Cortico-Cerebellar Coordination Flexibly Recombines Reusable Representations for Zero-Shot Compositional Generalization

Anonymous authors
Paper under double-blind review

Abstract

Compositionality, the ability to flexibly recombine learned components to solve novel tasks, is a hallmark of biological intelligence and a central challenge for artificial systems. However, the circuit mechanisms and dynamical principles underlying compositional generalization remain poorly understood. Inspired by distributed cortico-cerebellar architecture, we develop a computational framework that couples recurrent cortical dynamics with adaptive cerebellar feedback. We evaluate this framework across cognitive and motor tasks requiring zero-shot execution of novel rule combinations and motor primitive sequences. While cortex-only models achieve accurate performance on learned compositions but fail to recombine previously acquired components in unseen contexts, cortico-cerebellar models exhibit robust zero-shot compositional generalization across both domains. Mechanistic analyses reveal that cortex-only models retain task-relevant latent representations but fail to flexibly deploy them under novel cognitive contexts and exhibit predecessor-dependent state carry-over during motor transitions. In contrast, cerebellar feedback stabilizes reusable cortical representations, steers successor dynamics, and adaptively regulates state transitions during novel compositional execution. Together, these mechanisms enable cortico-cerebellar models to achieve robust zero-shot compositional generalization across temporal and structural task variations. Our findings reveal a circuit-level mechanism for flexible compositional computation and establish a bio-inspired principle for developing artificial systems capable of zero-shot generalization.

1 Introduction

Cognitive and motor flexibility are fundamental capabilities underlying intelligent behavior, enabling agents to adapt to changing task demands, recombine previously learned skills, and generalize to new demands. Compositional computation provides a fundamental mechanism for flexible intelligence by reusing representations and computational components across tasks (Riveland et al., 2026; Tafazoli et al., 2026; Cole et al., 2011; Reverberi et al., 2012). This capability supports continual adaptation, knowledge transfer, and dynamic generalization, providing a computational basis for multi-task learning, continual learning, and representation learning (Collins et al., 2024; Fang et al., 2026; Mendez & Eaton, 2022; Devin et al., 2017; Ito et al., 2022). However, despite growing interest in compositional computation across artificial intelligence and neuroscience, the neural principles enabling flexible reuse and recombination in biological systems remain poorly understood.

Studies of multi-task learning in recurrent neural networks (RNNs) have begun to identify candidate representational and dynamical substrates for cortical compositional computation. A work has shown that task representations can develop compositional structure, with algebraic combinations of rule inputs supporting performance on some tasks (Yang et al., 2019). At the level of network dynamics, another work has identified shared dynamical motifs that implement computations reused across tasks and can be reconfigured to support rapid learning of new tasks (Driscoll et al., 2024). These findings suggest that

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.

reusable representations and dynamical operations can serve as building blocks for flexible computation. However, such reuse and rapid transfer do not by themselves establish reliable zero-shot generalization to unseen task combinations. Rule recombination succeeds for some task relationships but fails for others in (Yang et al., 2019), whereas the transfer experiments of (Driscoll et al., 2024) require learning new task-specific input weights and depend on the dynamical repertoire acquired during pretraining. These findings motivate investigating the circuit mechanisms and learning principles that enable reusable computations to be reliably recruited and combined across novel tasks.

Cortico-cerebellar interactions provide a promising candidate substrate for supporting compositional recombination and generalization. Anatomical tracing has identified closed-loop linking the cerebellum with motor and prefrontal cortex, suggesting that cerebellar can influence perception, action, and cognition (Kelly & Strick, 2003). Cerebellar perturbations influence frontal preparatory activity during motor planning and adaptation (Israely et al., 2025; Gao et al., 2018). Complementing circuit-level evidence, recent human imaging studies have implicated distributed cortico-cerebellar networks in compositional recombination across diverse cognitive contexts (Tan et al., 2025; Mill & Cole, 2025). Computational models further show that cerebellar feedback can facilitate cortical learning and sustain task-dependent dynamics (Boven et al., 2023; Pemberton et al., 2024). Together, these findings motivate investigating cortico-cerebellar interactions as a potential computational mechanism for compositional computation.

The main contributions of our work are as follows:

- Cerebellar adaptive feedback enables cortical compositional recombination and zero-shot generalization. We show that cortical networks learn reusable task representations but lack the dynamical flexibility for novel compositional recombination. The proposed cerebellar feedback mechanism adaptively regulates cortical states through content and gating pathways, where structured corrective signals reshape task-relevant representations and contextual gating controls feedback strength and timing. This closed-loop cortico-cerebellar interaction stabilizes cortical dynamics, regulates state transitions, and enables flexible recombination of learned components across novel cognitive rules and motor primitive sequences.
- Comprehensive analysis of multi-task compositional generalization. We evaluate the framework across cognitive and motor benchmarks with held-out task combinations. Cerebellar feedback enables context-dependent cortical representation transformation in cognitive tasks and regulates successor-state dynamics in motor tasks, allowing flexible recombination of learned components while preserving behaviorally relevant information.

2 Related works

Cortico-cerebellar coupling. Anatomical and physiological studies have revealed extensive reciprocal cortico-cerebellar loops involving motor and prefrontal cortices, forming a circuit substrate for coordinating perception, action, and cognition (Kelly & Strick, 2003). Experimental studies suggest that cerebellar circuits can shape cortical dynamics and computations through structured feedback pathways (Wagner et al., 2019). Computational frameworks have highlighted cerebellar feedback as a mechanism for predictive learning, temporal credit assignment, delayed feedback compensation, and cortical dynamic regulation under changing task demands (Boven et al., 2023; Pemberton et al., 2021; 2024; Avraham & Mawase, 2025). Recent works extend this view by suggesting that cortico-cerebellar coupling also supports flexible cognitive computations, including evidence-based decision making and anomaly detection (Bao et al., 2026; Zhang et al., 2026).

Complementing these circuit and computational findings, recent human imaging studies have implicated distributed cortico-cerebellar networks in compositional recombination across diverse cognitive contexts (Tan et al., 2025; Mill & Cole, 2025). Although some cortical RNN models have provided insights into representation reuse and generalization, they have primarily studied these abilities within fixed task structures (Yang et al., 2019; Driscoll et al.,

2024). The underlying biological mechanisms and computational principles that enable compositional computations remain unresolved, particularly how distributed cortico-cerebellar coordination contributes to flexible representation recombination and generalization across novel tasks.

Cerebellar function and computations. A central question for compositional behavior is how shared neural representations are recruited for different task demands. Theoretical work suggests that the cortico-pontine pathway can emphasize task-relevant dimensions of cortical activity before they reach the cerebellar granular layer, while the structure of granule-cell representations shapes the transformations available to downstream readouts (Muscinelli et al., 2023; Tang et al., 2021b; Xie et al., 2023; Tang et al., 2021a; Yu et al., 2026). Human imaging provides evidence consistent with task-dependent recruitment of cortical inputs to the cerebellum during both movement and working memory (Shahshahani et al., 2024). Across two learned skills, simultaneous recordings further reveal similar cortical dynamics alongside distinct, coherently reoriented granule-cell trajectories (Garcia-Garcia et al., 2026). These findings motivate the hypothesis that cerebellar computations can transform shared cortical activity according to task context. Whether this capacity supports the recombination of learned components in entirely unseen task configurations remains to be tested.

3 Method Overview

3.1 Cognitive and motor tasks

We evaluate zero-shot compositional generalization across cognitive and motor domains by testing the reuse of discrete rules and continuous movement primitives (Figure 1). In cognitive C-PRO, each task combines one of four logical, sensory and motor rules, yielding $4^3 = 64$ task contexts. Models are trained on a subset of rule combinations and tested on held-out combinations without further training. In the motor task, models learn sequences of movement primitives and are tested on unseen transitions, with combinatorial complexity increasing with sequence length (up to 64 combinations for four-step sequences). Full task details are provided in the Appendix A.

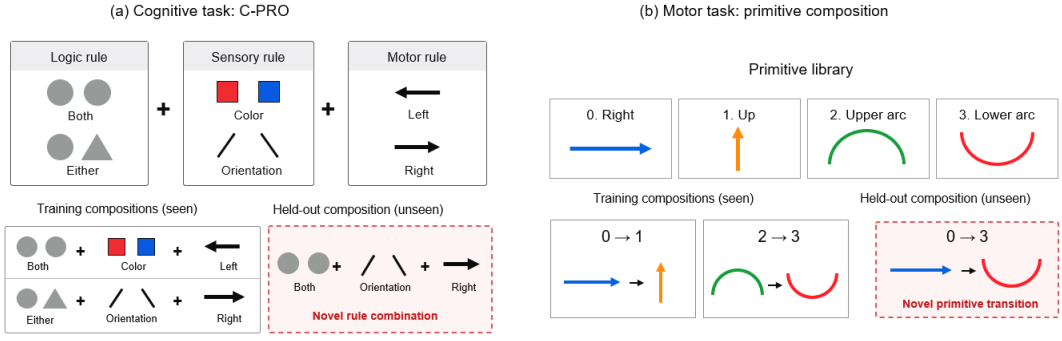


Figure 1: Two classes of compositional generalization tasks in cognitive and motor domains. (A) C-PRO combines logical, sensory, and motor rules into task contexts, with held-out contexts formed by unseen combinations. (B) The motor task combines movement primitives into sequential transitions, with held-out transitions formed by unseen combinations.

3.2 Cortex-only model

Cortical population activity is modeled as a leaky recurrent neural network (RNN) with hidden state $\mathbf{h}(t) \in \mathbb{R}^N$:

$$\begin{aligned} \mathbf{h}_{t+1} &= (1-\alpha)\mathbf{h}_t + \alpha\phi\big(W_{\text{rec}}\mathbf{h}_t + W_{\text{in}}\mathbf{x}_t + \mathbf{b}_h\big), \\ \mathbf{y}_t &= W_{\text{out}}\mathbf{h}_{t+1} + \mathbf{b}_{\text{out}}, \end{aligned} \tag{1}$$

where $\mathbf{x}_t$ denotes the task input. W_{in}, W_{rec}, W_{out}, and their corresponding biases $\mathbf{b}_h$, $\mathbf{b}_{\text{out}}$ are trainable parameters for input, recurrent and readout mappings, respectively. The leak parameter α controls state dynamics, and $\phi(\cdot) = \tanh(\cdot)$ is applied element-wise.

3.3 Cortico-cerebellar model

We model cortico-cerebellar coordination using a cortical recurrent network coupled to a feedforward cerebellar module (Figure 2). The architecture is inspired by the cortico-ponto-cerebellar pathway, in which cortical signals are transformed through pontine mossy fibers (MFs), granule cells (GrCs), Purkinje cells (PCs), and deep cerebellar nuclei (DCN), before feeding back to cortex via the thalamus. We captured these interactions as effective population-level transformations. The model captures cerebellar-inspired input integration, context-dependent transformation, and feedback regulation of cortical dynamics.

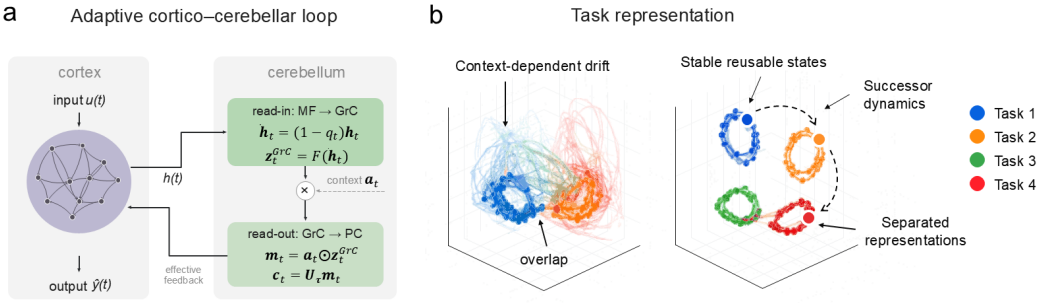


Figure 2: Adaptive cortico-cerebellar framework and task representations. (A) Closed-loop cortico-cerebellar architecture with content and gating feedback pathways. (B) Latent dynamics illustrating context-dependent transitions and stable reusable representations across task contexts.

The cerebellar module received the cortical state $\mathbf{h}_t$ and task input $\mathbf{x}_t$ as an effective MF input: $\mathbf{MF}_t^{\text{eff}} = [\mathbf{h}_t; \mathbf{x}_t]$. This captures cortical and contextual signals conveyed through pre-cerebellar pathways, consistent with evidence that GrCs encode diverse task-relevant information beyond immediate sensori-motor variables. The cerebellar module implemented two computational operations: adaptive regulation of cortical state retention and context-dependent transformation of task content.

$$
\begin{aligned}
\mathbf{z}_t^{\text{gate}} &= \tanh\Big(W_{\text{gate}}\mathbf{MF}_t^{\text{eff}} + \mathbf{b}_{\text{gate}}\Big)\,, \\
q_t &= \sigma\big(\mathbf{w}_q^\top \mathbf{z}_t^{\text{gate}} + b_q\big)\,, \qquad \overline{\mathbf{h}}_t = (1-q_t)\mathbf{h}_t, \\
\mathbf{c}_t &= \mathcal{C}_\theta^{(\tau)}\big(\overline{\mathbf{h}}_t, \mathbf{x}_t\big)\,, \\
\mathbf{z}_t^{\text{GrC}} &= \tanh\big(V_\tau \overline{\mathbf{h}}_t + W_{c,\tau}\mathbf{c}_t + \mathbf{b}_{m,\tau}\big)\,, \\
\mathbf{a}_t &= \mathcal{A}_\theta^{(\tau)}(\mathbf{x}_t), \qquad \mathbf{m}_t = \mathbf{a}_t \odot \mathbf{z}_t^{\text{GrC}}, \\
\mathbf{c}_t &= U_\tau \mathbf{m}_t.
\end{aligned}
\tag{2}
$$

Here, $W_{\text{gate}} = [G_h \;\; G_x]$, and σ denotes the logistic function. The scalar gate q_t modulates cortical state retention, producing the retained state $\tilde{\mathbf{h}}_t$. The task domain is indexed by $\tau \in \{\text{cog}, \text{mot}\}$, where the content map $C_\theta^{(\tau)}$ extracts task-relevant features and $A_\theta^{(\tau)}$ determines their modulation. The element-wise product $\odot$ combines the nonlinear feature representation $\mathbf{z}_t^{\text{GrC}}$ with the modulation vector $\mathbf{a}_t$ to generate the content signal $\mathbf{m}_t$. In the cognitive task, $\mathbf{c}_t$ links predicted intermediate decisions to response rules, while $\mathbf{a}_t$ restricts content feedback to the response period. In the motor task, $\mathbf{c}_t$ selects primitive-specific content according to the sequence slot, and $\mathbf{a}_t$ provides learned slot-dependent modulation. Thus, both tasks share the same computational framework while using task-specific content maps and modulation functions. For cerebellar information processing, nonlinear

transformations capture granule-cell-like feature expansion, while the gate provides adaptive regulation of cortical state persistence.

The resulting cortical update is

$$\mathbf{h}_{t+1} = (1-\alpha)\mathbf{h}_t + \alpha\phi(W_{\text{rec}}\mathbf{h}_t + W_{\text{in}}\mathbf{x}_t + \mathbf{c}_t + \mathbf{b}_h)\,. \tag{3}$$

Detailed model parameters and training procedures are described in the Appendix B.

4 Result

4.1 Cerebellar adaptive feedback enables cortical zero-shot compositional generalization in cognitive and motor tasks

We compare cortex-only (Ctx-only) and cortico-cerebellar (Ctx-cb) models on compositional learning and zero-shot generalization. Both models achieve near-perfect performance on trained combinations, but the Ctx-cb model exhibits substantially better generalization to held-out compositions (Fig.3A,B). Cognitive accuracy improves from 49.14% of RNN to 98.81%, while motor sequence NMSE decreases from 0.2428 to 9.99×10^{-4}. Consistently, Ctx-cb model exhibit a smaller generalization gap between trained and held-out conditions (Fig.3C), maintain stable performance as sequence length increases (Fig.3D), and preserve context-dependent representations (Fig.3E). These results demonstrate that cortico-cerebellar coordination enhances zero-shot compositional generalization beyond task-specific optimization across novel cognitive rules and motor sequences.

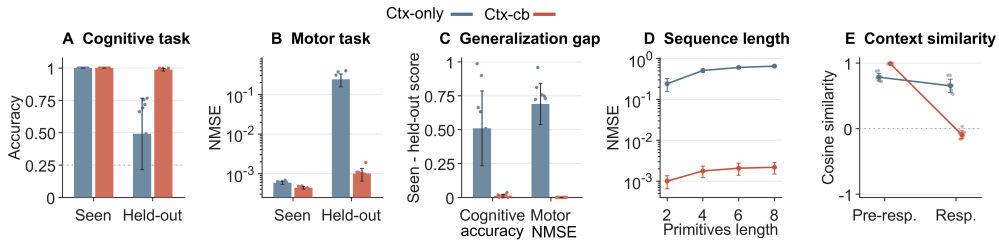


Figure 3: Compositional generalization across cognitive and motor domains. (A,B) Performance on trained and held-out compositions for cognitive and motor tasks. (C) Generalization gaps between seen and held-out compositions. (D) Generalization across increasingly long primitive sequences. (E) Similarity of task-context representations across pre- and post-response periods.

We further perform analysis to reveal the distinct contributions of cerebellar feedback components. In the cognitive task, removing content feedback ($f = 0$) reduces held-out accuracy from 98.80% to 23.65%, whereas gate removal ($q = 0$) causes a smaller reduction to 91.04% (Fig. 4A). In contrast, motor performance requires both components, with NMSE increasing from 0.000996 to 0.497 after gate removal and 0.816 after content removal. These results indicate that cognitive generalization relies primarily on task-specific feedback content, whereas motor precision requires coordinated content and adaptive regulation.

Mismatched contextual feedback nearly abolishes performance, reducing cognitive accuracy to 0.58% and increasing motor NMSE to 1.094 (Fig. 4B). Rotating or reversing the feedback correction direction while preserving its norm similarly impairs performance, reducing accuracy to 24.11% and 1.68%, and increasing NMSE to 1.356 and 5.948, respectively. These results indicate that effective feedback requires appropriate contextual content and state-dependent alignment.

Factorial perturbations also reveal non-additive interactions between content feedback and gating (Fig. 4C), while graded attenuation shows distinct sensitivity profiles across domains (Fig. 4D,E). Cognitive performance is primarily sensitive to loss of feedback content, whereas motor precision progressively degrades with weakening of either component, indicating complementary contributions of content transformation and adaptive state regulation.

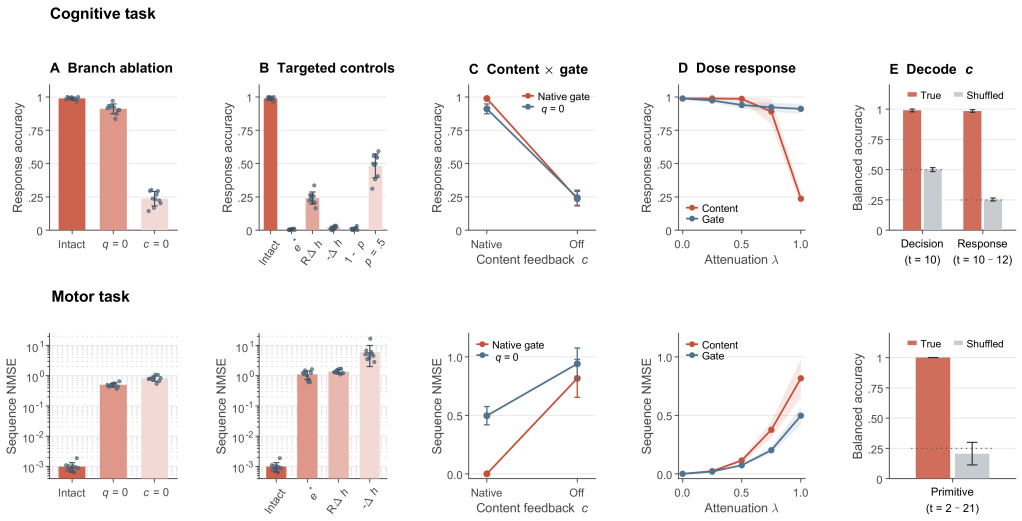


Figure 4: Distinct contributions of cerebellar feedback components. (A–B) Component ablations and targeted perturbations in cognitive and motor tasks. (C–D) Content–gate interactions and dose-response analyses. (E) Decoding of feedback content with intact and shuffled controls.

We further assess the robustness and stability of the coupled cortico-cerebellar model through perturbation and hyperparameter analyses. Across cognitive and motor tasks, the coupled model remains robust to input, state, temporal, and contextual perturbations and consistently outperforms the cortex-only model. Its performance is also stable across broad ranges of architectural and optimization hyperparameters, indicating that zero-shot generalization is not confined to a specific parameter setting. Detailed results are provided in Appendices C and D.

4.2 Representation reuse and dynamical stabilization enable compositional generalization

To determine whether compositional failure arises from impaired representations or ineffective reuse, we evaluate neural decodability with behavioral performance. In the cognitive task, intermediate logical decisions are highly decodable across two models ($\geq$ 90%; Fig. 5A, light bars), yet held-out behavioral accuracy declines sharply in Ctx-only model (49.26%) compared with Ctx-cb model (98.72%; dark bars). Even with correctly decoded decisions, Ctx-only model achieves only 52.08% response accuracy, respectively, whereas Ctx-cb model reaches 99.80% (Fig. 5B). These results reveal a dissociation between latent decision representations and their context-dependent utilization in cortex-only models, whereas cortico-cerebellar coordination enables effective reuse of preserved computations across novel compositions. This representation–utilization gap remains consistent across individual task splits (Fig. 5C).

In the motor task, primitive identities remains highly decodable across three models ($\geq$ 96.78%; Fig. 5D, light bars), whereas Ctx-only models fail to predict velocity kinematics during novel primitive transitions, resulting in degraded trajectory execution. Conditional on successful velocity prediction, behavioral performance recovers across models (Fig. 5E), with kinematics prediction success tightly covarying with behavioral performance across task splits (Fig. 5F). These findings indicate that cortex-only models preserve discrete primitive representations but fail in their dynamical deployment, whereas cortico-cerebellar coordination enables reliable transformation of component representations into continuous motor execution during compositional recombination.

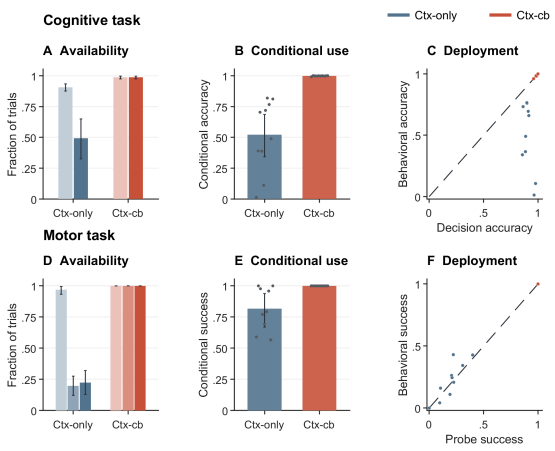


Figure 5: Linking component availability and behavioral deployment during compositional generalization. Linear probes are trained on seen compositions and evaluated on held-out compositions to assess task-relevant variable accessibility and behavioral deployment in cognitive (A) and motor (B) tasks.

We next examine how task variables are organized within cortical population activity. In the cognitive task, factorial variance decomposition reveals predominantly main-effect representations across two models (Fig. 6A–C). Ctx-only models distribute population variance across sensory, logical, and motor-rule factors, whereas the Ctx-cb model exhibits a more structured representation with stronger emphasis on the logical factor. This organization suggests that cortico-cerebellar coordination preferentially emphasizes task-relevant intermediate variables, resulting in a more structured representational organization that may facilitate reuse across novel rule combinations. Consistently, in the motor task, the Ctx-cb model exhibits near-perfect cross-context invariance of representations (cosine similarity: 0.995 vs. 0.761; Fig. 6D–F). Together, these results reveal a more structured and context-stable representational organization associated with the compositional generalization advantage of cortico-cerebellar coordination.

To characterize the kinematic basis of motor compositional generalization, we examine trajectory generation and the corresponding transition dynamics across four held-out primitive transitions (Fig. 7). We first compare the generated and target trajectories in integrated position space, with the primitive switch occurring at $t = 12$ (Fig. 7A–D). Across all four held-out transitions, the Ctx-cb model closely follows the target trajectories before and after the composition boundary, preserving the characteristic geometry of each movement primitive during recombination. In contrast, the Ctx-only model exhibits transition-dependent deviations from the target trajectories, with errors becoming particularly pronounced after switching to the successor primitive.

We next quantify these differences using time-resolved velocity errors to determine how trajectory deviations evolve around the composition boundary (Fig. 7E–H). The Ctx-only model shows substantial velocity-error accumulation around and after the primitive switch, with the magnitude and temporal profile of the error varying across held-out transitions. These deviations indicate that execution of the successor primitive remains sensitive to the preceding movement context. By comparison, the Ctx-cb model maintains consistently low velocity errors across the transition boundary and subsequent execution period, indicating that the learned successor dynamics can be reliably reinstated when the same primitive is embedded in a novel sequential context. Together, these results show that successful motor compositional generalization requires more than accurate representation or execution of individual movement primitives in isolation. Reliable recombination additionally depends on preserving the appropriate trajectory dynamics when transitioning between primitives, such that the successor movement can be initiated without inappropriate dynamical carry-over from its predecessor. The improved trajectory and velocity fidelity of the Ctx-cb

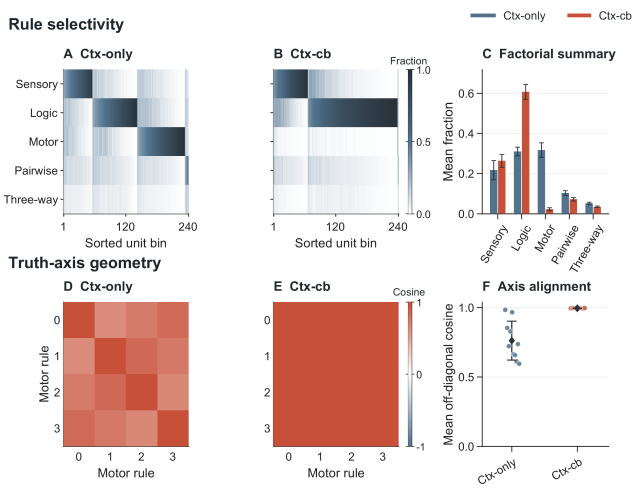


Figure 6: Organization of task-variable representations across two models. (A–C) Unit selectivity and factorial variance decomposition in the cognitive task. Heatmaps show factor-specific variance fractions across pooled units, and the summary quantifies variance contributions from sensory, logic, motor, pairwise, and three-way factors. (D–F) Cross-context alignment of truth-variable representations in the motor task. Matrices show signed cosine similarity between truth-coding population axes across motor rules, with summary statistics of off-diagonal alignment across models.

model therefore motivates a more direct examination of how cortico-cerebellar coordination regulates cortical states at composition boundaries.

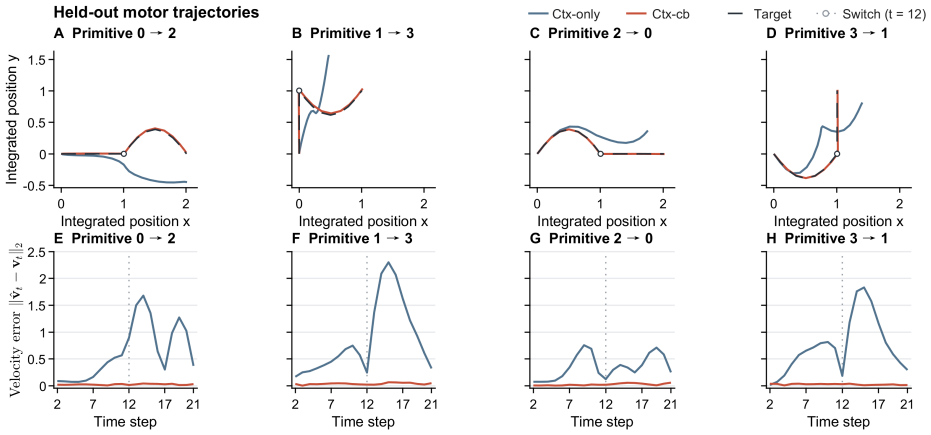


Figure 7: Dynamical fidelity during motor compositional recombination.(A–D) Reconstructed trajectories for four held-out primitive transitions, including Primitive 0→2, 1→3, 2→0, and 3→1. Solid curves show model-generated trajectories, black dashed curves indicate target trajectories, and colors denote cortex-only (Ctx-only) and cortico-cerebellar (Ctx-cb) models. Open circles mark the transition boundaries where primitive composition occurs ($t = 12$). (E–H) Time-resolved velocity errors corresponding to the transitions in (A–D). Vertical dashed lines indicate primitive boundaries, highlighting error accumulation and recovery during compositional transitions.

We further examine how predecessor-dependent states influence successor dynamics during motor transitions. For trials sharing the same successor primitive but different predecessors,

state variance accumulates during the preceding primitive and propagates into the transition boundary in the cortex-only model (Fig. 8A). In contrast, the Ctx-cb model shows a sharp reduction in boundary variance, yielding a boundary contraction ratio close to zero, whereas the cortex-only model retains substantial predecessor-dependent variability (Fig. 8B). Consistently, PCA projections reveal that successor trajectories remain separated by predecessor context in the cortex-only model, while the Ctx-cb model converges onto successor-specific low-dimensional manifolds after transitions (Fig. 8C,D). We next assess transition robustness using equal-amplitude perturbations applied at the primitive boundary. The Ctx-cb model exhibits faster state recovery and smaller terminal deviations compared with the Ctx-only model (Fig. 8E). Across task splits, larger successor-entry state variance associates with increased successor prediction error in the Ctx-only model, whereas the Ctx-cb model maintains low entry variance and NMSE (Fig. 8F). Similarly, perturbations causing larger terminal state deviations produce higher recovery costs, with the Ctx-cb model remaining less sensitive to boundary perturbations (Fig. 8G). Together, these results show that cortico-cerebellar coordination selectively contracts transition-state variability, reducing inappropriate predecessor-dependent carry-over while preserving stable successor-specific dynamics for primitive recombination.

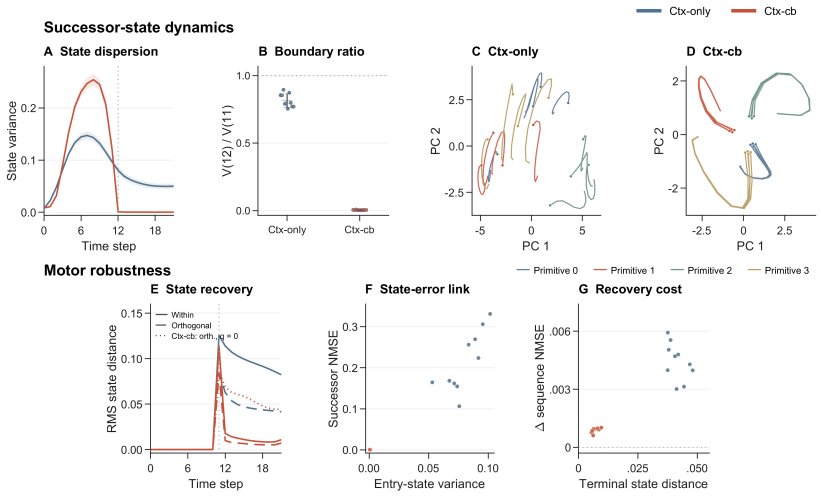


Figure 8: Successor-state dynamics and robustness at motor primitive boundaries.(A) Predecessor-dependent state variance around the primitive transition. (B) Boundary contraction measured by the state-variance ratio. (C,D) PCA trajectories of successor states in Ctx-only and Ctx-cb models. (E) State recovery following boundary perturbations. (F) Successor NMSE versus entry-state variance. (G) Perturbation-induced sequence NMSE versus terminal state deviation.

5 Discussion

Collectively, our results suggest that adaptive cortico-cerebellar coordination supports compositional generalization by stabilizing task-relevant cortical dynamics while suppressing inappropriate state carry-over. Across cognitive and motor tasks, cerebellar feedback promoted reliable reuse of learned components without requiring fully modular or disentangled cortical representations. In particular, the separation between feedback content and gating suggests a general computational principle in which structured corrective signals reshape cortical states, while contextual control determines when and how strongly these corrections are applied. The present model abstracts away several biological details of the cerebellar circuit. Future work could incorporate mossy fiber–granule cell expansion, structured granule-cell coding, and Purkinje-cell excitation–inhibition dynamics to examine how high-dimensional representations, inhibitory regulation, and cerebellar output jointly support flexible learning.

AI use statement

In this work, we used generative AI tools for language polishing and simulation-code debugging. The authors reviewed the text, code, results and cited sources and take responsibility for the final content. The experimental design, simulations and scientific claims are the authors' own work.

Ethics statement

This work does not involve human participants, animal experiments, personally identifiable information, or the collection and release of new datasets. The proposed computational framework is developed and evaluated using synthetic benchmark tasks and does not introduce direct risks related to privacy, safety, discrimination, or harmful applications. We have followed principles of research integrity and transparency throughout the study.

Reproducibility statement

We have made efforts to ensure the reproducibility of our results. Detailed descriptions of benchmark task generation, learning rules, training procedures, and evaluation protocols are provided in the Appendix.

References

Chen Avraham and Firas Mawase. Feedback scaffolds predictive control in de novo motor learning. bioRxiv, pp. 2025–04, 2025.

Yeyao Bao, Liao Yu, Liangfu Lu, Zhuoqin Yang, and Yunliang Zang. Cerebellar microcircuits enable robust evidence-based decisions through cortico–cerebellar coupling. Proceedings of the National Academy of Sciences, 123(35):e2616911123, 2026.

Ellen Boven, Joseph Pemberton, Paul Chadderton, Richard Apps, and Rui Ponte Costa. Cerebro-cerebellar networks facilitate learning through feedback decoupling. Nature communications, 14(1):51, 2023.

Michael W Cole, Joset A Etzel, Jeffrey M Zacks, Walter Schneider, and Todd S Braver. Rapid transfer of abstract rules to novel contexts in human lateral prefrontal cortex. Frontiers in human neuroscience, 5:142, 2011.

Katherine M Collins, Ilia Sucholutsky, Umang Bhatt, Kartik Chandra, Lionel Wong, Mina Lee, Cedegao E Zhang, Tan Zhi-Xuan, Mark Ho, Vikash Mansinghka, et al. Building machines that learn and think with people. Nature human behaviour, 8(10):1851–1863, 2024.

Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In 2017 IEEE international conference on robotics and automation (ICRA), pp. 2169–2176. IEEE, 2017.

Laura N Driscoll, Krishna Shenoy, and David Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. Nature neuroscience, 27(7):1349–1363, 2024.

Ziye Fang, Bo Wan, Shangqi Guo, and Jian K Liu. A scalable complexity-regularized modular network for continual learning. Neurocomputing, pp. 135079, 2026.

Zhenyu Gao, Courtney Davis, Alyse M Thomas, Michael N Economo, Amada M Abrego, Karel Svoboda, Chris I De Zeeuw, and Nuo Li. A cortico-cerebellar loop for motor planning. Nature, 563(7729):113–116, 2018.

Martha G Garcia-Garcia, Michał J Wójcik, Srijan Thota, Luke Drake, Amma Otchere, Oluwatobi Akinwale, Lizmaylin Ramos, Rui Ponte Costa, and Mark J Wagner. Granule cells reorient cortical trajectories to separate contexts. Nature, pp. 1–10, 2026.

Sharon Israely, Hugo Ninou, Ori Rajchert, Lee Elmaleh, Ran Harel, Firas Mawase, Jonathan Kadmon, and Yifat Prut. Cerebellar output shapes cortical preparatory activity during motor adaptation. Nature Communications, 16(1):2574, 2025.

Takuya Ito, Tim Klinger, Doug Schultz, John Murray, Michael Cole, and Mattia Rigotti. Compositional generalization through abstract representations in human and artificial neural networks. Advances in neural information processing systems, 35:32225–32239, 2022.

Roberta M Kelly and Peter L Strick. Cerebellar loops with motor cortex and prefrontal cortex of a nonhuman primate. The Journal of neuroscience, 23(23):8432–8444, 2003.

Jorge A Mendez and Eric Eaton. How to reuse and compose knowledge for a lifetime of tasks: A survey on continual learning and functional composition. arXiv preprint arXiv:2207.07730, 2022.

Ravi D Mill and Michael W Cole. Dynamically shifting from compositional to conjunctive brain representations supports cognitive task learning. Nature communications, 16(1): 10084, 2025.

Samuel P Muscinelli, Mark J Wagner, and Ashok Litwin-Kumar. Optimal routing to cerebellum-like structures. Nature neuroscience, 26(9):1630–1641, 2023.

Joseph Pemberton, Ellen Boven, Richard Apps, and Rui Ponte Costa. Cortico-cerebellar networks as decoupling neural interfaces. Advances in neural information processing systems, 34:7745–7759, 2021.

Joseph Pemberton, Paul Chadderton, and Rui Ponte Costa. Cerebellar-driven cortical dynamics can enable task acquisition, switching and consolidation. Nature Communications, 15(1):10913, 2024.

Carlo Reverberi, Kai Görgen, and John-Dylan Haynes. Compositionality of rule representations in human prefrontal cortex. Cerebral cortex, 22(6):1237–1246, 2012.

Reidar Riveland, Alexandre Pouget, and Laura Driscoll. The compositionality continuum as a principle for studying the neural basis of intelligence. Nature neuroscience, pp. 1–14, 2026.

Ladan Shahshahani, Maedbh King, Caroline Nettekoven, Richard B Ivry, and Jörn Diedrichsen. Selective recruitment of the cerebellum evidenced by task-dependent gating of inputs. Elife, 13:RP96386, 2024.

Sina Tafazoli, Flora M Bouchacourt, Adel Ardalan, Nikola T Markov, Motoaki Uchimura, Marcelo G Mattar, Nathaniel D Daw, and Timothy J Buschman. Building compositional tasks with shared neural subspaces. Nature, 650(8100):164–172, 2026.

Joshua B Tan, Isabella F Orlando, Jungwoo Kim, Christopher J Cueva, Jayson Jeganathan, Giulia Baracchini, Rebekah Wong, Eli J Müller, Claire O'Callaghan, and James M Shine. Compositional recombination relies on a distributed cortico-cerebellar network. bioRxiv, pp. 2025–10, 2025.

Yuanhong Tang, Lingling An, Quan Wang, and Jian K Liu. Regulating synchronous oscillations of cerebellar granule cells by different types of inhibition. PLoS computational biology, 17(6):e1009163, 2021a.

Yuanhong Tang, Lingling An, Ye Yuan, Qingqi Pei, Quan Wang, and Jian K Liu. Modulation of the dynamics of cerebellar purkinje cells through the interaction of excitatory and inhibitory feedforward pathways. PLoS computational biology, 17(2):e1008670, 2021b.

Mark J Wagner, Tony Hyun Kim, Jonathan Kadmon, Nghia D Nguyen, Surya Ganguli, Mark J Schnitzer, and Liqun Luo. Shared cortex-cerebellum dynamics in the execution and learning of a motor task. Cell, 177(3):669–682, 2019.

Marjorie Xie, Samuel P Muscinelli, Kameron Decker Harris, and Ashok Litwin-Kumar. Task-dependent optimal representations for cerebellar learning. Elife, 12:e82914, 2023.

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(2):297–306, 2019.

Liao Yu, Zhuoqin Yang, Yeyao Bao, and Yunliang Zang. Balancing inhibition and sparsity for stable, accurate cerebellar learning. bioRxiv, pp. 2026–04, 2026.

Yaoduo Zhang, Jian Zhang, and Yunliang Zang. Cerebellum-inspired kernel for efficient out-of-distribution detection. bioRxiv, pp. 2026–03, 2026.

A Task structure and input-output protocol

A.1 Compositional cognitive task

In cognitive task, each trial requires sequential application of a sensory rule, a logical rule, and a motor rule. The sensory rule selects relevant features from two sequentially presented stimuli. The logical rule maps selected-feature values into a Boolean decision, and the motor rule projects this decision onto one of four response channels. Each rule component admits four possible values, yielding $4 \times 4 \times 4 = 64$ distinct combinations. Each trial spans $T = 13$ discrete time steps indexed by $t = 0, \ldots, 12$. Trials start with a two-step preparation period. The first stimulus appears at $t = 2, 3$, followed by a one-step interval at $t = 4$. The second stimulus is presented at $t = 5, 6$, with a subsequent delay spanning $t = 7, 8, 9$. The response phase starts at $t_{\text{go}} = 10$ and persists until trial termination.

At every time step t, the model receives a twenty-dimensional input vector

$$\mathbf{x}_t = \left[\mathbf{r}^{\text{sens}}, \mathbf{r}^{\text{logic}}, \mathbf{r}^{\text{motor}}, \mathbf{s}_t, \mathbf{q}_t\right] \in \mathbb{R}^{20}.$$

The first twelve input dimensions encode the three rule components via separate four-dimensional one-hot vectors

$$\mathbf{r}^{\text{sens}} = \mathbf{e}_s^{(4)}, \qquad \mathbf{r}^{\text{logic}} = \mathbf{e}_\ell^{(4)}, \qquad \mathbf{r}^{\text{motor}} = \mathbf{e}_m^{(4)},$$

where $s, \ell, m \in \{0, 1, 2, 3\}$, and $\mathbf{e}_i^{(d)}$ denotes the i-th zero-indexed one-hot vector in d dimensions. These rule cues persist throughout the full trial.

The subsequent four input dimensions encode raw stimulus features. Each stimulus is represented by a binary vector $\mathbf{z}^{(1)}, \mathbf{z}^{(2)} \in \{0, 1\}^4$. Binary inputs are converted to signed feature values , where 0 maps to -1 and 1 maps to $+1$, giving

$$\mathbf{s}_t = \begin{cases} 2\mathbf{z}^{(1)} - \mathbf{1}_4, & 2 \le t < 4, \\ 2\mathbf{z}^{(2)} - \mathbf{1}_4, & 5 \le t < 7, \\ \mathbf{0}_4, & \text{otherwise.} \end{cases}$$

The final four input dimensions encode the current trial phase. Separate one-hot phase cues mark preparation, stimulus presentation, delay, and response epochs:

$$\mathbf{q}_t = \begin{cases} \mathbf{e}_0^{(4)}, & 0 \le t < 2, \\ \mathbf{e}_1^{(4)}, & 2 \le t < 4 \text{ or } 5 \le t < 7, \\ \mathbf{e}_2^{(4)}, & 4 \le t < 5 \text{ or } 7 \le t < 10, \\ \mathbf{e}_3^{(4)}, & 10 \le t < 13. \end{cases}$$

The sensory rule selects the same feature dimension from both stimuli, giving $a = z_s^{(1)}, \quad b = z_s^{(2)}$. The logical rule maps these two selected values to a Boolean decision:

$$d = F_\ell(a, b),$$

with

$$F_\ell(a,b) = \begin{cases} a \wedge b, & \ell = 0 \quad \text{(BOTH)}, \\ a \vee b, & \ell = 1 \quad \text{(EITHER)}, \\ \neg(a \wedge b), & \ell = 2 \quad \text{(NOT BOTH)}, \\ \neg(a \vee b), & \ell = 3 \quad \text{(NEITHER)}. \end{cases}$$

True and false outcomes are sampled with equal probability within each task combination.

The motor rule defines the response channel associated with a true decision. A false decision triggers the paired alternative response:

$$c = \begin{cases} m, & d = 1, \\ \pi(m), & d = 0, \end{cases} \qquad \pi(0) = 1, \quad \pi(1) = 0, \quad \pi(2) = 3, \quad \pi(3) = 2.$$

The target response is the four-dimensional one-hot vector $\mathbf{e}_c^{(4)}$. Behavioral predictions are computed by averaging the four output logits across the response period and selecting the largest component:

$$\overline{\mathbf{o}} = \frac{1}{T - t_{\text{go}}} \sum_{t=t_{\text{go}}}^{T-1} \mathbf{o}_t, \qquad \widehat{c} = \underset{k \in \{0,1,2,3\}}{\arg\max}\ \overline{o}_k.$$

Behavioral readout is scored during the response period. Earlier time steps receive no constraint from the response-classification objective.

This task requires the network to extract task-relevant sensory information, maintain information across temporally separated events, execute context-dependent logical operations, and map the resulting decision onto the appropriate response. Since rule cues and stimulus features are provided as separate inputs, successful task performance requires their coordinated combination within the network.

A.2 Compositional motor task

We construct motor task using four synthetic kinematic primitives: rightward translation, upward translation, upper arc, and lower arc. Each primitive is parameterized by a 2-D velocity profile defined over a normalized movement phase variable $u \in [0, 1]$. Horizontal and vertical transport components are modulated by the envelope function $E(u) = 6u(1-u)$. The four primitive velocity profiles are

$$\mathbf{v}_p(u) = \begin{cases} [E(u),\, 0]\,, & p = 0 \quad \text{(Right)}, \\ [0,\, E(u)]\,, & p = 1 \quad \text{(Up)}, \\ [E(u),\, 1.2\sin(2\pi u)]\,, & p = 2 \quad \text{(Upper arc)}, \\ [E(u),\, -1.2\sin(2\pi u)]\,, & p = 3 \quad \text{(Lower arc)}. \end{cases}$$

The corresponding position trajectories are obtained by integration, $\mathbf{r}_p(u) = \int_0^u \mathbf{v}_p(\xi)\, d\xi$. Thus, the rightward and two-arc primitives yield identical net horizontal displacement despite distinct paths, whereas the upward primitive yields purely vertical displacement. Each motor trial specifies an ordered primitive pair (i, j). The network executes primitive i followed by primitive j. Every primitive spans $L = 10$ discrete time steps. Each trial contains two initial preparation steps, giving total trial length $T = 2 + 2L = 22$. The first primitive runs for $2 \le t < 12$, and the second runs for $12 \le t < 22$.

At each time step, the model receives a 14-dimensional input vector

$$\mathbf{x}_t = \left[\mathbf{e}_i^{(4)}, \mathbf{e}_j^{(4)}, \mathbf{q}_t, u_t, A_t, \beta_t, g_t\right] \in \mathbb{R}^{14}.$$

The first eight dimensions $\left[\mathbf{e}_i^{(4)}, \mathbf{e}_j^{(4)}\right]$ encode the two planned primitive identities using separate 4-dimensional one-hot vectors. Both identity cues persist throughout the trial. The subsequent two dimensions correspond to $\mathbf{q}_t$, which signals the currently active position within the primitive sequence:

$$\mathbf{q}_t = \begin{cases} [0,0], & 0 \le t < 2, \\ [1,0], & 2 \le t < 12, \\ [0,1], & 12 \le t < 22. \end{cases}$$

The scalar phase input u_t encodes progression inside the currently active primitive. Velocity profiles are evaluated at the midpoint of each discrete phase bin:

$$u_t = \begin{cases} 0, & 0 \le t < 2, \\ \dfrac{t - 2 + \frac{1}{2}}{L}, & 2 \le t < 12, \\ \dfrac{t - 12 + \frac{1}{2}}{L}, & 12 \le t < 22. \end{cases}$$

A movement amplitude $A \sim \mathcal{U}(0.8, 1.2)$ is sampled independently per trial and applied to both primitives. The remaining scalar inputs encode amplitude, primitive onset, and movement execution via

$$A_t = A\,\mathbb{I}[t \ge 2], \qquad \beta_t = \mathbb{I}[t \in \{2, 12\}], \qquad g_t = \mathbb{I}[t \ge 2],$$

where $\mathbb{I}[\cdot]$ denotes the indicator function.

The target output is a two-dimensional velocity vector

$$\mathbf{y}_t^* = \begin{cases} A\,\mathbf{v}_i(u_t), & 2 \le t < 12, \\ A\,\mathbf{v}_j(u_t), & 12 \le t < 22. \end{cases}$$

Motor-output loss is evaluated only during movement execution. The two preparation steps are excluded from the trajectory loss.

For visualization, predicted and target velocities are integrated with discrete-phase increment $1/L$. Setting the initial position $\mathbf{R}_2 = \mathbf{0}_2$, trajectories evolve as

$$\mathbf{R}_{t+1} = \mathbf{R}_t + \frac{1}{L}\mathbf{y}_t, \qquad 2 \le t < 22.$$

Position state carries across primitive boundaries, so the second movement starts at the endpoint of the first. This protocol requires the network to select the intended planned primitive, produce its phase-dependent velocity profile, and switch to the second primitive at the prescribed boundary.

To test whether learned motor primitives can recombine into unseen sequences, we partition the full set of ordered primitive pairs into disjoint training, validation, and test splits. Given four primitive identities and two sequence positions, the set of all ordered combinations reads

$$\mathcal{P} = \{(i, j) : i, j \in \{0, 1, 2, 3\}\}, \qquad |\mathcal{P}| = 16$$

For the partition visualized in Figure , we define the relative primitive index $\delta(i, j) = (j - i) \bmod 4$, and the three dataset splits are specified as

$$(i,j) \in \begin{cases} \mathcal{P}_{\text{train}}, & \delta(i,j) \in \{0,1\}, \\ \mathcal{P}_{\text{val}}, & \delta(i,j) = 3, \\ \mathcal{P}_{\text{test}}, & \delta(i,j) = 2. \end{cases}$$

This yields $|\mathcal{P}_{\text{train}}| = 8, \quad |\mathcal{P}_{\text{val}}| = 4, \quad |\mathcal{P}_{\text{test}}| = 4$. The test set concretely corresponds to $\mathcal{P}_{\text{test}} = \{(0,2),(1,3),(2,0),(3,1)\}$. This partition operates at the granularity of full ordered pairs. Every primitive appears in both sequence positions during training, whereas validation and test combinations are entirely held-out. Each row and column of the pair matrix contains two training, one validation, and one test combination. Accordingly, held-out trials require the network to compose familiar primitives within novel ordered pairings. All splits share identical input representation, primitive definitions, movement durations, and amplitude distribution. Test-set evaluation is performed without parameter updates. For each trial, trajectory error is quantified via normalized mean-squared velocity error

$$\text{NMSE} = \frac{\sum_{t=2}^{T-1} \|\widehat{\mathbf{y}}_t - \mathbf{y}_t^*\|_2^2}{\sum_{t=2}^{T-1} \|\mathbf{y}_t^*\|_2^2}.$$

This setup evaluates compositional generalization by executing previously unseen primitive pairings. Since sequence position, movement phase, and primitive boundaries are provided as explicit inputs, the core computational challenge lies in context-dependent selection and reuse of learned movement components.

B Experimental settings and learning algorithms

B.1 Experimental settings

We evaluated compositional learning in a cognitive rule task and a motor primitive-sequencing task. The cognitive task comprised $4 \times 4 \times 4 = 64$ combinations of sensory, logical, and motor-response rules. For each split, 32 combinations were used for training, 16 disjoint combinations for diagnostic validation, and 16 disjoint combinations for final zero-shot testing. The motor task comprised $4 \times 4 = 16$ ordered pairs of motor primitives, divided into 8 training, 4 validation, and 4 test transitions. Every constituent cognitive rule and motor primitive was represented in the training data; only their complete combinations or ordered transitions were withheld.

We used ten independently generated task splits. Splits 1–5 were structured partitions defined by independently permuting the factor labels and grouping combinations according to their summed indices modulo four. Splits 6–10 were random partitions constrained so that the cognitive training set contained all levels of each individual factor and all 16 pairwise factor combinations. Ten independently initialized models were trained for each split, yielding 100 independent replicas per model and task.

The principal comparison included a cortex-only model and the cortico-cerebellar coupled model. Model widths were selected to approximately match the total number of trainable parameters rather than the number of recurrent units. For the cognitive task, the cortex-only and cortico-cerebellar coupled models contained 79 and 48 cortical units, corresponding to 8,460 and 8,424 trainable parameters, respectively. For the motor task, the respective cortical widths were 98 and 48 units, corresponding to 11,272 and 11,319 parameters. The cortico-cerebellar coupled model additionally used a 64-dimensional cerebellar content layer and a 16-unit gate network. Cognitive and motor models used the same cortico-cerebellar coupled update equations but were trained separately; the principal experiments therefore did not share parameter values across task domains.

Simulations were implemented in Python 3.12 using PyTorch and executed on CPUs. Cognitive trials contained 13 discrete time steps: two preparation steps, two presentations of the

first stimulus, one intervening step, two presentations of the second stimulus, three maintenance steps, and three response steps. The cognitive input contained 12 rule channels, four stimulus channels, and four task-phase indicators. Only the mean logits over the response period were used to calculate behavioral accuracy.

Motor trials contained two preparation steps followed by two ten-step motor primitives. Each motor input comprised eight plan channels, two current-slot channels, one phase channel, one amplitude channel, one primitive-boundary channel, and one execution indicator. Primitive amplitude was sampled independently from $\mathcal{U}[0.8, 1.2]$. The first 400 optimization updates used single-primitive trials, followed by 1,200 updates using the permitted two-primitive training transitions.

All models were optimized with AdamW using a fixed learning rate of 2×10^{-3}, weight decay 10^{-5}, mini-batches of 32 trials, and a per-replica gradient-norm limit of 1. No learning-rate scheduler, dropout, neural noise, within-trial synaptic update, or online cerebellar plasticity was used. Recurrent matrices were initialized orthogonally with gain 0.9. Other affine weights were sampled uniformly according to their input width. The cerebellar feedback projection was initialized from a zero-mean Gaussian distribution with standard deviation 0.03; the gate-output weights used standard deviation 0.015, and the initial update-gate bias was $b_q = -2$, favoring retention at initialization. The motor addressing matrix was initialized to the identity, constituting an explicit routing prior.

Model parameters were updated only from training combinations. Held-out validation and test combinations never contributed gradients. Checkpoint selection was performed independently for each replica using behavioral loss on the fixed trained-combination evaluation set; held-out validation performance was recorded but was not used for model selection. After checkpoint selection, all weights were frozen and performance was measured on 1,024 independently sampled held-out test trials. Uncertainty for the principal aggregate results was estimated using 5,000 hierarchical bootstrap samples, first resampling the ten task splits and then the ten initialization seeds within each split.

B.2 Learning algorithm

The cortico-cerebellar coupled model was trained end-to-end by backpropagation through time (BPTT). At each time step, the current cortical state and observable task input were passed to a feedforward cerebellar pathway, which jointly computed a state-dependent update gate and task-specific feedback content. The resulting feedback was then returned to the cortical recurrent network before the next state was computed. The cerebellar pathway contained no recurrent state of its own, and all model parameters remained fixed within each trial.

Let $\mathbf{h}_t \in \mathbb{R}^{48}$ denote the cortical state and $\mathbf{x}_t \in \mathbb{R}^I$ the observable task input, with $I = 20$ for the cognitive task and $I = 14$ for the motor task. The adaptive retention–update gate was computed as

$$\begin{aligned} \mathbf{a}_t &= \tanh\left(G_h \mathbf{h}_t + G_x \mathbf{x}_t + \mathbf{b}_g\right), \\ q_t &= \sigma\left(\mathbf{w}_q^\top \mathbf{a}_t + b_q\right), \\ \overline{\mathbf{h}}_t &= (1 - q_t)\mathbf{h}_t, \end{aligned}$$

where $q_t \in [0, 1]$ is the learned update gain and $(1 - q_t)$ is the corresponding retention gain. The gate was scalar but state- and input-dependent. It was not multiplied by a predefined task-type or boundary-specific reset signal.

The gated cortical state was subsequently updated according to

$$\begin{aligned} \mathbf{f}_t &= U \mathbf{m}_t, \\ \mathbf{h}_{t+1} &= (1 - \alpha)\overline{\mathbf{h}}_t + \alpha \tanh\left(W_{\text{rec}} \overline{\mathbf{h}}_t + W_{\text{in}} \mathbf{d}_t + \mathbf{f}_t + \mathbf{b}_h\right), \\ \widehat{\mathbf{y}}_t &= W_{\text{out}} \mathbf{h}_{t+1} + \mathbf{b}_{\text{out}}, \end{aligned}$$

with $\alpha = 0.6$. Here, $\mathbf{m}_t \in \mathbb{R}^{64}$ denotes the task-specific cerebellar content representation, $\mathbf{f}_t$ is the feedback returned to cortex, and $\mathbf{d}_t$ is the task-dependent cortical input defined below.

Cognitive task

Each cognitive task was specified by a sensory rule, a logical rule, and a motor-response rule. The sensory and logical operations produced an intermediate binary decision $v \in \{0, 1\}$. The cognitive cerebellar interface estimated this decision from the gated cortical state,

$$p_t = \sigma\left(\mathbf{w}_v^\top \overline{\mathbf{h}}_t + b_v\right),$$

and bound it to the four-dimensional motor-rule vector $\mathbf{r}_m$:

$$\mathbf{z}_t^{\text{bind}} = \begin{bmatrix} p_t \mathbf{r}_m \\ (1 - p_t)\mathbf{r}_m \end{bmatrix},$$

$$\mathbf{m}_t^{\text{cog}} = \chi_t \tanh\left(W_{\text{bind}} \mathbf{z}_t^{\text{bind}} + \mathbf{b}_{\text{bind}}\right),$$

where χ_t is the observable response-period indicator. The model used its own predicted decision p_t; the ground-truth intermediate decision was never supplied as an input during either training or evaluation. For the cognitive task, the complete observable input was passed to cortex, such that $\mathbf{d}_t = \mathbf{x}_t$.

Behavioral classification was based on the response-period average of the four output logits,

$$\frac{1}{|\mathcal{T}_{\text{resp}}|} \sum_{t \in \mathcal{T}_{\text{resp}}} \widehat{\mathbf{y}}_{i,t}, \qquad \mathbf{p}_i = \text{softmax}(\overline{\mathbf{y}}_i).$$

The cognitive training objective combined response classification with an auxiliary loss on the two selected sensory bits and their intermediate logical value:

$$-\frac{1}{B} \sum_{i=1}^{B} \log p_i^{y_i} + 2 \cdot \text{BCE}\left(\widehat{\mathbf{u}}_i, \mathbf{u}_i^*\right),$$

where

$$\widehat{\mathbf{u}}_i = W_{\text{aux}} \mathbf{h}_{i,t_{\text{resp}}-1} + \mathbf{b}_{\text{aux}}, \qquad \mathbf{u}_i^* = [a_i, b_i, v_i].$$

The auxiliary targets were used only to define the training objective. They were not provided to the recurrent dynamics or behavioral output during testing.

Motor task

For the motor task, each trial specified two planned primitives, $\mathbf{p}_{1,t}, \mathbf{p}_{2,t} \in \mathbb{R}^4$, and a two-dimensional current-slot signal $\mathbf{s}_t$. The cerebellar pathway selected the currently relevant primitive through

$$\boldsymbol{\omega}_t = \text{softmax}\left(3 W_a \mathbf{s}_t\right),$$

$$\mathbf{c}_t = \sum_{j=1}^{2} \omega_{j,t} \mathbf{p}_{j,t}.$$

The selected primitive was combined with the gated cortical state to construct the cerebellar feedback representation,

$$\tanh\left(V \overline{\mathbf{h}}_t + W_c \mathbf{c}_t + \mathbf{b}_c\right) \odot \left[\mathbf{1} + \tanh\left(W_g \widetilde{\mathbf{e}}_t + \mathbf{b}_e\right)\right],$$

where $\widetilde{\mathbf{e}}_t = [\mathbf{0}_8, \mathbf{s}_t]$ retained the current-slot information while excluding the two complete motor plans from the gain pathway. The full input remained available to the adaptive gate and primitive-addressing pathway. To prevent a parallel direct-plan solution through the cortical input projection, the cortical drive was

$$\mathbf{d}_t = [\mathbf{0}_8, \mathbf{x}_{t,8:14}].$$

Thus, the motor plans influenced cortical dynamics through the structured cerebellar content pathway rather than through the direct cortical input projection.

The motor readout $\widehat{\mathbf{v}}_t \in \mathbb{R}^2$ represented the predicted two-dimensional velocity. The training objective was

$$\mathcal{L}_{\text{mot}} = \frac{1}{B|\mathcal{T}_{\text{exec}}|} \sum_{i=1}^{B} \sum_{t \in \mathcal{T}_{\text{exec}}} \|\widehat{\mathbf{v}}_{i,t} - \mathbf{v}_{i,t}\|_2^2$$

$$+ 0.05 \cdot \frac{1}{B(|\mathcal{T}_{\text{exec}}| - 1)} \sum_{i=1}^{B} \sum_{t} \|\Delta\widehat{\mathbf{v}}_{i,t} - \Delta\mathbf{v}_{i,t}\|_2^2 + 0.02 \cdot \mathcal{H}_{\text{addr}},$$

Algorithm 1 Training cortico-cerebellar coupled model on cognitive and motor tasks

Require: Task $\tau \in \{\text{cognitive}, \text{motor}\}$; training combinations $\mathcal{C}^{\tau}_{\text{train}}$; model parameters Θ; batch size $B = 32$

Initialize cortical, cerebellar, gating, and readout parameters
Initialize AdamW optimizer with learning rate 2×10^{-3} and weight decay 10^{-5}
Set $K_{\text{cognitive}} = 3000$ and $K_{\text{motor}} = 1600$
for $k = 1, \ldots, K_{\tau}$ **do**
  if $\tau = \text{motor}$ and $k \leq 400$ **then**
    Sample a batch of single motor primitives
  else
    Sample a batch exclusively from $\mathcal{C}^{\tau}_{\text{train}}$
  end if
  Construct input sequence $\mathbf{x}_{1:T}$ and target sequence or response label
  Initialize cortical state $\mathbf{h}_0 = \mathbf{0}$
  for $t = 0, \ldots, T-1$ **do**
    Compute adaptive update gain $q_t = \sigma\big(\mathbf{w}_q^{\top} \tanh(G_h \mathbf{h}_t + G_x \mathbf{x}_t + \mathbf{b}_g) + b_q\big)$
    Compute gated state $\overline{\mathbf{h}}_t = (1 - q_t)\mathbf{h}_t$
    if $\tau = \text{cognitive}$ **then**
      Predict intermediate decision p_t and bind it to the motor rule
      Compute cognitive cerebellar representation $\mathbf{m}_t^{\text{cog}}$
    else
      Address the current plan slot and retrieve primitive content $\mathbf{c}_t$
      Compute motor cerebellar representation $\mathbf{m}_t^{\text{mot}}$
    end if
    Compute feedback $\mathbf{f}_t = U\mathbf{m}_t$
    Update cortical state $\mathbf{h}_{t+1}$ and record output $\widehat{\mathbf{y}}_t$
  end for
  if $\tau = \text{cognitive}$ **then**
    Compute $\mathcal{L}_{\text{cog}}$ from response-period logits and auxiliary decisions
  else
    Compute $\mathcal{L}_{\text{mot}}$ from velocity, velocity-difference, and addressing losses
  end if
  Backpropagate $\mathcal{L}_{\tau}$ through the complete sequence
  Clip each replica's gradient norm to 1
  Update Θ using AdamW
  if k is a multiple of 100 **then**
    Evaluate behavioral loss on a fixed set of 256 independently sampled trained-combination trials
    Retain the checkpoint if its trained-combination behavioral loss is lower
    Evaluate the held-out validation combinations for diagnostic reporting only
  end if
end for
Freeze all parameters and evaluate the selected checkpoint on 1024 trials sampled from the held-out test combinations

where $\mathcal{H}_{\text{addr}}$ is the entropy of the learned slot-addressing distributions. Because this term is minimized, it encourages sharper slot-to-plan assignments. The two preparation steps were excluded from the behavioral error.

C Robustness analysis

C.1 Cognitive robustness

The coupled cortico-cerebellar model (orange) showed substantially greater robustness than the cortex-only model (blue) across all perturbation conditions (Figure. 9). Under matched checkpoints and held-out task samples, the intact coupled model achieved 98.80% accuracy, compared with 48.97% for the cortex-only model.

Under sensory-input corruption, performance of the coupled model declined gradually with increasing Gaussian noise. Held-out accuracy remained high at 98.71%, 97.73%, and 92.76% for noise standard deviations (SDs) of 0.1, 0.25, and 0.5, respectively, and decreased to 78.34% at SD = 1.0. Because rule and timing cues were preserved, these results indicate that the coupled model tolerates moderate sensory noise while degrading only under stronger corruption.

Perturbations applied directly to the cortical state during the delay period produced a similar dose-dependent effect. Accuracy remained at 98.54% and 97.52% for injected RMS values of 0.15 and 0.3, respectively, and declined to 90.91% at RMS = 0.6. Global parameter perturbations also showed progressive degradation: relative weight noise of 0.03 and 0.1 yielded accuracies of 98.64% and 96.80%, respectively, whereas noise at 0.3 reduced accuracy to 79.70%. Because parameter perturbation was applied to all learned matrices, including the readout, this manipulation reflects overall model robustness rather than the contribution of any single cerebellar feedback branch. Each corruption level was averaged over ten noise realizations per trained replicate.

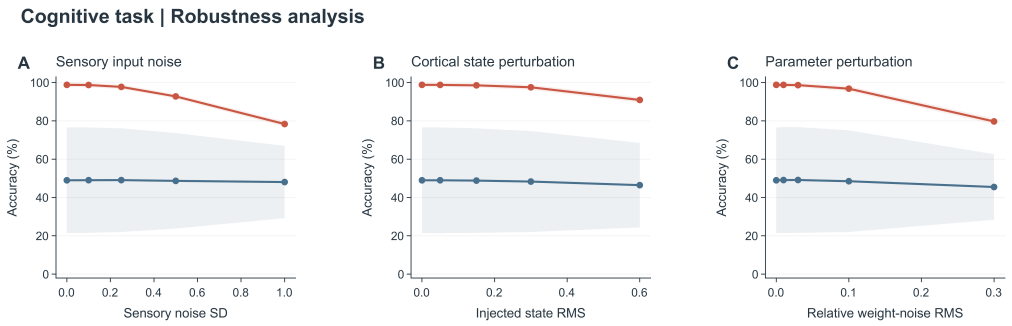


Figure 9: Robustness analysis in the cognitive task. Held-out accuracy of the cortex-only model (blue) and the coupled cortico-cerebellar model (orange) under sensory input noise (A), cortical state perturbation during the delay period (B), and global parameter perturbation (C). The coupled model remains substantially more robust across perturbation levels, with performance degrading progressively only at stronger corruption.

C.2 Motor robustness

The coupled cortico-cerebellar model (orange) showed substantially greater robustness than the cortex-only model (blue) across all motor perturbation conditions (Fig. 10). In the intact condition, the coupled model achieved a held-out sequence NMSE of 0.000996, compared with 0.2426 for the cortex-only model.

Perturbations to phase and amplitude observations produced graded increases in motor error. At the largest tested noise level (SD = 0.2), the coupled model reached NMSEs of 0.005760 under phase noise and 0.0144 under amplitude noise (Fig. 10A,B). Direct perturbation of the cortical state produced a similar dose-dependent effect, with NMSE increasing to 0.0206 at injected RMS = 0.6 (Fig. 10C). Global parameter perturbation caused the strongest degradation, with relative weight-noise RMS values of 0.1 and 0.3 increasing

NMSE to 0.0112 and 0.0902, respectively (Fig. 10D). Across all tested perturbation levels, the coupled model maintained substantially lower sequence error than the cortex-only model, although performance progressively degraded with increasing corruption strength.

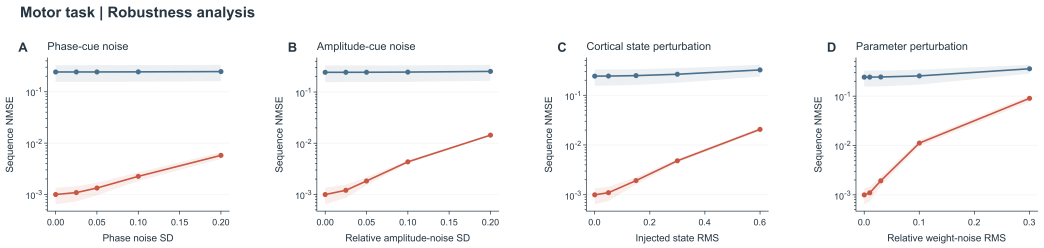


Figure 10: Robustness analysis in the motor task. Sequence NMSE of the cortex-only model (blue) and coupled cortico-cerebellar model (orange) under phase noise (A), amplitude noise (B), cortical state perturbation (C), and global parameter perturbation (D).

D Hyperparameter sensitivity

We next examined the sensitivity of the coupled cortico-cerebellar model to key architectural and optimization hyperparameters (Figs. 11–12). Each one-factor setting was evaluated with six trained replicas under the original training budget, while all remaining hyperparameters were fixed at their default values.

For the cognitive task, performance was broadly stable across architectural choices (Fig. 11A–C). Varying cortical width from $N = 24$ to 48 and 96 yielded held-out accuracies of 95.08%, 97.46%, and 100.00%, respectively. Changing the cerebellar feature width to 32 or 128 produced 97.79% and 99.61% accuracy, while gate-network widths of 8 and 32 yielded 99.32% and 99.02%. Performance was similarly robust to the cortical update coefficient, learning rate, and initial gate bias (Fig. 11D–F), with accuracies remaining above 93% across all tested settings. In contrast, generalization was strongly dependent on the auxiliary-loss weight (Fig. 11G): weights of 0, 1, 2, and 4 yielded held-out accuracies of 23.89%, 77.41%, 97.46%, and 100.00%, respectively. The relationship between seen-task fitting and held-out accuracy further showed that low training error alone did not guarantee successful compositional generalization (Fig. 11H).

Motor performance was also broadly stable across most one-factor perturbations, with held-out NMSE ranging from 5.23×10^{-4} to 2.36×10^{-3} across all tested settings (Fig. 12A–G). Increasing cortical width generally reduced trajectory error (Fig. 12A), whereas cerebellar feature width produced only modest variation across the tested range (Fig. 12B). Gate-network width showed greater sensitivity, with width 8 increasing NMSE to 2.36×10^{-3}, while wider gates reduced error (Fig. 12C). Performance remained low-error across cortical update coefficients and initial gate biases (Fig. 12D,F). Learning rate produced the clearest optimization effect, with rates of 0.001 and 0.004 yielding NMSEs of 2.02×10^{-3} and 5.23×10^{-4}, respectively (Fig. 12E). Varying the address-entropy penalty had comparatively little effect within the tested range (Fig. 12G). As in the cognitive task, seen-task NMSE was not tightly predictive of held-out trajectory error, indicating that successful training fit alone was insufficient to explain generalization performance (Fig. 12H).

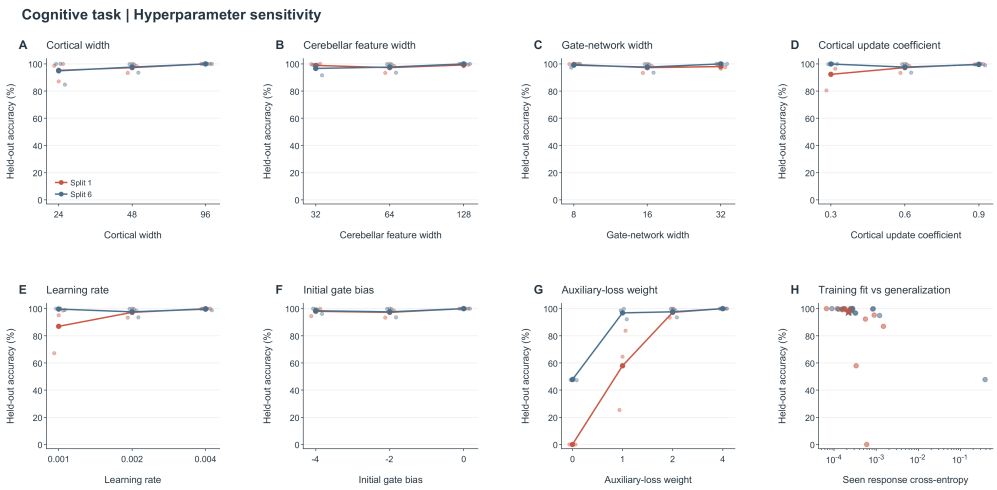


Figure 11: Hyperparameter sensitivity in the cognitive task. Held-out accuracy under one-factor variations of cortical width (A), cerebellar feature width (B), gate-network width (C), cortical update coefficient (D), learning rate (E), initial gate bias (F), and auxiliary-loss weight (G). (H) Relationship between seen-task response cross-entropy and held-out accuracy across tested settings. Red and blue lines denote the two evaluated task splits, with individual points showing trained replicas and stars indicating the default configuration.

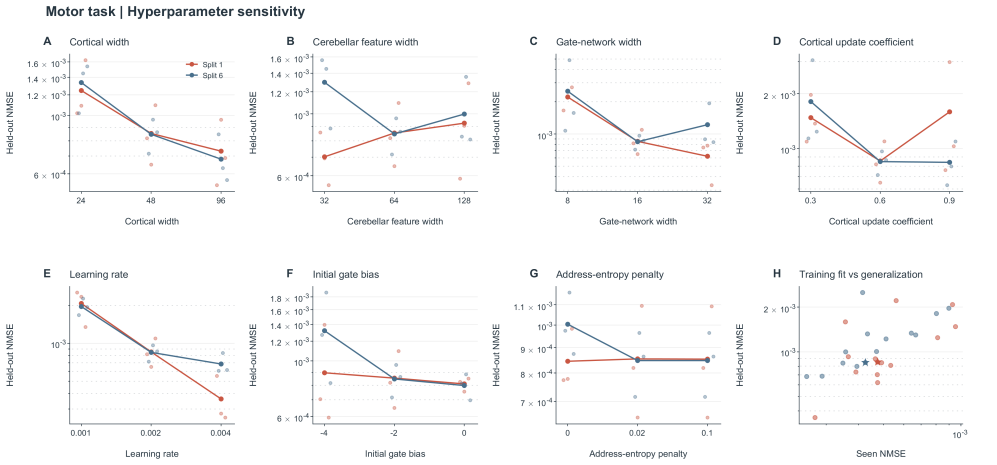


Figure 12: Hyperparameter sensitivity in the motor task. Held-out sequence NMSE under one-factor variations of cortical width (A), cerebellar feature width (B), gate-network width (C), cortical update coefficient (D), learning rate (E), initial gate bias (F), and address-entropy penalty (G). (H) Relationship between seen-task NMSE and held-out NMSE across tested settings. Red and blue lines denote the two evaluated task splits, with individual points showing trained replicas and stars indicating the default configuration.