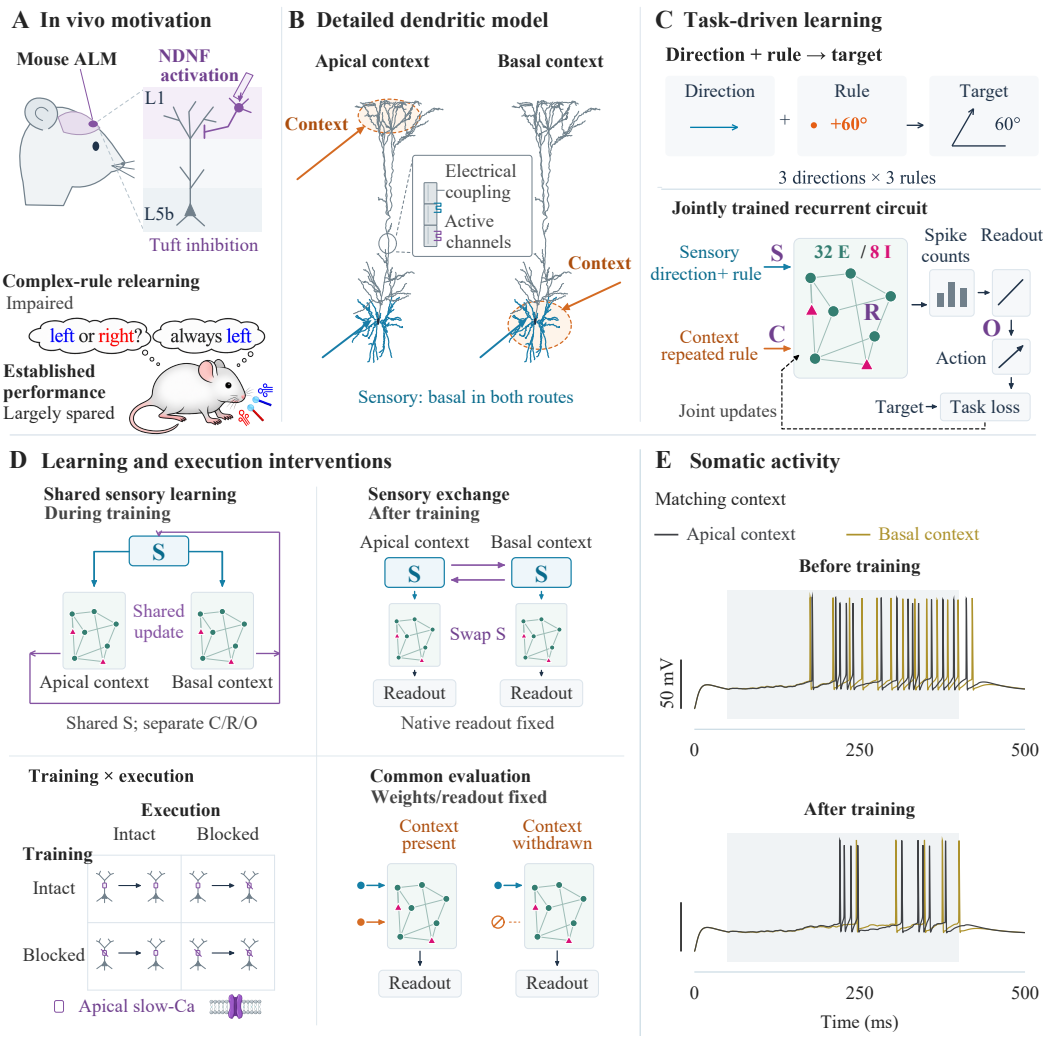


Figure 1: **Biological motivation, dendritic circuit model and intervention design.** **(A)** Activating dendrite-inhibiting NDNF interneurons impairs rule relearning while largely sparing established behavior (Maristany de las Casas et al., 2026). **(B)** Biophysically detailed neurons receive sensory input on basal dendrites. Context targets either apical or basal dendrites of excitatory neurons. **(C)** Nine direction–rule conditions define the task. Sensory input encodes both variables; context repeats the rule. Sensory, context, recurrent and readout parameters are jointly trained. **(D)** Interventions include shared sensory learning, post-training sensory-weight exchange and a 2×2 training–execution design for apical calcium blockade. Context withdrawal preserves learned parameters. The modeled conductance blockade is distinct from the interneuron-mediated inhibition in panel **(A)**. **(E)** Somatic-voltage examples from corresponding neurons in the apical- and basal-context circuits ($s_C = 1.8/1$), before and after training with matching context. Both plots share time and voltage scales; shading marks the 50–400 ms input window. Population rasters and dendritic voltages appear in Fig. S1 (Appendix B).

Computational models already connect dendritic integration to pathway gating, contextual processing and representation learning (Yang et al., 2016; Wybo et al., 2023; Islah et al., 2025). We ask how the dendritic location of an additional contextual input changes the learned mapping when its task information is already available through the sensory pathway. Addressing this question requires identifying which learned parameters carry the resulting behavioral differences and whether biophysical conditions during training leave effects that persist after those conditions are restored.

Here we develop an intervention framework using trainable recurrent circuits built from biophysically detailed neurons, retaining dendritic morphology, electrical coupling and active conductances. In a controlled rule-dependent sensorimotor remapping task, basal sensory input encodes both stimulus direction and task rule, while an auxiliary stream repeats the rule at apical or basal dendritic locations. Synaptic and readout parameters are jointly optimized, with synaptic credit assignment and gradient computation based on gradient replay (He et al., 2026b). The framework allows interventions on contextual input, learned synaptic parameters and dendritic conductances during learning and execution.

Our main contributions are:

1. **Dendritic organization as a cellular source of learning inductive bias.** Our trainable biophysical framework shows that dendritic input location shapes how circuits learn to use the same task information. Apical routing reduces learned reliance on redundant context, revealing how spatial input organization can guide the acquisition of different sensorimotor solutions.
2. **A mechanistic link between dendritic learning conditions and later behavior.** Targeted sensory-weight interventions show that learned synapses carry behavioral effects of input routing and, within apical-context circuits, calcium-conductance history. Separating training from execution further identifies lasting consequences of dendritic conductance conditions. Together, these interventions connect cellular biophysics during learning to the synaptic parameters and behavior acquired through experience.
3. **A route from cellular learning mechanisms to closed-loop control.** We extend the framework to robotic pushing, where actions continually change subsequent sensory input. Paired training histories show that context availability during learning changes later control under common execution conditions, providing a proof of concept for translating dendritic learning principles into brain-inspired artificial control.

Together, these findings identify dendritic computation as a source of learning bias and provide a framework for translating cellular mechanisms into principles for artificial learning and control.

2 Related Work

Context-dependent computation. Mixed selectivity and recurrent dynamics support flexible computation (Rigotti et al., 2013; Mante et al., 2013), while multitask training can produce shared and specialized representations (Yang et al., 2019). At the dendritic level, models implement pathway-specific gating (Yang et al., 2016), exploit NMDA-dependent modulation for multitask representation learning (Wybo et al., 2023), and learn to combine contextual and sensory signals under ambiguity (Islah et al., 2025). Feature-wise conditioning provides a related computational motif in artificial networks (Perez et al., 2018). These studies connect contextual input to both representation learning and online computation. We examine how the dendritic route of context shapes jointly learned sensory–action mappings and the resulting dependence on context during execution.

Dendritic integration and learning. Experimental studies link active dendrites to perceptual detection and cortical feedback (Takahashi et al., 2016; Fişek et al., 2023), and show that learning changes the relative influence of bottom-up and top-down inputs (Makino & Komiyama, 2015). Branch-specific calcium events can support persistent synaptic plasticity (Cichon & Gan, 2015). Maristany de las Casas et al. (2026) further demonstrate a learning–execution dissociation: activating dendrite-inhibiting layer 1 interneurons in mouse frontal motor cortex impairs rule relearning while sparing established behavior. Computational models propose roles for segregated dendrites in credit assignment (Guerguiev et al., 2017; Sacramento et al., 2018), learning rules that exploit nonlinear dendritic integration (Bicknell & Häusser, 2021), and local contextual plasticity supporting association and assembly formation (Baronig & Legenstein, 2024; Onasch et al., 2026). GPU-accelerated simulation and gradient-based learning have enabled task-driven training of morphologically detailed neuronal networks (Zhang et al., 2023; Deistler et al., 2025; He et al., 2026a). Using supervised circuit optimization, we test the functional contributions of learned sensory parameters to context use. Independently varying conductance during training and execution separates persistent training-history effects from acute execution effects.

Redundancy, reliance, and auxiliary information. Shortcut learning highlights the importance of examining which features support a model's predictions beyond its performance on the training distribution (Geirhos et al., 2020). Simplicity bias and gradient starvation describe how optimization can favor some predictive features over alternatives (Shah et al., 2020; Pezeshki et al., 2021). Learning using privileged information explicitly exploits additional training information to improve a predictor evaluated without it (Vapnik & Vashist, 2009), while dropout exposes networks to randomly omitted units during learning (Srivastava et al., 2014). We study learned reliance on auxiliary context with different degrees of overlap with sensory information. Our tasks use either a rule cue fully redundant with sensory task variables or a partially redundant phase cue that incorporates trajectory history and provides an explicit task-stage organization of sensory observations.

3 METHODS

Task and inputs. The controlled sensorimotor remapping task combines three input directions, $\theta \in \{0°, 120°, 240°\}$, with three rotation rules, $\phi \in \{60°, 180°, 300°\}$, yielding nine conditions but three target directions (Fig. 1C). The target action is $y = (\cos(\theta + \phi), \sin(\theta + \phi))^\top$. Sensory input encodes direction and rule category through 40 channels, with 20 channels for each variable; an auxiliary stream repeats the same rule category through 30 context channels. Thus, withdrawing context leaves both task variables available through sensory input. Encoding details are given in Appendix A.1.

Biophysical circuit and routing. Each circuit has 32 excitatory and 8 inhibitory multicompartment neurons from Larkum et al. (2009) and the Allen Institute (Allen Institute, 2015). Their dynamics obey

$$\begin{aligned} C_k^{\mathrm{mem}} \dot{V}_k &= \sum_{\ell \in \mathcal{N}(k)} g_{k\ell}^{\mathrm{ax}} (V_\ell - V_k) - I_k^{\mathrm{ion}}(V_k, q_k) + I_k^{\mathrm{syn}} + I_k^{\mathrm{bg}}, \\ \dot{q}_k &= F_k(V_k, q_k). \end{aligned} \tag{1}$$

Here, k indexes compartments, $\mathcal{N}(k)$ their axial neighbors, C^{mem} capacitance, g^{ax} coupling, and q gating/calcium states. Synaptic current sums the inputs arriving at compartment k:

$$I_k^{\mathrm{syn}}(t) = \sum_{e \in \mathcal{S}_k(r)} \sum_{\alpha \in \{\mathrm{AMPA}, \mathrm{NMDA}, \mathrm{GABA}\}} g_{e\alpha}(t, V_k; w_e)(E_\alpha - V_k). \tag{2}$$

Here, $\mathcal{S}_k(r)$ contains synapses on compartment k under route r. Each synaptic-current term is the product of the conductance $g_{e\alpha}$ and the driving force $E_\alpha - V_k$. Here, E_α is the reversal potential; inward current is positive. Sensory/recurrent inputs target basal dendrites; context targets excitatory apical or basal dendrites. Morphology and sites stay fixed; background current is separate.

Task-driven joint learning. Raw somatic counts $h \in \mathbb{R}^{40}$ over $50 \le t < 500$ ms determine

$$z = W_O h + b_O, \qquad \hat{y} = \tanh z, \qquad \mathcal{L} = \tfrac{1}{2} \|\hat{y} - y\|_2^2. \tag{3}$$

Sensory (S), context (C), recurrent (R) and readout parameters jointly learn. Exact readout derivatives provide count credit:

$$\delta_z = (\hat{y} - y) \odot (1 - \hat{y}^{\,2}), \qquad \delta_h = W_O^\top \delta_z. \tag{4}$$

We extended HelioX (Lu et al., 2026) to support the replay-based synaptic gradient learning algorithm of He et al. (2026b). A voltage-dependent surrogate converts count credit to voltage credit for replay. Training comprises 4,500 trial-level optimization steps, with each trainable parameter group updated once per trial using Adam (Kingma & Ba, 2015). Algorithm 1 summarizes training; details are in Appendix A.

Algorithm 1 Joint learning in the controlled nine-condition task

Input: Routing and conductance conditions, parameters Θ, Adam states M.

```
for epoch = 1, ..., 500 do
    for each of the nine task conditions do
        Resample paired input events and reset neural states.
        Simulate each circuit; obtain spike counts, actions and loss.
        Compute readout gradients and replay-based synaptic gradients.
        If sharing sensory weights, average their gradients before updating.
        Apply regularized Adam updates and sign constraints; retain Θ and M.
    end for
end for
```

Learning and execution interventions. Post-training exchanges replace sensory, direction/rule-specific or recurrent weights across routes or training histories, then rerun recipients with other parameters and native readouts fixed. Shared learning ties sensory weights across two unconnected circuits, using one Adam state and

$$\bar{g}_S = \tfrac{1}{2}(g_{S,\mathrm{apical}} + g_{S,\mathrm{basal}}). \tag{5}$$

Both receive the updated S; C, R and readouts retain private weights/optimizer states. A 2×2 design crosses intact/blocked conductance during training and execution. Blockade zeros the maximal high-voltage-activated calcium conductance throughout excitatory apical dendrites; other intrinsic parameters stay fixed. Histories share initial weights within route; blockade can change initial activity. Restoration compares histories under common intact execution (Appendices C and E).

Controls and evaluation. Initial context strength scales a common $\mathcal{U}(0, 1.2\,\mathrm{nS})$ reference; common strengths are $1, 1.4, 1.8$. Equal strengths can yield different firing across input locations. Mechanistic analyses therefore use apical/basal $1.8/1$ for comparable initial population firing with matching context (Fig. S2A), without equating cellwise states. Despite unconstrained activity during learning, endpoint withdrawal-induced firing reductions are also similar; $1.8/1.8$ instead yields a substantially larger basal-context reduction (Fig. S2B,D). Initial S, R and readouts are paired across routes. Withdrawal removes context events; nonselective context equalizes rule-group rates, while conflict cyclically shifts the cued rule. Content changes preserve expected event budget, not necessarily effective drive; sensory/background events are preserved. A separate reference learns without context.

At the fixed epoch-500 endpoint, networks are evaluated without further learning using 20 new spike-train realizations for each of the nine conditions. Vector NMSE $E = \langle \|\hat{y} - y\|_2^2 \rangle = 2\langle \mathcal{L} \rangle$ (unit targets) gives withdrawal cost $D_r = E_{r,\mathrm{withdrawn}} - E_{r,\mathrm{matching}}$. Gaps are $\Gamma_{\mathrm{comp}} = E_{\mathrm{basal,withdrawn}} - E_{\mathrm{apical,withdrawn}}$ and $\Gamma_{\mathrm{dep}} = D_{\mathrm{basal}} - D_{\mathrm{apical}}$; percentage reductions use group-mean gaps. Means $\pm$ sample SD aggregate within-network metrics and paired contrasts across three independent seed pairs.

Closed-loop validation. We extend the framework to a simulated robotic manipulation task, CoffeePush in Meta-World (Yu et al., 2020), where circuits learn from expert demonstrations and their actions determine subsequent observations. An auxiliary phase cue incorporates trajectory history and is therefore not strictly redundant with the current observation. At epoch 20, each controller is duplicated with identical weights and Adam states into continuations that retain or withdraw context through epoch 50. Both training histories are evaluated with context present and withdrawn at matched checkpoints and rollout initial states. This comparison tests how context availability during continued learning affects subsequent closed-loop control under common execution conditions (Appendix D).

4 RESULTS

4.1 DENDRITIC ROUTING SHAPES LEARNED CONTEXT DEPENDENCE

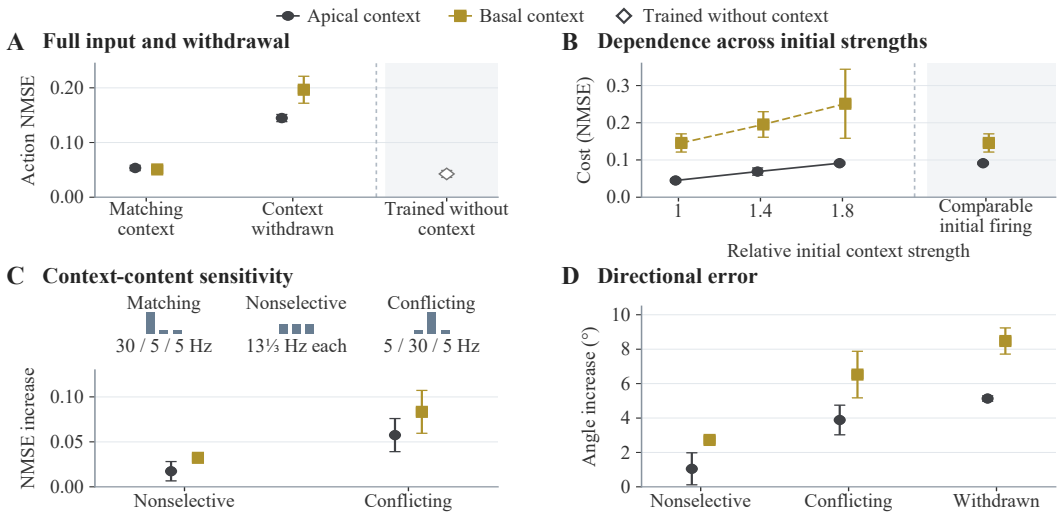


Figure 2: **Dendritic routing shapes learned context dependence.** **(A)** Matching- and withdrawn-context NMSE, with a context-free training reference. **(B)** Withdrawal cost across common initial strengths and a separate comparable-initial-firing pair. A, C and D use this pair ($s_C = 1.8/1$ for apical/basal context). **(C)** NMSE increases under nonselective or conflicting context. Insets show per-channel rates across rule groups, preserving the expected total event budget. **(D)** Angular-error increases under context interventions. Changes in C–D are relative to matching context. Points and bars show means and sample SD across three paired seeds.

We first asked whether dendritic input location influences how circuits learn to use redundant context. Sensory input encodes both direction and rule, so withdrawing the additional rule cue preserves the task variables needed to select the correct action. After joint training, apical- and basal-context circuits with comparable initial population firing achieve similar matching-context NMSE: 0.053 and 0.051, respectively. Context withdrawal reveals a difference: the corresponding error increases are 0.091 and 0.146, with smaller withdrawal costs and lower post-withdrawal errors for apical routing in all three seed pairs (Fig. 2A–B). A separate circuit group trained without context achieves the lowest context-free error. The apical advantage therefore concerns reduced dependence relative to basal routing.

We next tested whether this difference holds across initialization conditions and changes in context content. Apical routing yields smaller withdrawal costs at each of three common initial context strengths, as well as with route-specific strengths producing comparable initial population firing (Fig. 2B; Fig. S3). Redistributing context events across rule groups while preserving their expected total budget also produces different behavioral effects. Nonselective and conflicting context increase errors more in basal-context circuits, including larger angular-error increases (Fig. 2C–D). Conflicting context also produces a larger mean signed angular shift along the false-rule direction under basal routing (Fig. S4B). Thus, similar full-input performance conceals different learned use of contextual information, evident in both withdrawal costs and context-induced changes in action direction.

4.2 LEARNED SENSORY PARAMETERS CONTRIBUTE TO THE ROUTING DIFFERENCE

Where is the behavioral difference expressed, and does sensory learning contribute to its formation? We first exchanged sensory weights between trained apical- and basal-context circuits, retaining each recipient's other parameters and original readout without refitting (Fig. 3A). Basal-trained sensory weights raise apical-context post-withdrawal NMSE from 0.145 to 0.197; the reverse exchange lowers basal-context error from 0.197 to 0.159. Both directions hold in all three seed pairs (Fig. 3B). Direction- and rule-only exchanges each raise apical-context error in all three pairs; they lower basal-context error in three and two pairs, respectively. The effect therefore extends beyond the

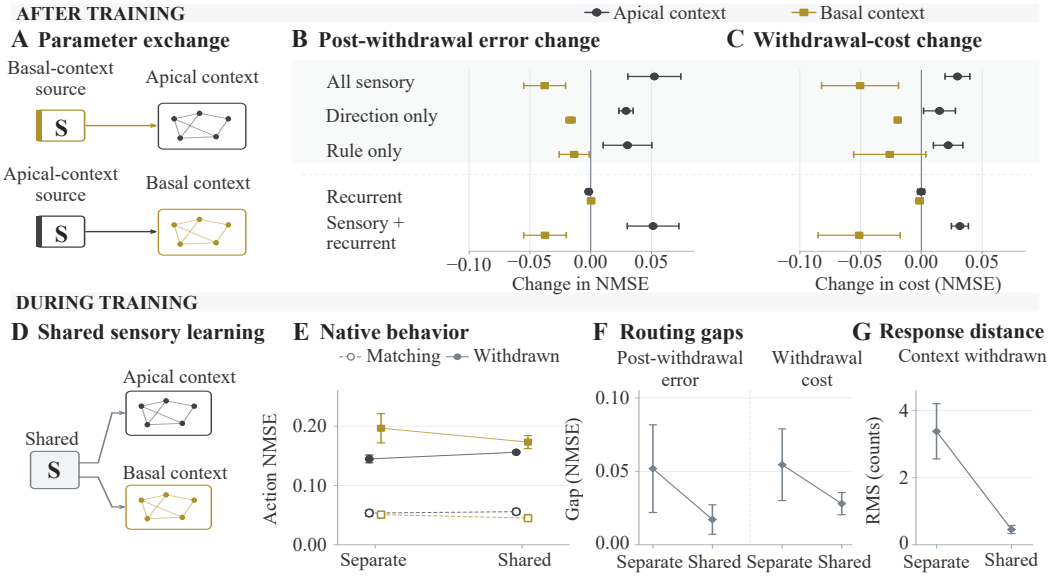


Figure 3: **Learned sensory parameters contribute to the expression and formation of context dependence.** **(A)** Reciprocal sensory-weight (S) exchange with other parameters and native readouts retained. **(B–C)** Changes in withdrawn-context NMSE and withdrawal cost relative to recipient baselines; colors identify recipients. **(D)** Shared sensory learning; other parameters update separately. **(E–F)** Native errors and basal-minus-apical gaps under separate or shared sensory learning; Separate denotes independently updated sensory weights. **(G)** RMS Euclidean distance across the nine conditions between corresponding raw 40D condition-mean spike-count vectors under withdrawal. Means ± SD across three paired seeds; gaps and distances are computed within seed.

variable duplicated by context, implicating the jointly learned sensory–action mapping. Recurrent-only exchanges have much smaller mean effects, and combined sensory–recurrent exchanges yield changes close to sensory exchange alone (Fig. 3B–C). In basal-context recipients, reduced withdrawal cost can combine better withdrawn-context performance with worse matching-context performance (Fig. 3C; Appendix C.2).

To test whether sensory-parameter divergence helps establish the routing difference, we trained two otherwise separate circuits with shared sensory weights. Context, recurrent and readout parameters remained independently trainable (Fig. 3D). Sharing raises apical-context withdrawal error and lowers basal-context error, reducing the post-withdrawal error gap by 67% and the withdrawal-cost gap by 49% (Fig. 3E–F). These percentages quantify the full shared-training intervention, including changes in joint optimization. Population responses under withdrawal become closer across routes (Fig. 3G); residual behavioral differences also depend on the jointly learned readouts (Appendix C.3). Separate controls show that the routing contrast persists with frozen context weights and varies with the context learning rate (Appendix B, Fig. S5). Together, post-training exchange and training-time sharing show that learned sensory parameters contribute to both the expression and formation of the routing difference.

4.3 Training-time apical calcium conductance shapes later behavior

Can dendritic conductance influence the mapping acquired through learning, leaving effects after normal conductance is restored? We independently varied the slow, high-voltage-activated calcium conductance in apical dendrites during training and execution. Within each routing configuration, intact- and blockade-trained circuits started from identical trainable parameters. With context withdrawn, acute blockade after intact training changes apical-context NMSE by only -0.00055. In contrast, blockade throughout training increases subsequent error by 0.014 even after conductance restoration (Fig. 4A). The corresponding training effect is smaller for basal routing, narrowing the post-withdrawal error gap from 0.052 to 0.040 under common intact execution—a 22.3% reduction

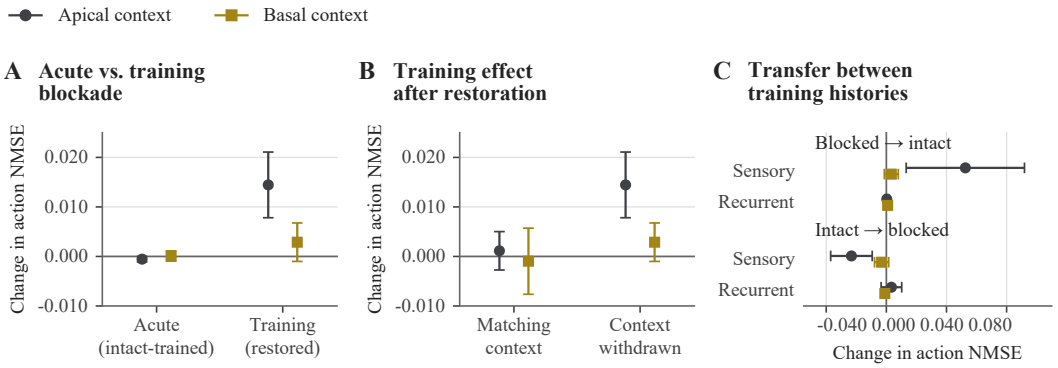


Figure 4: **Training-time apical calcium blockade alters withdrawal behavior. (A)** Changes in context-withdrawn NMSE after acute blockade in intact-trained circuits or training blockade followed by restoration. **(B)** Blocked-minus-intact training effects under intact execution, with matching or withdrawn context. A–B reference intact training and execution. **(C)** Within-route sensory or recurrent weight transfers, with intact conductance, withdrawn context and the recipient's native readout. Arrows indicate donor-to-recipient training history; changes reference unexchanged recipients. Points and bars show means ± sample SD across three paired seeds.

driven mainly by the larger apical-context error increase (Fig. S13). Training-history effects are also smaller under matching-context evaluation (Fig. 4B).

We asked whether learned sensory parameters contribute to this persistent effect. Within apical-context circuits, we exchanged sensory weights between the conductance-training histories, evaluating all recipients with intact conductance and their original readouts. Transferring blockade-trained sensory weights into intact-trained recipients increases post-withdrawal NMSE by 0.053; the reverse transfer decreases it by 0.023. Both directions hold in all three seeds (Fig. 4C), whereas recurrent-only exchanges have much smaller mean effects. Exchange effects depend on the recipient's jointly learned parameters and can exceed the original training-history difference, so they do not identify additive mediation fractions (Appendix E). These interventions link apical calcium conditions during learning to persistent behavioral differences partly expressed through learned sensory parameters.

4.4 Context availability during learning shapes closed-loop control

Context withdrawal measures dependence in an acquired controller; assessing training benefit requires comparing histories under common execution conditions. We tested this distinction in simulated CoffeePush, where actions change subsequent observations. Circuits were trained by imitation from 20 demonstrations, using low-dimensional geometric observations and a reach/contact/push cue that incorporates trajectory history and is not strictly redundant with the current observation (Fig. 5A; Appendices D.1 and D.2). Standard withdrawal costs vary substantially across networks (Fig. 5B; Table S7); descriptive ANN/SNN comparisons appear in Appendix D.5.

At epoch 20, each controller was duplicated with identical weights and Adam states into two branches: one retained context during continued training, while the other withdrew it. Training samples and sensory events were paired, and both branches were evaluated with context present and withdrawn at matched checkpoints and rollout initial states. This design isolates the effect of continued context availability within each source controller. Effects are retain-minus-withdraw success differences, averaged over 15 post-branch checkpoints within each source controller and then summarized across three sources per route (Appendix D.4).

Under common context-present evaluation, retaining context during training improves checkpoint-averaged apical-context success by 9.42 ± 8.43 percentage points, positive in all three source seeds (Fig. 5C; Table S6B). The basal-context effect is -2.13 ± 1.29 percentage points, negative in all three source seeds. Under common context-withdrawn evaluation, effects have mixed signs across seeds in both configurations. The clearest benefit therefore concerns how apical-context controllers learn to use context that remains available during execution. These paired continuations provide a

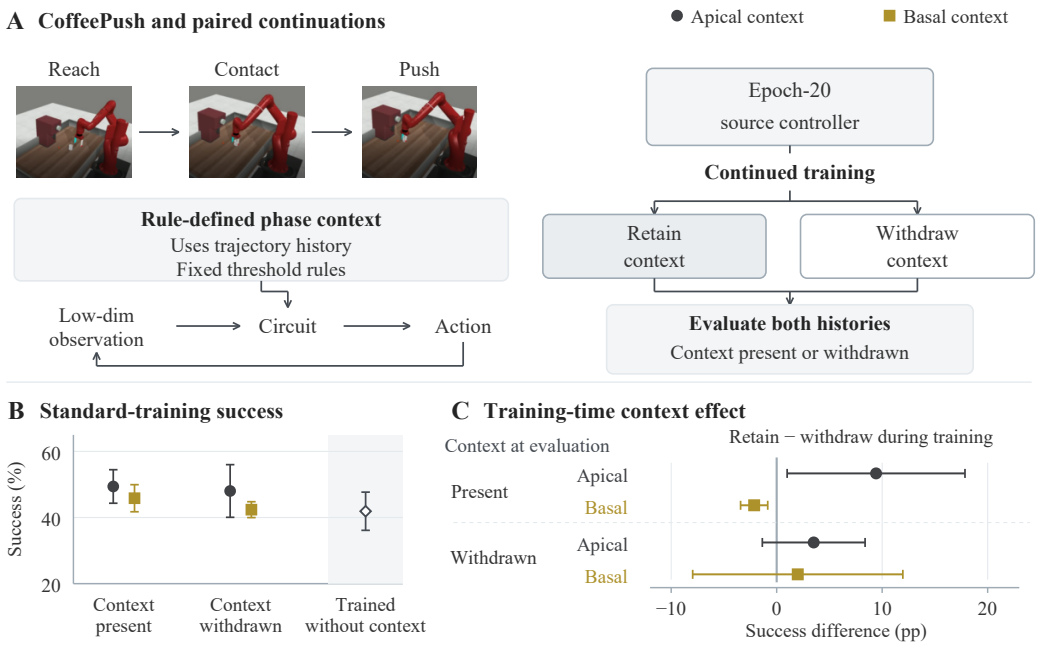


Figure 5: **Training-time context availability shapes closed-loop control. (A)** CoffeePush with a history-derived phase cue. Identical epoch-20 weights and Adam states initialize paired continuations with context retained or withdrawn through epoch 50; both histories are evaluated under both context conditions. **(B)** Standard-training success and an independent context-free reference. **(C)** Paired success differences (retain minus withdraw during training) under common execution conditions. Points and bars show means ± sample SD across three networks (B) or source controllers (C) per route, averaging 21 checkpoints (epochs 10–50; B) or 15 (epochs 22–50; C) within networks.

proof of concept that context availability during learning can alter subsequent closed-loop control, with outcomes depending on routing configuration and execution conditions.

5 Discussion and Conclusion

Our results show that dendritic input organization can bias the sensorimotor mappings acquired through learning. Circuits receiving the same task information achieve similar performance yet develop different dependence on an auxiliary context stream. Interventions on sensory weights and dendritic conductances connect these differences to learned sensory parameters and biophysical conditions during training. This implies that dendritic computation shapes not only responses to current inputs, but also how learning establishes the influence of those inputs on behavior.

Applications and implications. The framework provides a computational testbed for investigating how dendritic mechanisms contribute to context-dependent learning. Manipulating input location and conductance at different stages allows hypotheses about behavioral acquisition and execution to be examined within the same detailed neuronal circuit. For artificial systems, this approach motivates treating input organization as a design variable in controllers that combine sensory feedback with task or phase cues. Such settings are particularly relevant when information overlaps across pathways but differs in reliability or availability. Our robotic experiment provides an initial proof of concept: context availability during learning changes subsequent closed-loop behavior under common execution conditions. This opens a route from cellular mechanisms to computational principles for context-sensitive control.

Limitations and future directions. The present study focuses on learning under fixed task rules. A natural next step is to examine how dendritic input organization shapes adaptation when rules change or context cues vary in reliability. Such settings would help clarify how the learned dependence on context evolves with experience and influences subsequent control.

AI USE STATEMENT

AI tools were used during manuscript preparation to assist with writing, editing, and language polishing, and to support literature retrieval and the discovery of potentially relevant prior work. The authors retained control over the scientific content and take full responsibility for the accuracy and integrity of the manuscript, including all claims, interpretations, conclusions, and references.

REPRODUCIBILITY STATEMENT

We document the experimental procedures and evaluation protocols in the appendices. Appendix A specifies the controlled task, input encoding, neuronal models, numerical settings, optimization, and statistical aggregation. Appendices B–C describe the context controls, parameter interventions, and shared sensory learning. Appendix D details the CoffeePush demonstrations, controller training, paired continuations, and closed-loop evaluation. Appendix E specifies the calcium-conductance interventions and training–execution comparisons. These descriptions include the training budgets, initialization and pairing procedures, and evaluation conditions needed to interpret and reproduce the reported experiments.

REFERENCES

Allen Institute. Allen Cell Types Database: Pvalb-IRES-Cre VISp layer 5 model 471085845. Allen Institute for Brain Science, 2015. URL https://modeldb.science/184152. ModelDB accession 184152.

Maximilian Baronig and Robert Legenstein. Context association in pyramidal neurons through local synaptic plasticity in apical dendrites. *Frontiers in Neuroscience*, 17:1276706, 2024. doi: 10.3389/fnins.2023.1276706. URL https://doi.org/10.3389/fnins.2023.1276706.

Brendan A. Bicknell and Michael Häusser. A synaptic learning rule for exploiting nonlinear dendritic computation. *Neuron*, 109:4001–4017.e10, 2021. doi: 10.1016/j.neuron.2021.09.044. URL https://doi.org/10.1016/j.neuron.2021.09.044.

Joseph Cichon and Wen-Biao Gan. Branch-specific dendritic Ca^{2+} spikes cause persistent synaptic plasticity. *Nature*, 520:180–185, 2015. doi: 10.1038/nature14251. URL https://doi.org/10.1038/nature14251.

Michael Deistler, Kyra L. Kadhim, Matthijs Pals, Jonas Beck, Ziwei Huang, Manuel Gloeckler, Janne K. Lappalainen, Cornelius Schröder, Philipp Berens, Pedro J. Gonçalves, and Jakob H. Macke. Jaxley: Differentiable simulation enables large-scale training of detailed biophysical models of neural dynamics. *Nature Methods*, 22:2649–2657, 2025. doi: 10.1038/s41592-025-02895-w. URL https://doi.org/10.1038/s41592-025-02895-w.

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

Timo Flesch, Keno Juechems, Tsvetomira Dumbalska, Andrew Saxe, and Christopher Summerfield. Orthogonal representations for robust context-dependent task performance in brains and neural networks. *Neuron*, 110(7):1258–1270.e11, 2022. doi: 10.1016/j.neuron.2022.01.005. URL https://doi.org/10.1016/j.neuron.2022.01.005.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673, 2020. doi: 10.1038/s42256-020-00257-z. URL https://doi.org/10.1038/s42256-020-00257-z.

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://doi.org/10.7554/eLife.22901.

Gan He, Kai Du, and Tiejun Huang. Going deeper with morphologically detailed neural networks by simulation-based gradient propagation. *Frontiers in Computational Neuroscience*, 20:1904220, 2026a. doi: 10.3389/fncom.2026.1904220. URL https://doi.org/10.3389/fncom.2026.1904220.

Gan He, Shenjian Zhang, Kai Du, and Tiejun Huang. Online synaptic credit assignment in active dendrites. *bioRxiv*, 2026b. doi: 10.64898/2026.09.14.751357. URL https://www.biorxiv.org/content/10.64898/2026.09.14.751357v1. Version 1, posted September 18, 2026.

James B. Heald, Máté Lengyel, and Daniel M. Wolpert. Contextual inference underlies the learning of sensorimotor repertoires. *Nature*, 600:489–493, 2021. doi: 10.1038/s41586-021-04129-3. URL https://doi.org/10.1038/s41586-021-04129-3.

Nizar Islah, Guillaume Etter, Mashbayar Tugsbayar, Busra Tugce Gurbuz, Blake Richards, and Eilif B. Muller. Learning to combine top-down context and feed-forward representations under ambiguity with apical and basal dendrites. *Cerebral Cortex*, 35(6):bhaf134, 2025. doi: 10.1093/cercor/bhaf134. URL https://doi.org/10.1093/cercor/bhaf134.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

M. E. Larkum, T. Nevian, M. Sandler, A. Polsky, and J. Schiller. Synaptic integration in tuft dendrites of layer 5 pyramidal neurons: A new unifying principle. *Science*, 325(5941):756–760, 2009. doi: 10.1126/science.1171958. URL https://doi.org/10.1126/science.1171958.

Matthew E. Larkum, J. Julius Zhu, and Bert Sakmann. A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398:338–341, 1999. doi: 10.1038/18686. URL https://doi.org/10.1038/18686.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Junfeng Lu, Zijie Yu, Shaoyang Cui, Gan He, Ruiqin Xiong, Kai Du, and Tiejun Huang. HelioX: A GPU-native framework for simulation and training of biophysically detailed networks. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=XjTRtfxK9g.

Hiroshi Makino and Takaki Komiyama. Learning enhances the relative impact of top-down processing in the visual cortex. *Nature Neuroscience*, 18:1116–1122, 2015. doi: 10.1038/nn.4061. URL https://doi.org/10.1038/nn.4061.

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://doi.org/10.1038/nature12742.

Eduardo Maristany de las Casas, Kris Killmann, Moritz Drüke, Lukas Münster, Christian Ebner, Robert Sachdev, Dieter Jaeger, and Matthew E. Larkum. Tuft dendrites in frontal motor cortex enable flexible learning. *Science*, 392(6798):eadx4358, 2026. doi: 10.1126/science.adx4358. URL https://doi.org/10.1126/science.adx4358.

S. Onasch, C. Miehl, M. M. Miękus, and J. Gjorgjieva. Assembly-based computations through contextual dendritic gating of plasticity. *Neuron*, 2026. doi: 10.1016/j.neuron.2026.07.028. URL https://doi.org/10.1016/j.neuron.2026.07.028. Advance online publication, August 21, 2026.

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. URL https://doi.org/10.1038/s41586-024-08433-6.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11671. URL https://doi.org/10.1609/aaai.v32i1.11671.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C. Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1256–1272, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/0987b8b338d6c90bbedd8631bc499221-Abstract.html.

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

Alon Polsky, Bartlett W. Mel, and Jackie Schiller. Computational subunits in thin dendrites of pyramidal cells. *Nature Neuroscience*, 7:621–627, 2004. doi: 10.1038/nn1253. URL https://doi.org/10.1038/nn1253.

Mattia Rigotti, Omri Barak, Melissa R. Warden, Xiao-Jing Wang, Nathaniel D. Daw, Earl K. Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497:585–590, 2013. doi: 10.1038/nature12160. URL https://doi.org/10.1038/nature12160.

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://proceedings.neurips.cc/paper_files/paper/2018/hash/1dc3a89d0d440ba31729b0ba74b93a33-Abstract.html.

Jackie Schiller, Guy Major, Helmut J. Koester, and Yitzhak Schiller. NMDA spikes in basal dendrites of cortical pyramidal neurons. *Nature*, 404:285–289, 2000. doi: 10.1038/35005094. URL https://doi.org/10.1038/35005094.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9573–9585, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6cfe0e6127fa25df2a0ef2ae1067d915-Abstract.html.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL https://www.jmlr.org/papers/v15/srivastava14a.html.

Naoya Takahashi, Thomas G. Oertner, Peter Hegemann, and Matthew E. Larkum. Active cortical dendrites modulate perception. *Science*, 354(6319):1587–1590, 2016. doi: 10.1126/science.aah6066. URL https://doi.org/10.1126/science.aah6066.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5–6):544–557, 2009. doi: 10.1016/j.neunet.2009.06.042. URL https://doi.org/10.1016/j.neunet.2009.06.042.

Willem A. M. Wybo, Matthias C. Tsai, Viet Anh Khoa Tran, Bernd Illing, Jakob Jordan, Abigail Morrison, and Walter Senn. NMDA-driven dendritic modulation enables multitask representation learning in hierarchical sensory processing pathways. *Proceedings of the National Academy of Sciences*, 120(32):e2300558120, 2023. doi: 10.1073/pnas.2300558120. URL https://doi.org/10.1073/pnas.2300558120.

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. URL https://doi.org/10.1038/ncomms12815.

Guangyu Robert Yang, Madhura R. Joglekar, H. Francis Song, William T. Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22:297–306, 2019. doi: 10.1038/s41593-018-0310-2. URL https://doi.org/10.1038/s41593-018-0310-2.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100, 2020. URL https://proceedings.mlr.press/v100/yu20a.html.

Yichen Zhang, Gan He, Lei Ma, Xiaofei Liu, J. J. Johannes Hjorth, Alexander Kozlov, Yutao He, Shenjian Zhang, Jeanette Hellgren Kotaleski, Yonghong Tian, Sten Grillner, Kai Du, and Tiejun Huang. A GPU-based computational framework that bridges neuron simulation and artificial intelligence. *Nature Communications*, 14:5798, 2023. doi: 10.1038/s41467-023-41553-7. URL https://doi.org/10.1038/s41467-023-41553-7.

A Detailed experimental methods

A.1 Task and input encoding

Input directions are $\theta \in \{0°, 120°, 240°\}$. Rule categories $c = 0, 1, 2$ have encoding centers $\psi_c = 0°, 120°, 240°$ and target rotations $\phi_c = 60°, 180°, 300°$, respectively. The four-dimensional observation is $(\cos\theta, \sin\theta, \cos\psi_c, \sin\psi_c)$; the target is defined in Section 3. The nine combinations yield three target directions. In particular, the encoded rule angle ψ is not the target rotation ϕ.

Each observation coordinate is represented by ten sensory population channels, giving 20 direction channels and 20 rule channels. For each observation coordinate x_d, normalization uses its mean μ_d and standard deviation s_d across the nine training conditions. The normalized value is clipped to $[-4, 4]$, and channel $k = 0, \ldots, 9$ has Gaussian tuning

$$z_d = \mathrm{clip}\left(\frac{x_d - \mu_d}{s_d}, -4, 4\right), \qquad r_{d,k} = 5 + 95\exp\left[-\frac{(z_d - c_{d,k})^2}{2\sigma_d^2}\right] \text{ Hz}.$$

Centers are equally spaced over the fitted normalized range. For the present discrete training conditions, the ranges are $[-0.707107, 1.414214]$ for the cosine coordinates and $[-1.224745, 1.224745]$ for the sine coordinates; thus $c_{d,k} = a_d + k(b_d - a_d)/9$. Normalization gives $\mu_d \simeq 0$ and $s_d \simeq 0.707107$ for all four coordinates. The width is the median adjacent-center spacing, with a minimum of 0.15: $\sigma_d = 0.235702$ for cosine and 0.272166 for sine coordinates. These fitted parameters remain fixed during training and evaluation; reported values are rounded. Context consists of three groups of ten channels. Under matching context, the rate of each channel in group g is

$$r_g = \begin{cases} 30\,\text{Hz}, & g = c, \\ 5\,\text{Hz}, & g \neq c. \end{cases}$$

Nonselective context sets every context channel to $13\frac{1}{3}$ Hz, whereas conflicting context cyclically shifts the active group to $(c + 1) \bmod 3$. These three conditions have an expected total rate of 400 events/s, or 140 events over the 350-ms input window. Sensory and context events use a discrete-time Poisson approximation: each channel independently emits an event with probability $r\Delta t$ per 0.1-ms step, with rate and time step expressed in consistent units. Separate random streams are used for sensory and context channels. Thus, repeated presentations of the same direction–rule condition retain the same encoded rates but use different sampled spike trains. Context withdrawal removes all context events. Sensory and background events are preserved across execution conditions.

A.2 Morphology, input routing, and initial population firing

Excitatory neurons use the model of Larkum et al. (2009) (ModelDB 124043). Inhibitory neurons use Allen perisomatic model 471085845, derived from a mouse primary visual cortex layer-5 Pvalb-IRES-Cre cell (ModelDB 184152; cell/experiment ID 313861608) (Allen Institute, 2015).

Sensory connection probability is one and recurrent connection probability is 0.1. Sensory and recurrent contacts lie on basal dendrites. Every excitatory neuron receives all context channels, assigned across three dendritic clusters. Inhibitory neurons receive no context input. In the apical-context configuration, the clusters are located on apical dendrites while the basal-context configuration uses the basal dendritic clusters. Figure 1 illustrates the input regions schematically.

Sensory-weight initialization is specified in Appendix A.3. The apical-context and basal-context configurations are compared at common relative initial context strengths $s_C = 1$, 1.4, and 1.8. The route-specific comparison uses $s_C = 1.8$ for apical context and $s_C = 1$ for basal context, yielding comparable initial population firing. Within each paired seed, sensory and recurrent weights and readout parameters share the same initialization. Figure S2A illustrates the pretraining population firing of these configurations. Initial population matching does not equate individual-cell responses or local dendritic states, and activity is not constrained to remain matched during learning. Thus the route-specific comparison combines dendritic location with activity-aligned initial context strength.

Within each paired seed, sensory and recurrent connection masks, contact locations, and cell identities are matched across routes, in addition to the initial weights and readout parameters.

A.3 Dynamics, optimization, and numerical settings

Synaptic-weight initialization and units. We report synaptic amplitudes as $G = |w| \times 1\,\mu\mathrm{S}$, where w is the signed, dimensionless synaptic multiplier and each incoming event has reference weight $1\,\mu\mathrm{S}$. Conductances are in $\mu\mathrm{S}$, membrane voltage in mV, and synaptic current in nA. For positive w, the normalized AMPA and NMDA kernels have amplitude coefficients $G/(1+r)$ and $rG/(1+r)$, respectively, with $r = 2$ for excitatory target neurons and $r = 1$ for inhibitory target neurons.

Sensory connections are initialized with positive sign with probability 0.8 and negative sign otherwise. Their initial amplitudes follow $G_S \sim \mathcal{U}(0, 3.6864\,\mathrm{nS})$ for connections to excitatory neurons and $G_S \sim \mathcal{U}(0, 1.8432\,\mathrm{nS})$ for connections to inhibitory neurons. The same sampled sensory weights are used within each paired seed.

Context connections onto excitatory neurons are positive. We define the reference initialization $G_{C,\mathrm{ref}} \sim \mathcal{U}(0, 1.2\,\mathrm{nS})$ and write $G_C(0) = s_C G_{C,\mathrm{ref}}$, where s_C is the dimensionless relative initial context strength. The common strengths $s_C = 1$, 1.4, and 1.8 therefore give $\mathcal{U}(0, 1.20\,\mathrm{nS})$, $\mathcal{U}(0, 1.68\,\mathrm{nS})$, and $\mathcal{U}(0, 2.16\,\mathrm{nS})$, respectively. The activity-compensated calcium condition uses $s_C = 2$, giving $\mathcal{U}(0, 2.40\,\mathrm{nS})$. This notation describes initialization; synaptic weights subsequently change through training.

For existing recurrent connections, initial conductance magnitudes are sampled uniformly from 0 to $9\,\mathrm{nS}$ for excitatory targets and from 0 to $4.5\,\mathrm{nS}$ for inhibitory targets. The sign is positive for excitatory presynaptic neurons and negative for inhibitory presynaptic neurons. Recurrent connection masks are sampled with probability 0.1 and remain fixed during training.

Receptor kinetics and current convention. For receptor α, an event arriving at t_m contributes the unit-peak double-exponential kernel

$$K_\alpha(u) = \frac{e^{-u/\tau_{d,\alpha}} - e^{-u/\tau_{r,\alpha}}}{e^{-t_{p,\alpha}/\tau_{d,\alpha}} - e^{-t_{p,\alpha}/\tau_{r,\alpha}}}\,\mathbf{1}_{u\geq 0}, \qquad t_{p,\alpha} = \frac{\tau_{r,\alpha}\tau_{d,\alpha}}{\tau_{d,\alpha} - \tau_{r,\alpha}} \log \frac{\tau_{d,\alpha}}{\tau_{r,\alpha}}.$$

The rise/decay constants (ms) and reversal potentials (mV) are AMPA: $(0.3, 3, 0)$; NMDA: $(2, 65, 0)$; and GABA: $(1, 20, -80)$. With $Q_\alpha(t) = \sum_m K_\alpha(t - t_m)$, positive weights give $g_\mathrm{A} = GQ_\mathrm{A}/(1+r)$ and $g_\mathrm{N} = rGQ_\mathrm{N}B(V)/(1+r)$, where

$$B(V) = \left[1 + 0.280\exp(-0.062\,V/\mathrm{mV})\right]^{-1}.$$

Negative weights select GABA, with $g_\mathrm{G} = GQ_\mathrm{G}$ and no AMPA/NMDA contribution.

Table S1 summarizes numerical and optimization settings. After each update, context weights are projected to nonnegative values and recurrent weights retain the sign of their presynaptic cell type. Sensory weights are not constrained to retain their initial signs and may cross zero during learning. Connection masks are fixed. Recurrent learning-rate groups in Table S2 are indexed by *postsynaptic* excitatory/inhibitory cell type; this differs from the presynaptic sign constraint.

Table S1: Controlled-task simulation and optimization settings.

Setting	Value
Total simulation / time step	500 ms / 0.1 ms
Initial voltage / initialization interval	-80 mV / 0–50 ms
Sensory and context input	50–400 ms
Spike-count readout window	$50 \leq t < 500$ ms
Background	50 nominal 2-Hz sources per cell
Readout	40-neuron spike counts; $\hat{y} = \tanh(W_O h + b_O)$
Training budget	500 epochs, 9 updates each; 4,500 total
Adam	$\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
Weight decay	10^{-4}, excluded for readout bias
Readout initialization	$(W_O)_{ji}(0) \sim \mathcal{N}(0, 0.05^2/40)$; $b_O(0) = 0$

Each epoch presents all nine conditions once in a pseudorandom order generated from the data-order seed plus the epoch index. Paired routes use the same condition order.

During training, sensory and context spike trains are resampled each epoch using the discrete-time scheme in Appendix A.1. Neural state is reset between trials. The readout vector $h \in \mathbb{R}^{40}$ contains the spike count of each neuron over $50 \le t < 500$ ms. The matrix $W_O \in \mathbb{R}^{2\times 40}$ and bias $b_O \in \mathbb{R}^2$ map these counts to two action coordinates, with $\tanh$ applied separately to each coordinate. The initial readout weights have mean zero and standard deviation $0.05/\sqrt{40}$; the second argument of $\mathcal{N}$ in Table S1 denotes variance.

During simulation, each cell receives 50 background sources with a firing rate of 2 Hz. Each source drives one Exp2Syn contact with rise/decay time constants of 1/20 ms. A signed weight is sampled uniformly from $[-0.1, 0.1]$ nS; its magnitude sets the conductance and its sign selects a reversal potential of 0 or -80 mV. Background contacts are sampled uniformly over segment indices across basal and apical dendrites for excitatory cells and over dendritic segments for inhibitory cells. Background contacts and amplitudes are not trained.

Table S2: Controlled-task learning rates by parameter group. Recurrent groups are indexed by postsynaptic cell type.

Parameter group	Learning rate
Sensory to excitatory / inhibitory neurons	9×10^{-6} / 3×10^{-6}
Recurrent excitatory / inhibitory groups	9×10^{-6} / 3×10^{-6}
Context to excitatory neurons	6.4453125×10^{-6}
Context to inhibitory neurons	0 (no connections)
Readout weights / bias	3×10^{-4} / 10^{-5}

A.4 Interventions and parameter handling

The context execution conditions are defined in Appendix A.1. The context-free training reference receives no context from initialization.

Sensory and recurrent exchanges replace only the selected weight arrays at corresponding presynaptic-channel or cell, postsynaptic-cell, and contact indices. The recipient retains its connection masks, anatomical contact sites, all nonexchanged parameters, and native readout. Shared sensory learning averages the paired sensory gradients before one common Adam update, while contextual, recurrent, and readout parameters retain separate optimizer states (Appendix A.6). The calcium training–execution comparison is specified in Appendix E; other intrinsic parameters remain unchanged.

A.5 Outcomes, evaluation support, and statistical units

Vector normalized mean squared error (NMSE) is $|\mathcal{D}|^{-1}\sum_i \|\hat{y}_i - y_i\|^2/\|y_i\|^2$; targets have unit norm. The training loss averages error over the two action coordinates and therefore equals half the vector NMSE. Angular error is the unsigned shortest prediction-to-target angle, averaged within each network. The primary routing and context-control analyses use the two-vector $\mathrm{atan2}$ expression. Across apical and basal routes at $s_C = 1, 1.4, 1.8$, all four context execution conditions, and all three seeds (20 realizations of each of nine conditions), outputs have nonzero norm (minimum 0.00623); no observations are excluded by output norm.

For the primary controlled-task evaluations, sampled input spike trains provide 20 realizations per condition for the nine direction–rule conditions, yielding 180 trials per network and evaluation configuration. Evaluation spike trains are generated separately from those used during training and are paired across the compared configurations. Performance is averaged over these trials within each network before aggregation across the three paired network seeds.

Execution-condition differences are computed within each network. Routing and training-history contrasts are computed within paired seeds before equal-weight aggregation across three seeds. Error bars report the sample standard deviation across these network-level values. Percentage gap reductions compare group-mean gaps defined in Section 3.

A.6 HELIOX EXTENSION AND GRADIENT-BASED TRAINING

We extended HelioX (Lu et al., 2026) to support replay-based synaptic gradient learning using the gradient-current/gradient-voltage algorithm of He et al. (2026b), connecting the spike-count action readout to synaptic gradient computation and the shared sensory update below.

Loss and gradient computation. For the two-dimensional target y and prediction $\hat{y}$ defined in Appendix A.3, training minimizes

$$\mathcal{L} = \tfrac{1}{2}\|\hat{y} - y\|_2^2. \tag{A.1}$$

With δ_z defined in Section 3, readout gradients are $g_{W_O} = \delta_z h^\top$ and $g_{b_O} = \delta_z$. The controlled-task implementation uses raw upward threshold crossings, $h_i = \sum_{n\in\mathcal{W}} \mathbf{1}\{V_{i,n-1} < -25\,\mathrm{mV},\ V_{i,n} \geq -25\,\mathrm{mV}\}$, where $\mathcal{W}$ is the 50–500-ms readout window on the 0.1-ms grid. It replaces the derivative of this count with a voltage-dependent surrogate. Define $[a]_0^1 = \min(1, \max(0, a))$, $V_r = -80$ mV and $\Delta V = |-25-(-80)| = 55\,\mathrm{mV}$. The base derivative and excitatory-cell floor are

$$s_0(V) = \frac{1}{\Delta V}\left[\frac{V - V_r}{\Delta V}\right]_0^1, \qquad s_f(V) = \frac{0.1}{\Delta V}\left[\frac{V - (V_r - 20\,\mathrm{mV})}{20\,\mathrm{mV}}\right]_0^1.$$

The derivative used for credit assignment is $s_i(V) = \max\{s_0(V), s_f(V)\}$ for excitatory cells and $s_i(V) = s_0(V)$ for inhibitory cells. The floor fades to zero between -80 and -100 mV. The additional s_f term preserves a learning signal for excitatory neurons in low-voltage, non-spiking states. With $\delta_h = W_O^\top \delta_z$, the voltage credit supplied to the learner is

$$\eta_{i,n} = \mathbf{1}_{n\in\mathcal{W}}\, \delta_{h,i}\, s_i(V_{i,n}).$$

There is no division by window duration or sample count, and no extra Δt multiplier in this count-to-voltage conversion. Forward features remain integer spike counts; the floor changes only the backward signal.

Synaptic gradients and optimization. Synaptic weight gradients are computed using the replay-based synaptic gradient method of He et al. (2026b), with the somatic voltage credit η as the backward input. The resulting gradients update sensory, context and recurrent synaptic weights once per trial.

Before Adam, $10^{-4}w$ is added to each trainable synaptic/readout-weight gradient; the readout bias is excluded. Adam uses the rates in Appendix A.3, followed by the sign projections and connection masks specified there. Neural state resets between trials, while learned weights and optimizer moments persist throughout training.

Shared sensory learning. In Fig. 3E–G, Separate denotes independently updated sensory weights and Shared denotes sensory weights tied throughout training. Both conditions use separately learned contextual, recurrent, and readout parameters. For paired circuits with tied sensory weights S, we average their sensory gradients before a single Adam update,

$$\bar{g}_S = \tfrac{1}{2}\big(g_{S,\mathrm{apical}} + g_{S,\mathrm{basal}}\big), \tag{A.2}$$

and apply the updated S to both circuits using a shared sensory Adam state. Contextual, recurrent, and readout parameters are optimized separately in each circuit. The two 40-cell circuits each receive credit from their own loss; private gradients retain their original scale. Averaging regularized sensory gradients adds $10^{-4}S$ once.

B COMPLETE ROUTING AND CONTEXT CONTROLS

Figure S1 extends the corresponding-neuron examples in Fig. 1E with population rasters, apical dendritic voltages, and context-withdrawal conditions.

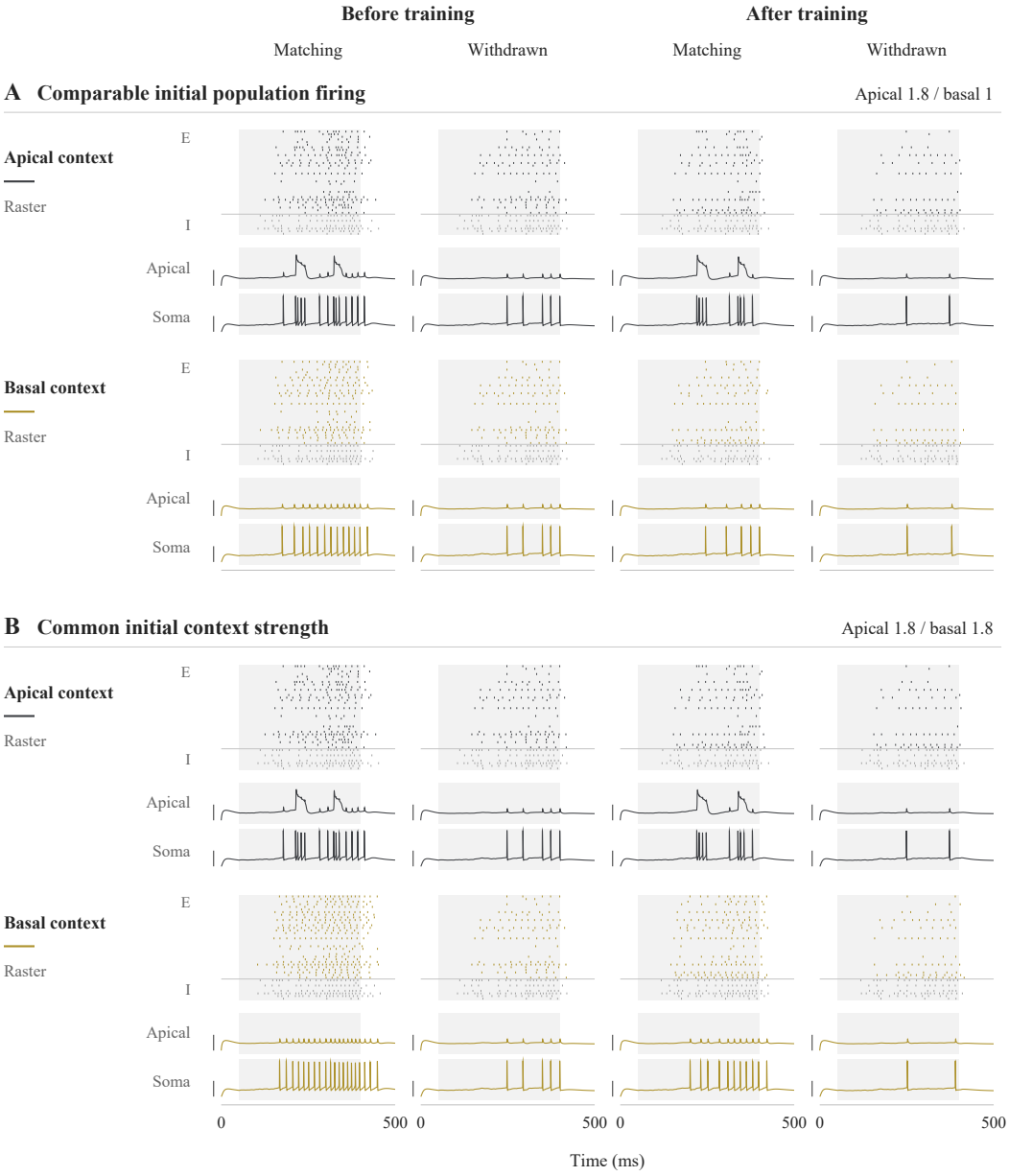


Figure S1: **Population rasters and corresponding-neuron voltages. (A)** Circuits with comparable initial population firing, with apical/basal relative context strengths $s_C = 1.8/1$. **(B)** Circuits with a common initial context strength, with $s_C = 1.8$ for both routes. Columns show matching or withdrawn context before and after training. Each raster contains 32 excitatory (E) and eight inhibitory (I) neurons in fixed order; ticks mark upward somatic crossings of -25 mV. Dark gray and gold identify apical- and basal-context circuits; light-gray ticks identify inhibitory neurons. Below each raster, apical dendritic and somatic voltages are shown for corresponding example neurons, with the same cell and recording sites throughout. All recorded voltage samples are retained without smoothing. Time and voltage scales are shared; vertical bars denote 50 mV and shading marks the 50–400 ms input window. These examples illustrate individual responses, rather than network-level effects.

Networks trained without context attain mean NMSE 0.04256 on the same evaluation panel. Figure S2 distinguishes initial activity alignment from learned activity: although firing is unconstrained during learning, withdrawal-induced population firing reductions at the trained endpoint are similar for the apical/basal $s_C = 1.8/1$ pair. The common-strength $1.8/1.8$ comparison produces a substantially larger basal-context reduction (panels B,D). These population summaries do not imply matched cellwise responses or local dendritic states.

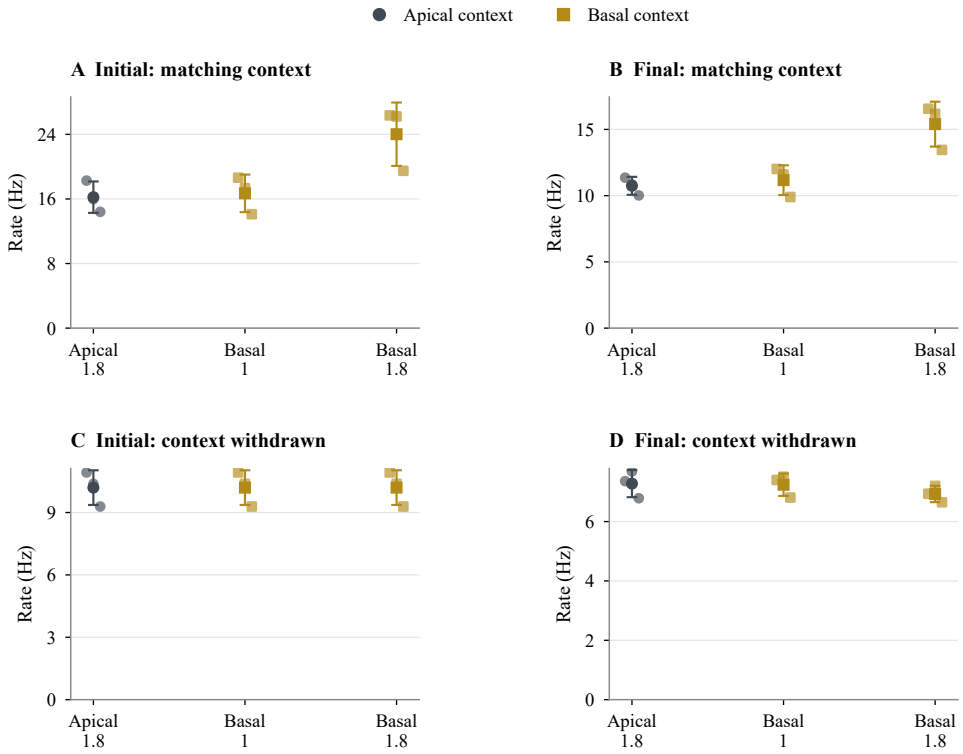


Figure S2: **Initial and learned activity.** Somatic spikes counted over 50–500 ms are divided by the 0.45 s readout window. The apical/basal $s_C = 1.8/1$ pair has comparable initial population firing; basal $s_C = 1.8$ provides the common-strength comparison. Each small point is one of three network seeds; large markers and whiskers give mean and sample SD.

Action geometry and signed direction changes. Figure S4A shows mean actions with matching or withdrawn context in the original output coordinates, without magnitude normalization. For a paired observation, let p_0 and p_1 denote the matching- and conflicting-context outputs, respectively, and let J rotate a vector counterclockwise by 90°. We compute

$$\Delta\alpha = \operatorname{atan2}(p_{0x}p_{1y} - p_{0y}p_{1x},\, p_0 \cdot p_1)\,, \qquad T_0 = (p_1 - p_0) \cdot J\frac{p_0}{\|p_0\|}. \tag{B.1}$$

Angles are reported in degrees. The false-rule target lies 120° counterclockwise from the true target, defining the positive direction. Positive rescaling of an output leaves $\Delta\alpha$ unchanged. If $p_1 = ap_0$ with $a > 0$, then $\Delta\alpha = T_0 = 0$. The tangential component still depends on the magnitude of p_1. Both measures are computed per observation and averaged within each network before aggregation over seeds. No output in this diagnostic had zero norm; the minimum matching- and conflicting-context output norms were 0.4033 and 0.3037, respectively, so the angular and tangential quantities were evaluated without a small-norm exclusion.

Conflict-induced angular changes average $3.48 \pm 0.96°$ for apical context and $4.59 \pm 1.94°$ for basal context; the paired basal-minus-apical difference is $1.11 \pm 1.01°$, positive in all three pairs. Corresponding tangential changes are 0.0551 ± 0.0237 and 0.0757 ± 0.0290 (Fig. S4B–C).

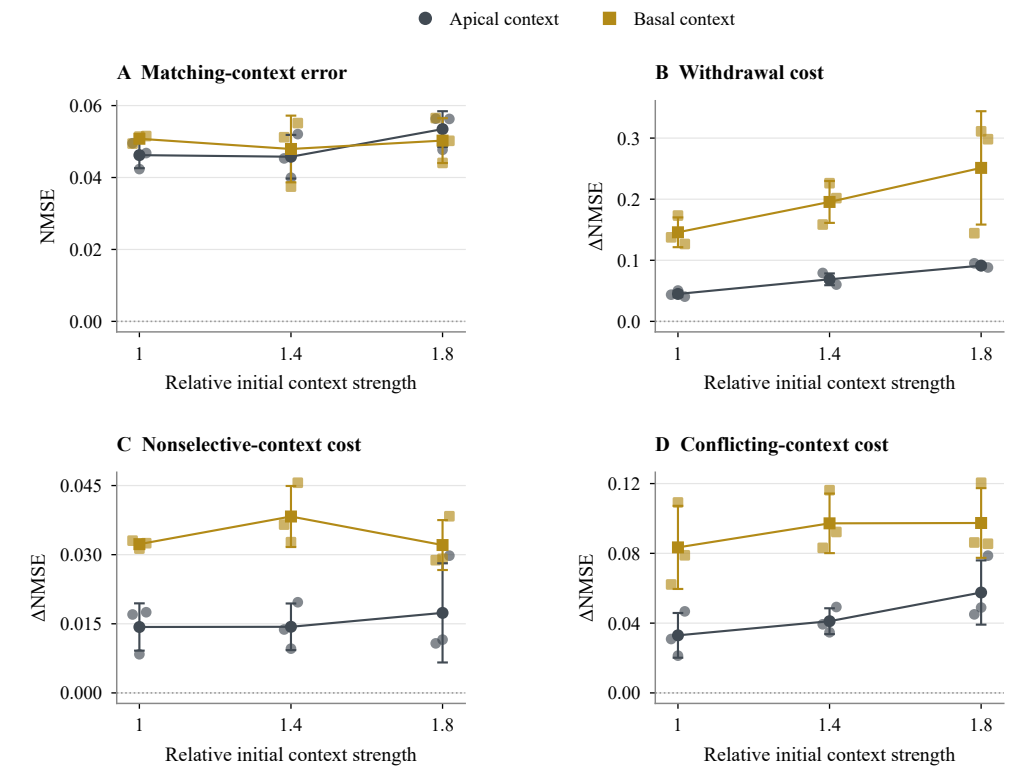


Figure S3: **Common-strength context controls.** Each configuration includes three trained networks. Panels B–D subtract the same network's matching-context error. Nonselective and conflicting context preserve the expected input-event budget.

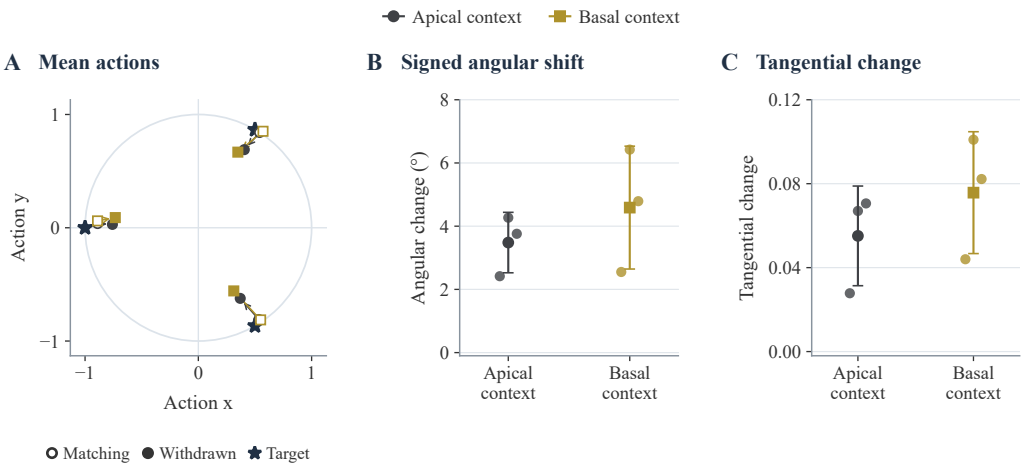


Figure S4: **Mean action geometry and conflict-induced direction changes.** All panels use the apical/basal $s_C = 1.8/1$ pair. **(A)** Mean actions with matching (open) or withdrawn (filled) context; stars mark targets and arrows connect paired means. For each network and context condition, actions are averaged over the 20 event realizations and the three direction–rule conditions sharing each target. The resulting target-specific means are then averaged equally across the three network seeds. **(B)** Signed angular change from matching to conflicting context, positive along the false-rule direction. **(C)** The same output change projected onto the tangent to the matching output direction. B–C show all three network seeds as small points and means ± sample SD as large points and bars.

For comparison, a chord-based projection of the output change is defined as

$$P = \left\langle (\hat{y}_{\text{conflicting}} - \hat{y}_{\text{matching}}) \cdot \frac{y_{false} - y}{\|y_{false} - y\|} \right\rangle . \tag{B.2}$$

It averages per observation before network aggregation, yielding means of 0.0871 for apical context and 0.1316 for basal context. Because this chord has an inward radial component, pure contraction from y to $0.8y$ gives $P = 0.1732$ without rotation; positive P alone therefore does not establish a directional turn. Nonselective context is an expected-rate control, so ratios of its behavioral effect to the withdrawal effect cannot identify a causal drive fraction.

Context-weight update conditions. Context-weight learning rates are multiplied by 0.5, 1, or 2 relative to 6.4453125×10^{-6}; all other parameter-group rates and the 500-epoch budget remain fixed. When context weights are frozen, context input remains present while context weights and their Adam moments are held fixed. Sensory, recurrent, and readout parameters continue learning. Each configuration uses seeds 1, 2, and 3, initial context strengths $s_C = 1.8$ for apical context and $s_C = 1$ for basal context, and 20 realizations per condition at evaluation.

Figure S5 shows post-withdrawal error and withdrawal cost when the same context learning-rate multiplier is applied to both configurations. The basal-minus-apical withdrawal-cost gaps are 0.04830, 0.05451, and 0.07600 for multipliers 0.5, 1, and 2, respectively. Freezing context weights yields a gap of 0.03741. The learned contrast depends on the context learning-rate setting and persists when context weights are frozen.

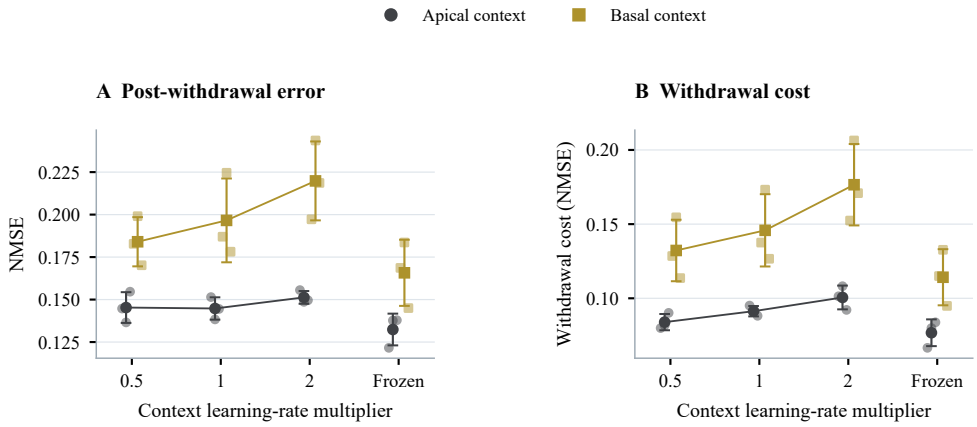


Figure S5: **Context-weight learning and context dependence.** A shows post-withdrawal NMSE and B shows withdrawal cost (post-withdrawal minus matching-context NMSE) for context learning-rate multipliers 0.5, 1, and 2, and frozen context weights. The other parameter groups continue learning under the same training budget. Large markers and bars show mean ± sample SD; small markers show all three network seeds. Context input remains present throughout training, including when context weights are frozen.

C SENSORY INTERVENTIONS AND READOUT COMPATIBILITY

C.1 RESPONSE GEOMETRY

Let $H_{r,c}$ be the event-averaged count vector for condition c. The corresponding-centroid distance is

$$d = \left[\frac{1}{9}\sum_{c=1}^{9} \|H_{\text{apical},c} - H_{\text{basal},c}\|_2^2\right]^{1/2}. \tag{C.1}$$

It retains raw 40D counts and neuron correspondence. Separate-to-shared sensory-learning distances are 2.83245 to 0.39826, 4.33119 to 0.59395, and 2.99166 to 0.37491 counts for seeds 1–3.

C.2 SENSORY AND RECURRENT WEIGHT EXCHANGES

Post-training exchanges replace the recipient's designated weights with those of the paired donor configuration at corresponding synapses. The recipient retains its morphology, synapse locations, other parameters, and trained readout, and the circuit is rerun after each exchange. We exchange all sensory weights, recurrent weights, or both groups. Direction-only and rule-only exchanges replace the two disjoint 20-channel sensory blocks: channels 0–19 encode $\cos\theta, \sin\theta$, and channels 20–39 encode $\cos\psi, \sin\psi$. The rule channels encode the rule category rather than the target rotation angle directly.

Figure S6 gives absolute errors and angular errors for the full sensory and recurrent exchanges. Table S3 additionally reports matching-context error, post-withdrawal error, and withdrawal cost for direction-only and rule-only exchanges, alongside each recipient's native baseline.

Table S3: Sensory-block exchanges with the recipient's native readout retained. **(A)** Absolute outcomes, mean ± sample SD across three networks; withdrawal cost is calculated within each network before averaging. **(B)** Per-seed changes in withdrawn-context NMSE, exchange minus the same recipient's native baseline. Positive values indicate increased error.

A. Absolute outcomes

Route	Exchange	E_{present}	$E_{\text{withdrawn}}$	D
Apical	Native	0.05346 ± 0.00498	0.14476 ± 0.00657	0.09130 ± 0.00350
Apical	Direction	0.06758 ± 0.01397	0.17378 ± 0.00438	0.10620 ± 0.01385
Apical	Rule	0.06168 ± 0.00882	0.17501 ± 0.01724	0.11333 ± 0.01186
Apical	All sensory	0.07611 ± 0.02552	0.19709 ± 0.02045	0.12098 ± 0.01168
Basal	Native	0.05078 ± 0.00117	0.19659 ± 0.02466	0.14582 ± 0.02436
Basal	Direction	0.05371 ± 0.00307	0.17999 ± 0.02094	0.12628 ± 0.02381
Basal	Rule	0.06319 ± 0.02235	0.18297 ± 0.01775	0.11978 ± 0.00533
Basal	All sensory	0.06333 ± 0.01808	0.15867 ± 0.01161	0.09534 ± 0.00734

B. Per-seed changes in withdrawn-context NMSE

	Apical recipient		Basal recipient	
Seed	Direction-only	Rule-only	Direction-only	Rule-only
1	$+0.02658$	$+0.02771$	-0.01549	-0.01926
2	$+0.03568$	$+0.05145$	-0.02077	-0.02216
3	$+0.02480$	$+0.01159$	-0.01355	$+0.00057$

Nonadditivity of direction and rule contributions. For each context condition, define $\Delta E_K = E_K - E_{\text{native}}$, where K denotes the exchanged parameter block. We quantify nonadditivity as

$$R_E = \Delta E_S - \Delta E_{\text{direction}} - \Delta E_{\text{rule}}.$$

The corresponding withdrawal-cost residual is $R_D = R_{E,\text{withdrawn}} - R_{E,\text{matching}}$. Figure S7 reports these residuals for post-withdrawal error and withdrawal cost. The residuals can have either sign: exchanging the full sensory block need not reproduce the sum of the two separate block effects because the circuit and readout are nonlinear. They do not represent additive causal fractions.

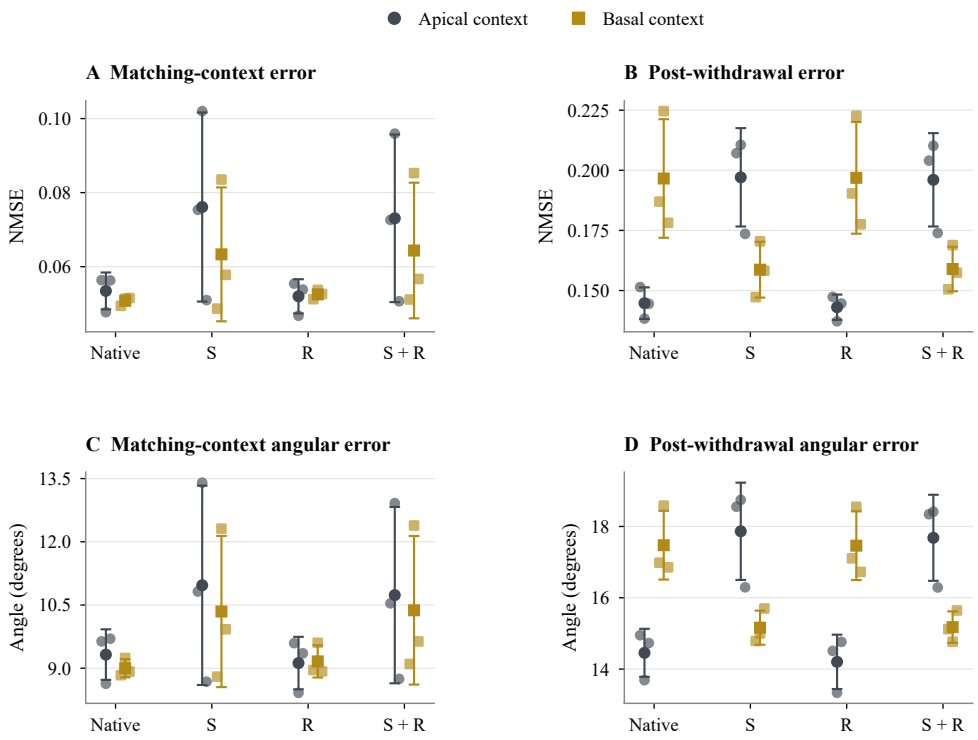


Figure S6: **Absolute exchange outcomes.** S, R, and S+R replace sensory, recurrent, or both parameter groups with those of the other route in the same seed. The recipient's native readout is retained. Errors and angles are reported separately for matching and withdrawn context; the exchanges re-execute the circuit without learning. The unchanged native state supplies the comparison for the changes shown in Figure 3.

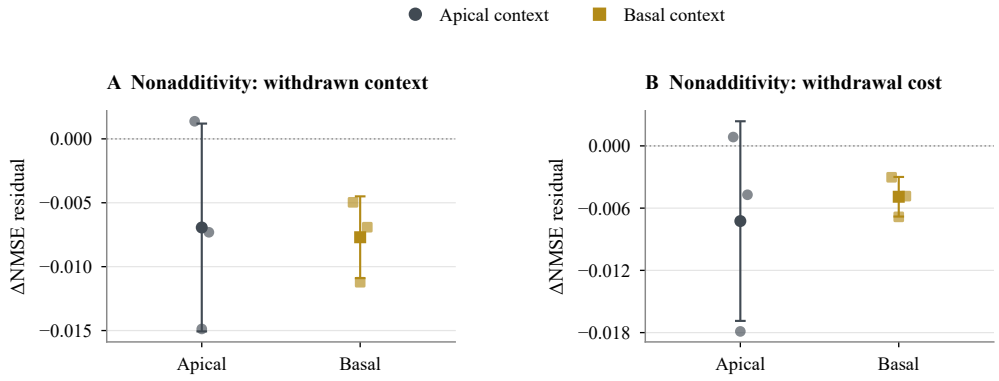


Figure S7: **Nonadditive block contributions.** The behavioral residual in A–B is the full-S exchange effect minus the direction-only and rule-only effects, relative to the same native state. These parameter interventions establish functional contributions; their effects are not additive causal fractions of a unique mechanism.

C.3 Cross-evaluation of recorded responses with fixed trained readouts

For each paired seed, we apply the two trained readouts to the recorded responses from each configuration, keeping both responses and readout parameters fixed (Fig. S8C–D; Table S4). After shared sensory learning, applying the apical-context readout to basal-context responses reduces

post-withdrawal NMSE from 0.17318 to 0.15742 on average, with improvement in all three seeds. The same substitution after separate sensory learning has little effect (0.19659 to 0.19605). Thus, the remaining behavioral difference after shared sensory learning also depends on how the learned readouts transform neural responses into actions.

Table S4: Saved responses passed through the two existing readouts; NMSE mean ± sample SD. No readout was fitted in this diagnostic.

Training	Mode	Activity	Apical readout	Basal readout
Separate	Matching	Apical	0.05346 ± 0.00498	0.06800 ± 0.01052
Separate	Matching	Basal	0.09183 ± 0.05826	0.05078 ± 0.00117
Separate	Withdrawn	Apical	0.14476 ± 0.00657	0.15896 ± 0.00927
Separate	Withdrawn	Basal	0.19605 ± 0.01941	0.19659 ± 0.02466
Shared	Matching	Apical	0.05577 ± 0.00436	0.05531 ± 0.00358
Shared	Matching	Basal	0.05615 ± 0.00459	0.04497 ± 0.00194
Shared	Withdrawn	Apical	0.15599 ± 0.00098	0.17067 ± 0.00923
Shared	Withdrawn	Basal	0.15742 ± 0.00144	0.17318 ± 0.01098

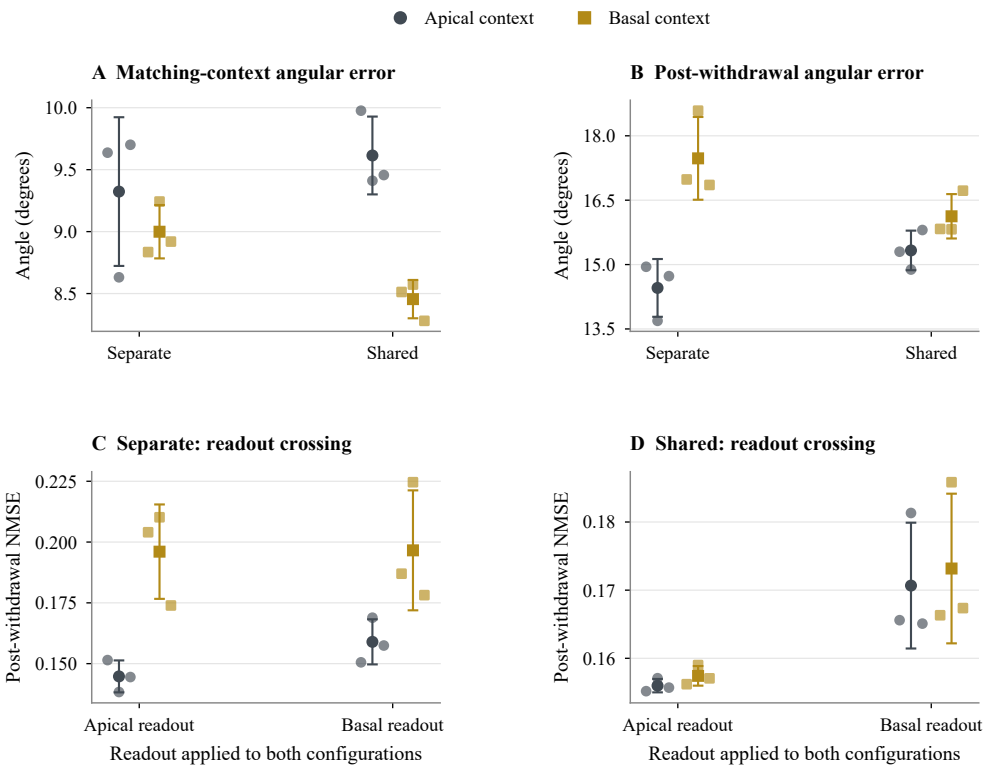


Figure S8: **Shared learning and native-readout compatibility.** A–B give angular error with each circuit's native readout. For C–D, the two trained readouts are applied offline to withdrawn-context responses generated by the original recurrent circuits, without refitting or rerunning the circuits.

D COFFEEPUSH METHODS AND SUPPLEMENTARY RESULTS

D.1 TASK, TRAINING, AND EVALUATION

Demonstrations and observations. Twenty successful demonstrations from a scripted expert in MetaWorld `coffee-push-v3` contain 1,080 observation–action pairs. The 11-dimensional observation comprises the three-dimensional positions of the end effector, object, and goal, together with $x_{\text{object}} - x_{\text{goal}}$ and $x_{\text{end effector}} - x_{\text{object}}$. Only current-frame coordinates enter this observation. Each frame is one training sample and receives one optimizer update per epoch, with frames shuffled across demonstrations; thus 50 epochs give 54,000 updates. Input, recurrent, and readout parameters are jointly trained. Table S5 summarizes the changes from the controlled task.

Observation encoding. Observation means and standard deviations are fitted across all 1,080 training frames. For each coordinate, a fitted standard deviation below 10^{-6} is replaced by 1 before standardization. Standardized values are clipped to $[-4, 4]$ and encoded by ten Gaussian channels per dimension, using the 5–100-Hz tuning function in Appendix A.1. Centers are uniformly spaced over a range fitted to each normalized training coordinate. Widths equal the median adjacent-center spacing, with minimum 0.15. Normalization, centers, and widths remain fixed during training and evaluation; rollout observations do not enter their fitting. The resulting 110 sensory channels and 30 phase-context channels supply a recurrent circuit of 32 excitatory and 8 inhibitory neurons.

Neural state and optimization. Before every training and evaluation decision, simulator initialization restores membrane voltage to -80 mV and reinitializes channel, synaptic, and other dynamic states. Recurrence operates within each 200-ms simulation, but neural state does not persist between environment steps. The external history used to construct the phase cue is maintained separately. Adam settings, weight decay, and learning rates follow Appendix A.3.

Table S5: Controlled-task and CoffeePush biological-circuit methods. Both use a recurrent 32 E/8 I circuit, a 0.1-ms integration step, raw-count features, and an affine readout followed by $\tanh$. Shared optimizer settings and the CoffeePush action transformation are specified in Appendix D.1; reference-model differences are in Appendix D.5.

Method item	Controlled task	CoffeePush
Observation / input channels	4 coordinates; 40 sensory + 30 context	11 coordinates; 110 sensory + 30 context
Simulation / input / count windows	500 / 50–400 / 50–500 ms	200 / 20–150 / 20–200 ms
Neural-state boundary	Reset before each trial	Reset before each environment decision
Action / training loss	2 direction coordinates; two-coordinate MSE	3 displacement + 1 gripper; four-coordinate normalized MSE
Samples / update budget	9 conditions per epoch; $500 \times 9 = 4{,}500$ updates	1,080 frames per epoch; $50 \times 1{,}080 = 54{,}000$ updates
Initial context amplitude	$\mathcal{U}(0, 1.2 s_C\,\text{nS})$; s_C specified per comparison	Apical: $\mathcal{U}(0, 2.1\,\text{nS})$; basal: $\mathcal{U}(0, 0.84\,\text{nS})$
Readout initialization	Zero-mean normal, SD $0.05/\sqrt{40}$; zero bias	Zero-mean normal, SD $1/\sqrt{40}$; zero bias

The route-specific initial context amplitudes in Table S5 were calibrated to match initial population firing between apical- and basal-context controllers.

Four-dimensional action interface. Let $h \in \mathbb{R}^{40}$ contain raw neuron spike counts over 20–200 ms. The native readout is $\hat{a}^{\text{norm}} = \tanh(W_O h + b_O)$, with $W_O \in \mathbb{R}^{4\times 40}$ and $b_O \in \mathbb{R}^4$. Its outputs are three displacement commands and a continuous gripper command. For each action coordinate j, training-demonstration extrema define $m_j = \min a_j$ and $s_j = \max a_j - m_j$ (with $s_j = 1$ for

spans below 10^{-6}). Targets and loss are

$$a_j^{\mathrm{norm}} = 2\frac{a_j - m_j}{s_j} - 1, \qquad \mathcal{L} = \frac{1}{4}\sum_{j=1}^{4}\big(\hat{a}_j^{\mathrm{norm}} - a_j^{\mathrm{norm}}\big)^2.$$

Execution applies the inverse transformation $\hat{a}_j = m_j + s_j(\hat{a}_j^{\mathrm{norm}} + 1)/2$ before passing the action to the environment.

Phase cue and spike encoding. Let $\mathbf{h}_t$, $\mathbf{o}_t$, and $\mathbf{g}_t$ denote the end-effector, object, and goal positions in the horizontal XY plane, in meters. Define hand–object distance $d_t = \|\mathbf{h}_t - \mathbf{o}_t\|_2$ and progress toward the goal $p_t = \|\mathbf{o}_0 - \mathbf{g}_0\|_2 - \|\mathbf{o}_t - \mathbf{g}_t\|_2$. The geometric phase is

$$\tilde{q}_t = \begin{cases} 2 \text{ (push)}, & p_t \geq 0.01, \\ 1 \text{ (contact)}, & p_t < 0.01 \text{ and } d_t \leq 0.07, \\ 0 \text{ (reach)}, & \text{otherwise.} \end{cases}$$

Training labels use $\tilde{q}_t$ computed within each demonstration before frames are shuffled. In all 20 training demonstrations (1,080 frames), these labels are already nondecreasing, so applying the cumulative-maximum rule would leave every training label unchanged. During online evaluation, the supplied phase is $q_t = \max(q_{t-1}, \tilde{q}_t)$, with $q_{-1} = 0$ at each rollout start, so phase cannot move backward. The initial object–goal distance and the online phase history are retained externally; the cue therefore uses information beyond the current observation alone.

The 30 context channels form three groups of ten, corresponding to reach, contact, and push. At each decision, the selected group's channels have rate 30 Hz; the other twenty channels have rate 5 Hz. Sensory and context rates generate Poisson spike trains over 20–150 ms, resampled across training epochs. Context withdrawal removes events from all 30 context channels.

Closed-loop evaluation. Each evaluation restores the same saved set of 50 distinct full simulator initial states, including positions, velocities, and task state, across checkpoints and compared execution conditions. Each rollout lasts at most 120 environment steps and is counted as successful if the environment's success flag is reached within that horizon; it stops on success or environment termination/truncation. Context events are either supplied or omitted; a separate reference is trained without context from initialization. Evaluation occurs at every even-numbered epoch from 10 through 50. Success rates are averaged across these 21 checkpoints within each network, then summarized as the mean and sample SD across three independently trained networks per configuration. Figure 5B reports success. The within-network withdrawal cost is defined as context-present minus context-withdrawn success (Table S7). The paired-continuation checkpoint window is specified separately in Appendix D.4.

D.2 Current-observation phase decoding

We decode the training phase labels (Appendix D.1) from current 11-dimensional observations in the shared set of 20 expert demonstrations (1,080 frames). A class-balanced linear softmax classifier is evaluated by five-fold cross-validation, holding out four complete demonstrations per fold. Feature standardization and class weights are fitted only on the other 16 demonstrations. Pooled held-out balanced accuracy, the mean of the three phase recalls, is 97.7%.

The high decoding accuracy indicates substantial overlap between phase labels and current sensory observations on the expert-demonstration distribution. The phase cue nevertheless incorporates trajectory history and provides an explicit reach/contact/push organization of these observations. We therefore describe it as partially redundant with sensory input: its task-stage information is substantially accessible from current observations, while its construction also uses information beyond the current frame. This interpretation does not establish strict single-frame redundancy or generalization to controller-generated trajectories.

D.3 Activity and maintained calcium blockade

Figure S9A compares checkpoint-averaged firing rates with context present and withdrawn, using the 21 checkpoints at every even-numbered epoch from epoch 10 through epoch 50. Context

withdrawal lowers mean firing in both configurations. These activity summaries complement the behavioral withdrawal costs in Table S7.

A separate comparison blocks the slow calcium conductance in excitatory apical dendrites throughout both training and evaluation (Fig. S9B–C). The apical-context withdrawal cost is 5.11 ± 4.85 percentage points, compared with 1.37 ± 4.99 for intact apical-context controllers and 3.46 ± 6.50 for intact basal-context controllers (mean $\pm$ sample SD across three networks). Because blockade is maintained during both training and evaluation, this comparison does not separate training-history effects from acute execution effects.

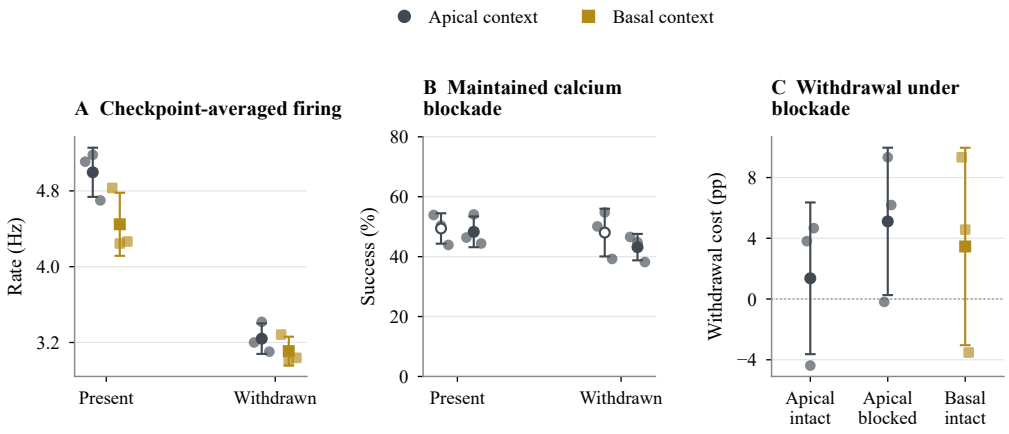


Figure S9: **Firing and maintained-blockade controls.** A shows checkpoint-averaged firing under present and withdrawn context. B compares intact-trained/intact-executed (open) with blocked-trained/blocked-executed (filled) apical-context controllers. C gives the corresponding withdrawal costs, with the intact-trained/intact-executed basal-context reference. A–C average over the 21 checkpoints at every even-numbered epoch from epoch 10 through epoch 50.

D.4 Paired continuations

Paired continuation design. Each of six epoch-20 source controllers, three per configuration, is duplicated with identical weights and Adam states into branches that retain or withdraw context during continued training through epoch 50. Each branch receives another 32,400 updates. Training samples and sensory events are paired. In the withdraw-context branch, all context event rates are set to zero, but context weights remain in the original optimization path. Their Adam states are restored at the fork and continue to update, and the same weight regularization and nonnegativity projection are applied. Both branches are evaluated with context present and withdrawn at matched checkpoints, initial states, and rollout indices. Training effects are computed as retain-context minus withdraw-context success within each source controller under the same execution condition (Fig. 5C). This design tests the effect of context during learning after epoch 20.

Evaluation summary. Evaluation is performed at every even-numbered epoch from epoch 22 through epoch 50. The continuation summary averages success rates across these 15 checkpoints within each network. Table S6A reports success under both execution conditions and the within-network withdrawal cost for each continuation history. Panel B reports the paired training effects under common execution conditions, including each source seed. Effects are positive in all three apical-context sources and negative in all three basal-context sources under context-present execution; signs are mixed in both configurations under context-withdrawn execution. These comparisons concern continued learning after epoch 20 and do not test benefits of earlier context exposure.

D.5 Model references

The reference ANN and SNN controllers are trained on the same 20 CoffeePush demonstrations (1,080 observation–action pairs) and 11-dimensional observation definition as the biological cir-

Table S6: CoffeePush paired continuations. **(A)** Absolute success and execution withdrawal cost (context-present minus context-withdrawn success). **(B)** Paired training effects: retain-context minus withdraw-context training success under the same execution condition, in percentage points (pp). Each source value averages 15 checkpoints (epochs 22, 24, ..., 50); summary values are means ± sample SD across three source networks per route. Panel B SDs are computed from the unrounded paired effects.

A. Absolute success and execution withdrawal cost

Route	Continuation context	Execution success (%) Present	Withdrawn	Withdrawal cost (pp)
Apical	Retain	52.49 ± 2.53	47.33 ± 8.49	5.16 ± 6.69
Apical	Withdraw	43.07 ± 9.19	43.82 ± 4.31	-0.76 ± 9.41
Basal	Retain	49.16 ± 3.09	44.89 ± 3.04	4.27 ± 6.13
Basal	Withdraw	51.29 ± 1.87	42.89 ± 7.06	8.40 ± 5.60

B. Paired training effects under common execution

Route	Execution context	Seed 1	Seed 2	Seed 3	Mean ± SD
Apical	Present	$+3.47$	$+19.07$	$+5.73$	$+9.42 \pm 8.43$
Basal	Present	-1.60	-1.20	-3.60	-2.13 ± 1.29
Apical	Withdrawn	$+3.73$	$+8.27$	-1.47	$+3.51 \pm 4.87$
Basal	Withdrawn	-1.33	-5.87	$+13.20$	$+2.00 \pm 9.96$

cuits. Each configuration has three independently trained networks, seeds 1–3, trained for 50 epochs. Table S7 reports their results; Table S8 specifies architecture, parameter counts, optimization, and temporal treatment. Equal epoch counts do not imply equal optimizer-update counts or equal temporal memory.

For the biological circuits, apical-context success after withdrawal exceeds the independently trained context-free reference by 6.13 ± 9.59 percentage points (mean ± sample SD of same-seed differences across three networks, after averaging the 21 checkpoints at epochs 10, 12, ..., 50 within each network). This descriptive comparison varies across networks and does not establish a reliable benefit of context exposure for subsequent context-free control.

Table S7: CoffeePush model references. Success (%) and withdrawal cost (percentage points) are means ± sample SD across three networks. Withdrawal gives zero context input: absent events in spiking models and zero normalized channel features in ANNs (Appendix D.5). The first two rows repeat the same independently trained context-free reference.

Model	Success (%) Present	Withdrawn	Context-free	Withdrawal cost (pp)
Apical context	49.40 ± 5.07	48.03 ± 7.96	41.90 ± 5.78	1.37 ± 4.99
Basal context	45.84 ± 4.10	42.38 ± 2.40	41.90 ± 5.78	3.46 ± 6.50
RNN	35.11 ± 0.96	20.83 ± 15.53	26.48 ± 10.16	14.29 ± 14.58
LSTM	41.52 ± 8.51	0.48 ± 0.67	40.60 ± 11.68	41.05 ± 7.85
GRU	43.14 ± 6.91	12.10 ± 13.20	38.29 ± 16.81	31.05 ± 12.23
FF-SNN	36.41 ± 2.42	26.76 ± 11.84	30.35 ± 4.05	9.65 ± 11.03
RSNN	33.21 ± 1.87	14.10 ± 2.75	30.86 ± 2.24	19.11 ± 3.58

Observation, context, and action encoding. All model families use the training-fitted observation encoding and four-dimensional action normalization described in Appendix D.1. The fixed encoder supplies 110 sensory rates between 5 and 100 Hz; the phase rule supplies 30 context rates between 5 and 30 Hz. RNN, LSTM, and GRU receive continuous rate features: sensory features are $(r - 5)/95$, clipped to $[0, 1]$, and context features are $(r - 5)/25$. Context-trained ANNs concatenate 110 sensory and 30 context features; their separately trained context-free references have 110 inputs. The 32-unit context-recipient set is chosen to match the biological circuits, where con-

Table S8: CoffeePush model configurations. Parameter counts are allocated scalar entries in optimized weight/bias tensors, shown as context-trained / independently context-free. Fixed encoders and neuronal dynamics are excluded. The counts include entries disabled by connectivity masks, so they do not imply matched numbers of free parameters.

Model	Hidden architecture	Weight/bias entries	Optimizer and temporal treatment
Apical / basal circuit	One recurrent 32 E/8 I population; four-output readout	7,364 / 6,164	Adam, rates in Appendix A.3; one decision/update; neural state reset each decision.
RNN	One 40-unit $\tanh$ recurrent layer	7,444 / 6,244	Adam, 3×10^{-4}; up to ten steps/update; state retained within trajectory.
LSTM	One 40-unit LSTM layer	29,284 / 24,484	Same ANN optimizer and sampling; hidden and cell states retained.
GRU	One 40-unit GRU layer	22,004 / 18,404	Same ANN optimizer and sampling; hidden state retained.
FF-SNN	One 40-unit LIF layer	5,804 / 4,604	AdamW, 2×10^{-4} input, 5×10^{-3} readout; one decision/update; state reset each decision.
RSNN	One recurrent 40-unit LIF layer	7,404 / 6,204	Same SNN settings; recurrent rate 2×10^{-4}; state reset each decision.

All references have four readout outputs. With context present, masking excludes 8×30 input weights per gate. All conditions use 20 demonstrations and 50 epochs. ANNs use 120 chunks/epoch (6,000 updates total); spiking references and biological circuits use 1,080 single-frame samples/epoch (54,000 updates). Encoding, loss, state resets, and evaluation interventions are detailed in the text.

text targets only the 32 excitatory neurons and excludes the eight inhibitory neurons. Accordingly, context input in the ANN references connects to the first 32 of 40 hidden units in each recurrent gate, with nonnegative weights, and is masked away from the remaining eight. The FF-SNN and RSNN use the same 110/140 input dimensions and context mask, but sample Poisson spikes directly from the rates. These baseline masks are dense onto the 32 context-recipient units; they match the recipient count without reproducing the biological circuits' dendritic routing. Each reference has a linear four-output readout followed by $\tanh$ and uses equal-weight mean squared error over the four normalized action coordinates, with the inverse action transformation in Appendix D.1 for execution.

Recurrent ANN architecture and training. The RNN uses one 40-unit $\tanh$ recurrent layer; LSTM and GRU each use one 40-unit gated layer, with no additional hidden readout layers or dropout. Training uses Adam with learning rate 3×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and no weight decay. Demonstration order is shuffled each epoch, while frames within each demonstration remain ordered. Batch size is one sequence, split into consecutive nonoverlapping chunks of at most ten environment steps, including the shorter final chunk. Mean squared action error is averaged over all steps and four outputs in each chunk. Hidden state is initialized to zero at the start of each demonstration, carried numerically between its chunks, and detached from the gradient graph after each optimizer step. LSTM carries both hidden and cell state; RNN and GRU carry hidden state. Thus backpropagation spans at most ten steps, but numerical memory can span the entire demonstration. The saved demonstration lengths give 120 chunks per epoch and 6,000 optimizer updates over 50 epochs. During evaluation, states start at zero for each rollout and persist across environment steps until rollout termination; they are not reset every ten steps.

Spiking-reference architecture and training. FF-SNN and RSNN each have one layer of 40 leaky integrate-and-fire units; RSNN additionally has a dense 40×40 recurrent weight matrix with-

out recurrent bias. The membrane time constant is 20 ms, threshold 0.615, and integration step 1 ms. Each decision simulates 200 steps, with Poisson input and mean-spike readout both spanning 0–200 ms. Training uses a triangular surrogate derivative with dampening 0.3 and backpropagation through this within-decision simulation. AdamW (Loshchilov & Hutter, 2019) uses learning rate 2×10^{-4} for input weights, input biases, and recurrent weights, and 5×10^{-3} for readout weights and biases; weight decay is 10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and the global gradient norm is clipped to 1. The loss is mean squared normalized action error. Frames are shuffled and processed individually with batch size one and an effective sequence length of one frame, giving 1,080 updates per epoch and 54,000 in total. Membrane voltage and previous-spike state are reset to zero before every decision in both training and evaluation; no neural state or gradient passes between environment steps.

Context withdrawal. Context withdrawal in Table S7 sets the context-channel input actually received by each model to zero. For the biological circuits, FF-SNN, and RSNN, all context events are removed. For RNN, LSTM, and GRU, every context feature is set to 0. The rollout support, 120-step horizon, success definition, checkpoint averaging, and within-network withdrawal-cost calculation follow Appendix D.1.

E CALCIUM TRAINING, EXECUTION, AND ACTIVITY COMPENSATION

E.1 MANIPULATION AND COMPLETE MAIN COMPARISON

Blockade sets the maximal slow, high-voltage-activated (HVA) calcium conductance to zero throughout the apical dendrites of all 32 excitatory neurons. This is the only voltage-gated calcium conductance there. Calcium-activated potassium channels, NMDA receptors and other intrinsic parameters are retained. The primary comparison preserves trainable initial weights within each route (apical/basal $s_C = 1.8/1$). Intact and blocked execution are crossed with both training histories while keeping learned weights and the native readout fixed.

For withdrawal cost, the training-history contrast at common execution e is $D_{blocked\text{-}trained,e} - D_{intact\text{-}trained,e}$. Acute blockade instead compares execution states of intact-trained networks. Table S9 reports every condition of the cross as mean $\pm$ sample SD over the three networks, including both execution states (Fig. S10). It also includes the separate apical $s_C = 2$ activity-compensated condition, which changes context initialization and cannot be treated as a pure conductance comparison. Basal context uses $s_C = 1$.

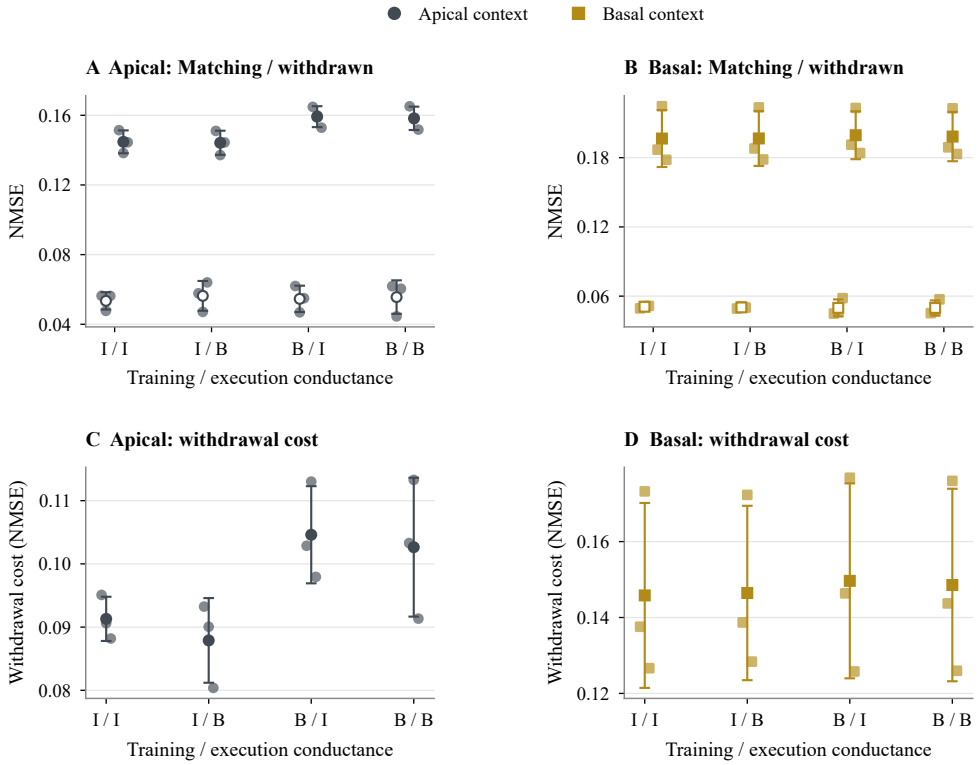


Figure S10: **Complete training-by-execution calcium cross.** I denotes intact and B blocked conductance. Open points in A–B are matching context, filled points withdrawn context; C–D show their paired difference. All evaluations use the apical/basal $s_C = 1.8/1$ pair and the same 20 realizations per condition. Blockade targets the slow HVA calcium conductance in excitatory apical dendrites.

E.2 INITIAL-ACTIVITY COMPENSATION

Under calcium blockade, the initial apical context strength was increased from $s_C = 1.8$ to $s_C = 2$ to approximately restore initial population firing (Fig. S11A). This compensation changes initial context strength as well as the conductance condition. Under restored execution, its mean post-withdrawal NMSE is 0.17356, compared with 0.15920 for blocked-conductance training from matched initial weights at $s_C = 1.8$ and 0.14476 for intact training at $s_C = 1.8$. With blockade

Table S9: Calcium cross summaries, mean ± sample SD over three networks. Apical strength 2 compensation is separate from the primary matched-initialization training-by-execution comparison.

Training	Route	Exec.	E_{present}	$E_{\text{withdrawn}}$	D
Intact	Apical	Intact	0.05346 ± 0.00498	0.14476 ± 0.00657	0.09130 ± 0.00350
Intact	Basal	Intact	0.05078 ± 0.00117	0.19659 ± 0.02466	0.14582 ± 0.02436
Intact	Apical	Blocked	0.05632 ± 0.00855	0.14421 ± 0.00693	0.08789 ± 0.00670
Intact	Basal	Blocked	0.05023 ± 0.00100	0.19669 ± 0.02378	0.14646 ± 0.02297
Blocked	Apical	Blocked	0.05561 ± 0.00962	0.15825 ± 0.00670	0.10264 ± 0.01099
Blocked	Basal	Blocked	0.04976 ± 0.00652	0.19832 ± 0.02138	0.14856 ± 0.02537
Blocked	Apical	Intact	0.05460 ± 0.00756	0.15920 ± 0.00600	0.10461 ± 0.00769
Blocked	Basal	Intact	0.04981 ± 0.00735	0.19947 ± 0.02078	0.14966 ± 0.02568
Blocked (2)	Apical	Blocked	0.05953 ± 0.00861	0.17406 ± 0.00319	0.11453 ± 0.00602
Blocked (2)	Apical	Intact	0.06169 ± 0.00819	0.17356 ± 0.00349	0.11187 ± 0.00578

maintained, the corresponding blocked-training $s_C = 2/1.8$ values are 0.17406/0.15825. Without an intact-trained $s_C = 2$ control, this comparison does not isolate a conductance-by-initial-context-strength interaction.

Initial population firing rates with matching context in the apical-context configuration over 50–500 ms are 16.219/15.518/16.785 Hz for intact $s_C = 1.8$, blocked $s_C = 1.8$, and blocked $s_C = 2$; endpoint rates are 10.743/10.323/10.865 Hz. Matching an initial rate does not match dendritic states or subsequent credit signals. Figure S11B compares withdrawal costs under common intact execution.

Blockade maintained during evaluation. Separately from the restored-conductance comparison in Section 4.3, blocked-trained/blocked-executed apical-context circuits have mean withdrawal cost 0.10264 at fixed initial strength $s_C = 1.8$ and 0.11453 at activity-compensated strength $s_C = 2$. The intact-trained/intact-executed references have costs 0.09130 for apical context at $s_C = 1.8$ and 0.14582 for basal context at $s_C = 1$ (Table S9). Both maintained-blockade configurations increase the apical cost and narrow its difference from the basal reference, but combine training and acute execution effects; the $s_C = 2$ comparison additionally changes initial context strength.

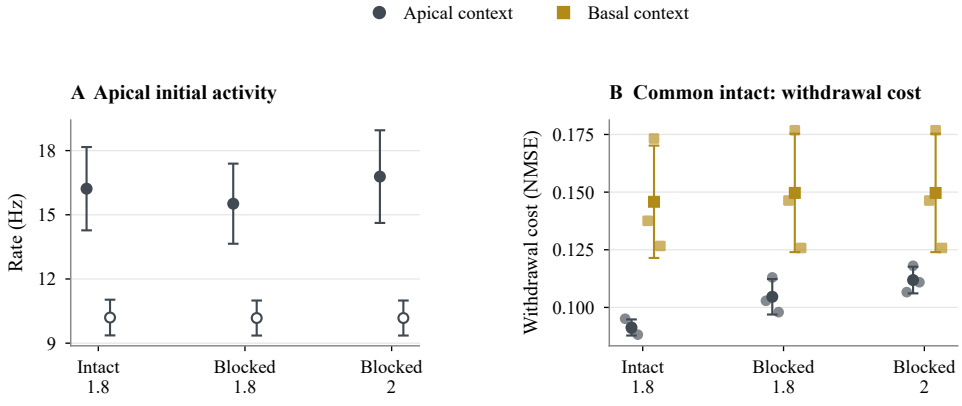


Figure S11: **Initial-activity compensation for calcium blockade.** A shows initial firing in apical-context circuits with matching (filled) or withdrawn (open) context. B compares withdrawal costs under common intact execution. Intact- and blocked-conductance training start from identical trainable parameters within each paired seed at apical $s_C = 1.8$; the activity-compensated blocked-training condition uses apical $s_C = 2$, with basal context fixed at $s_C = 1$. Points and bars summarize three networks as mean ± sample SD; small points in B show individual networks. Compensation changes context initialization as well as the conductance condition.

E.3 TRANSFER BETWEEN CONDUCTANCE-TRAINING HISTORIES

For each route and seed, intact-trained (I) and calcium-blocked-trained (B) endpoint networks exchange all sensory weights S or all recurrent weights R in both directions. The recipient retains its context weights, native readout, and every nonexchanged parameter. All evaluations use intact conductance, apical/basal $s_C = 1.8/1$ relative initial context strengths, and the same 20 realizations per condition as the primary results. Each exchange re-executes the circuit; no readout is refitted. This comparison uses the fixed-initialization calcium histories, separately from the activity-compensated apical $s_C = 2$ condition.

For a metric M, either post-withdrawal NMSE or withdrawal cost, define

$$T = M_B - M_I, \quad U_K = M_{I \leftarrow K_B} - M_I, \quad V_K = M_B - M_{B \leftarrow K_I}, \quad K \in \{S, R\}. \tag{E.1}$$

Figure 4C plots U_K for B-to-I transfer and $-V_K$ for I-to-B transfer, so that both transfer directions are expressed as exchange-minus-recipient changes. Figure S12 includes T and both transfer directions for post-withdrawal NMSE and withdrawal cost. Table S10 reports absolute matching/withdrawn-context errors and paired withdrawal costs for every recipient history and exchange state as mean $\pm$ sample SD over three networks.

For apical-context circuits, mean S-transfer changes in post-withdrawal NMSE are $+0.05255$ and -0.02323, with these directions in all three seeds; the corresponding R changes are $+0.00028$ and $+0.00337$. For basal-context circuits, mean S changes are $+0.00330$ and -0.00317, each with the corresponding sign in two seeds; R changes are $+0.00086$ and -0.00102. Apical-context withdrawal-cost changes for S are $+0.03741$ and -0.03258, compared with the native blocked-minus-intact history effect $T = +0.01331$. Transfer effects exceeding T reflect dependence on the recipient's jointly learned parameters; ratios to T do not identify unique mediated fractions. Small R-exchange effects do not show that recurrent dynamics are unimportant.

Table S10: Absolute conductance-history exchange outcomes under intact execution, mean $\pm$ sample SD over three networks. Training denotes the recipient history; S/R replace that group with the paired opposite-history donor. Withdrawal cost is computed within network before aggregation.

Route	Training	Exchange	E_{present}	$E_{\text{withdrawn}}$	D
Apical	Intact	Native	0.05346 ± 0.00498	0.14476 ± 0.00657	0.09130 ± 0.00350
Apical	Intact	S	0.06859 ± 0.02127	0.19731 ± 0.04284	0.12871 ± 0.02232
Apical	Intact	R	0.05190 ± 0.00399	0.14503 ± 0.00545	0.09314 ± 0.00264
Apical	Blocked	Native	0.05460 ± 0.00756	0.15920 ± 0.00600	0.10461 ± 0.00769
Apical	Blocked	S	0.06394 ± 0.00878	0.13597 ± 0.00792	0.07203 ± 0.01126
Apical	Blocked	R	0.05563 ± 0.01074	0.16257 ± 0.01217	0.10694 ± 0.00967
Basal	Intact	Native	0.05078 ± 0.00117	0.19659 ± 0.02466	0.14582 ± 0.02436
Basal	Intact	S	0.05033 ± 0.00370	0.19989 ± 0.02011	0.14956 ± 0.01886
Basal	Intact	R	0.05069 ± 0.00118	0.19745 ± 0.02582	0.14676 ± 0.02569
Basal	Blocked	Native	0.04981 ± 0.00735	0.19947 ± 0.02078	0.14966 ± 0.02568
Basal	Blocked	S	0.04915 ± 0.00414	0.19630 ± 0.02549	0.14715 ± 0.02865
Basal	Blocked	R	0.04957 ± 0.00681	0.19845 ± 0.01992	0.14888 ± 0.02393

E.4 ROUTING-GAP SUMMARY UNDER INTACT EXECUTION

The unequal error increases partially narrow the routing contrast. With intact conductance at evaluation, the post-withdrawal error gap decreases from 0.05184 to 0.04027, and the withdrawal-cost gap from 0.05451 to 0.04505 (Fig. S13), corresponding to reductions of 22.3% and 17.4% in the group-mean gaps. Narrowing primarily reflects the larger increase in apical-context withdrawal error, weakening its prior advantage. This differs from shared sensory learning, where basal-context post-withdrawal performance improves as apical-context performance worsens (Fig. 3E).

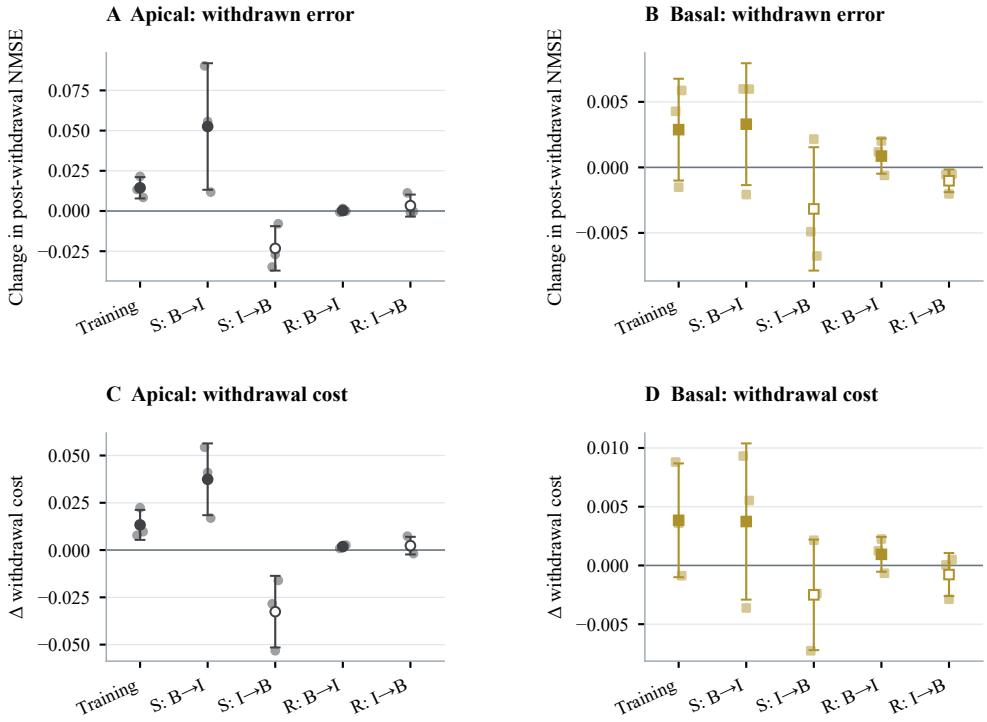


Figure S12: **Complete conductance-history and parameter-transfer contrasts.** I and B denote intact- and blocked-conductance training histories; all evaluations use intact conductance. Training shows the native B-minus-I history effect T. B-to-I transfers show U_K and I-to-B transfers show $-V_K$, relative to each recipient's native baseline, for S and R separately. A–B show post-withdrawal NMSE changes; C–D show changes in withdrawal cost, withdrawn minus matching-context NMSE. Large markers and bars summarize three seeds as mean ± sample SD; small markers show individual seeds. The complete absolute outcomes are in Table S10.

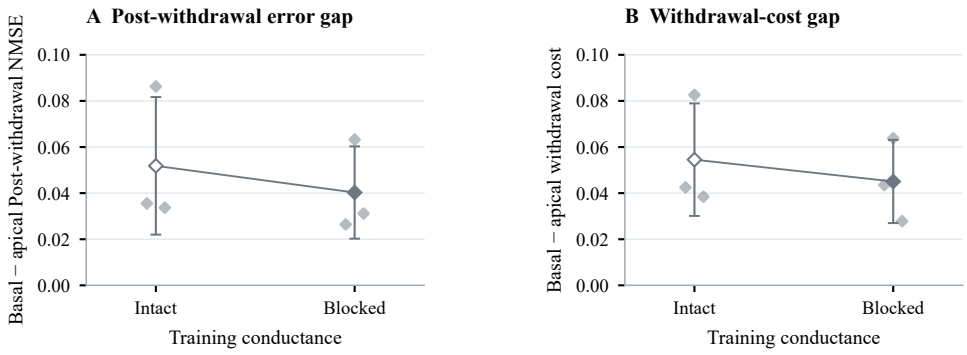


Figure S13: **Conductance-training effects on the routing gap.** All evaluations use intact conductance. A shows the basal-minus-apical post-withdrawal NMSE gap; B shows the basal-minus-apical withdrawal-cost gap. Open and filled diamonds indicate intact and blocked conductance during training. Large points and bars give mean ± sample SD across the three paired seeds; small points show individual seed pairs, and lines connect group means. The gap reduction primarily reflects a larger increase in apical-context error after blocked training.