CORTICO-CEREBELLAR MODULARITY AS AN ARCHITECTURAL INDUCTIVE BIAS FOR EFFICIENT TEMPORAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The cerebellum and cerebral cortex establish closely integrated loops that are involved in temporal processing; however, the manner in which their distinct circuit architectures influence learning remains ambiguous. This study investigates whether the integration of recurrent dynamics with a cerebellar-inspired feedforward pathway alters the learning of temporal computations. We enhance a recurrent neural network (RNN) by incorporating an expansion–compression module that feeds back into the recurrent dynamics, with both pathways being trained concurrently towards the same task objective. In the context of curriculum-trained temporal cognition tasks, this cortico-cerebellar RNN (CB-RNN) demonstrates more efficient and adaptive learning compared to parameter-matched recurrent networks across single-task, multitask, and task-switching scenarios. Limiting the plasticity of recurrent weights retains much of this learning advantage, suggesting that learning can increasingly be carried by the feedforward pathway while recurrent dynamics provide a substrate for computation. However, causal ablation indicates that the resulting recurrent representations remain reliant on cerebellar input. Mechanistic analyses reveal that this interaction is organised differently depending on task demands. Collectively, these findings illustrate that recurrent–feedforward modularity can enhance temporal learning without imposing a rigid functional decomposition between pathways, thereby allowing a task-dependent division of labour to emerge through joint optimisation.

1 INTRODUCTION

Biological neural systems demonstrate greater learning efficiency, flexibility, and robustness than artificial systems (Kudithipudi et al., 2022; Lake et al., 2017). This disparity may stem from architectural organisation; biological intelligence arises from heterogeneous systems with distinct architectures, functions, and timescales, which can serve as an inductive bias to promote efficient and generalisable learning (Goyal & Bengio, 2022; Marblestone et al., 2016).

The cortico-cerebellar system offers a well-motivated instance of such heterogeneous modularity. Cortical circuits support recurrent, context-dependent computation over time (Miller & Cohen, 2001; Mante et al., 2013), while the cerebellum (CB) is primarily a feedforward expansion-compression circuit, suited to rapid transformation and adaptive correction (Marr, 1969; Cayco-Gajic & Silver, 2019; Johansson et al., 2014). These systems interact through closed reciprocal loops (Middleton & Strick, 2001), and evidence from CB perturbation studies demonstrates that disrupting this interaction alters cortical activity and impairs cognitive performance (Gao et al., 2018; Deverett et al., 2019; Verpeut et al., 2023). We propose that modifying a recurrent cortical network with a CB-inspired system could enhance learning efficiency and task performance by distributing learning across diverse modules.

Here, we augment a cortex-inspired Elman RNN (Elman, 1990) with a CB-inspired feedforward module that integrates with recurrent dynamics (Fig. 1). In contrast to previous cortico-cerebellar models, in which the CB module is assigned specific loss functions (Agueci & Cayco-Gajic, 2025; Pemberton et al., 2024; Boven et al., 2023), both recurrent and CB pathways are trained jointly using the same task loss, without distinct objectives. This approach allows examination of whether archi-

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.

tectural heterogeneity influences computational distribution between pathways during learning. In temporal cognition tasks of increasing difficulty, the CB-RNN learns more efficiently than comparable recurrent networks in single-task, multitask, and task-switching scenarios. This efficiency persists even when recurrent-weight plasticity is limited, indicating that ongoing optimisation of recurrent weights is not essential for effective learning, though ablation shows that recurrent representations remain reliant on CB input. Mechanistic analyses reveal that this interaction does not lead to a single functional decomposition, but rather the CB role adapts depending on task-specific computational requirements. Collectively, these findings illustrate how recurrent–feedforward modularity serves as an architectural inductive bias, offering distinct computational pathways that enable flexible learning distribution.

2 Related Work

Biological inductive biases, modularity, and heterogeneity. It is increasingly suggested that biological intelligence relies on architectural inductive biases that shape representations and learning dynamics within a system (Marblestone et al., 2016; Goyal & Bengio, 2022). Modularity and heterogeneity can mitigate interference between learned functions, enhance robustness, and facilitate more efficient computation reuse (Goyal & Bengio, 2022; Salatiello, 2026). Evidence demonstrates that heterogeneity in neuronal properties boosts learning stability and robustness in spiking networks (Perez-Nieves et al., 2021), while biologically inspired augmentations of recurrent networks can modify learning dynamics and representational geometry (Soo et al., 2023). Learned modulatory signals, such as biases, can shape recurrent dynamics and facilitate flexible task performance without full synaptic plasticity (Mason et al., 2026; Williams et al., 2024; Delamare et al., 2024). Collectively, these findings suggest that architectural organisation, rather than just parameter count or learning rule, is an important determinant of learning efficiency in neural systems.

Cortico-cerebellar circuits as heterogeneous learning systems. The cortico-CB system provides a concrete instance of heterogeneous modular interaction. Cortical circuits facilitate recurrent, context-dependent computation and long-term temporal integration (Mante et al., 2013; Caligiore et al., 2017), while the CB is characterised by largely feedforward connectivity (Tanaka et al., 2020; Marr, 1969). The CB input layer expands incoming mossy fibre signals through numerous granule cells (GCs), generating a high-dimensional code conducive to pattern separation and rapid adaptive transformation of inputs (Cayco-Gajic & Silver, 2019; Person, 2019; Rancz et al., 2007), before being compressed through Purkinje cell (PC) outputs to deep cerebellar nuclei, completing closed cortico-CB loops (Middleton & Strick, 2001).

Computational models of cortico-cerebellar interaction. Recent modelling has begun to elucidate how CB-like modules can enhance learning in recurrent cortical networks. Boven et al. (2023) demonstrated that CB feedback can accelerate cortical learning by predicting feedback signals, thus alleviating delayed temporal credit assignment challenges. Pemberton et al. (2024) revealed that CB-driven cortical dynamics facilitate task acquisition, switching, and consolidation, with CB signals remaining effective even when cortical weights are static, suggesting that the CB can drive cortical computation without destabilising learned representations. Agueci & Cayco-Gajic (2025) further indicated that distributing plasticity across a rapid CB system and a slower cortical network improves motor adaptation by enabling quick error corrections via the CB, while slower cortical plasticity ensures long-term retention of adapted behaviours. Together, these studies demonstrate that CB pathways can enhance cortical learning when given explicit predictive or error-corrective targets.

3 Methods

3.1 Network Model

Baseline RNN. Compute resources are provided in Appendix A. The cortical component is an Elman-type recurrent neural network (Elman, 1990). Given input $x_t \in \mathbb{R}^d$ and previous hidden state $h_{t-1} \in \mathbb{R}^{N_h}$, the model updated its activity as

$$h_t = (1 - \alpha)h_{t-1} + \alpha\phi(W_h h_{t-1} + W_x x_t + b). \tag{1}$$

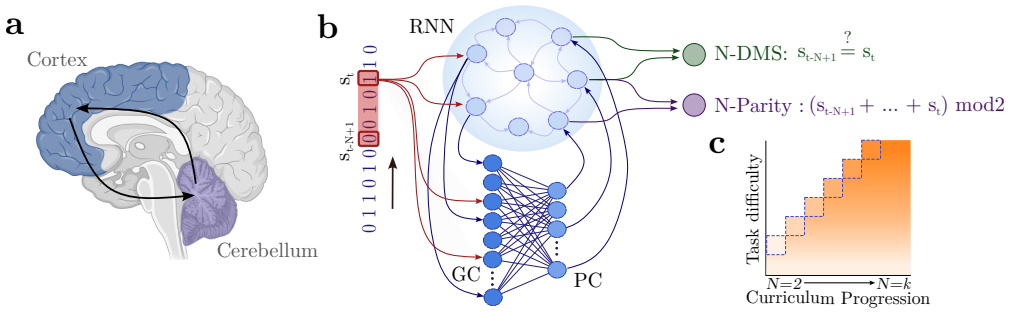


Figure 1: **The cortico-cerebellar RNN (CB-RNN) architecture and curriculum design.** (a) Cortico-CB loop motivating the architecture. (b) A feedforward CB module transforms recurrent state and task input. (c) Curriculum learning increases DMS and Parity difficulty from $N = 2$ to $N = k$, with outlined boxes denoting the current curriculum level.

where W_h and W_x are recurrent and input weights, respectively, and b is a recurrent bias. For task head k, the model output was obtained by applying a linear readout to the final recurrent hidden state,

$$\hat{\mathbf{y}}^{(k)} = W_{\text{out}}^{(k)} \mathbf{h}_T + \mathbf{b}_{\text{out}}^{(k)}, \tag{2}$$

where $W_{\text{out}}^{(k)} \in \mathbb{R}^{C \times H}$ and $\mathbf{b}_{\text{out}}^{(k)} \in \mathbb{R}^C$ are the parameters of readout head k, H is the recurrent hidden-state dimension, and C is the number of output classes. No output nonlinearity was applied before the cross-entropy loss. The CB pathway had no direct connection to the readout and could therefore influence the output only through its effect on the recurrent hidden state $\mathbf{h}_T$. The leak parameter $\alpha = 1/\tau$ controls the effective integration timescale and was fixed at $\tau = 1.5$. The activation function was a Leaky ReLU, denoted $[\cdot]_\gamma$ with a negative slope ($\gamma = 0.01$) (Maas et al., 2013). RNN-only baselines were scaled to match the number of trainable parameters in the fully-plastic CB-RNN variant, allowing comparisons at approximately matched model capacity (see Appendix B). GRU-based controls are reported in Appendix C.

Cortico-cerebellar model. The CB-RNN augments the baseline recurrent network with an additional feedforward module designed to provide a CB-inspired modulatory signal to the recurrent core (Fig. 1). In all main experiments, the recurrent hidden size was fixed at 64 units. The CB module used a GC expansion layer of 256 units; performance of additional GC sizes (64, 128, 512) is presented in Appendix D.

At each timestep, the CB module received a concatenated input $c_t = [h_{t-1}; x_t]$, comprising both the previous recurrent hidden state h_{t-1} and the task input x_t, motivated by evidence that the CB receives convergent information from the cortex and sensory pathways (Rancz et al., 2007; Person, 2019). Control comparisons using h_{t-1} and x_t alone are reported in Appendix E.1. The CB module consisted of two nonlinear feedforward transformations followed by a linear output projection:

$$z_t = \phi_{\text{CB}}(W_{\text{GC}} c_t + b_{\text{GC}}), \tag{3}$$

$$u_t = \phi_{\text{CB}}(W_{\text{PC}} z_t + b_{\text{PC}}), \tag{4}$$

$$b_{\text{cb},t} = W_{\text{out}} u_t + b_{\text{out}}. \tag{5}$$

Here, z_t and u_t denote the activity of the expanded GC-like and PC-like stages, respectively. The final linear projection produces the CB bias ($b_{\text{cb},t}$), which is returned to the recurrent core. ReLU nonlinearities were applied in the GC-like and PC-like stages. The recurrent update in the CB-RNN was therefore:

$$h_t = (1 - \alpha) h_{t-1} + \alpha [W_h h_{t-1} + W_x x_t + b + b_{\text{cb},t}]_\gamma. \tag{6}$$

where $b_{\text{cb},t}$ denotes the CB contribution and $[\cdot]_\gamma$ denotes the Leaky ReLU nonlinearity. This adaptation makes the recurrent update conditional on a learned feedforward transformation, relating the CB-RNN to recent architectures that augment recurrent networks with feedforward components to enrich recurrent hidden-state dynamics and improve sequence learning (Pascanu et al., 2014; Sun et al., 2024).

3.2 Task Suite

Models were evaluated on two binary sequence-processing tasks commonly used to benchmark recurrent memory (Khajehabdollahi et al., 2023): N-bit delayed match-to-sample (DMS) and N-bit Parity (Parity). Across tasks, models received variable-length binary sequences, with sequence length sampled as a function of the largest active N.

N-Bit Delayed Match-to-Sample. In DMS, the model received a binary input sequence and was required to report whether the final input bit matched the bit presented $N-1$ timesteps earlier. The task requires the model to retain inputs in memory, thus, increasing N increases memory demands.

N-Bit Parity. In the Parity task, the model was required to report the parity of the final N input bits, i.e. whether the number of ones in the previous N-bit window was odd or even. Unlike DMS, which depends on comparison with a specific past input, parity requires nonlinear integration over the full temporal window.

3.3 Training

3.3.1 Optimisation

Models were trained using stochastic gradient descent (SGD) with Nesterov momentum ($\mu = 0.1$) and cross-entropy loss, using learning rate $\eta = 0.01$ and gradient clipping at a maximum ℓ_2 norm of 7.5. Each epoch comprised 100 gradient steps over batches of 64 sequences, followed by evaluation on held-out batches. Main results were repeated across 8 random seeds unless otherwise stated. Full optimisation, initialisation, and evaluation details are provided in Appendix F.1.

3.3.2 Curriculum Learning

Task difficulty was controlled by the delay/window parameter N. Training commenced at $N = 2$ and advanced incrementally once performance exceeded 98% accuracy at test (Bengio et al., 2009; Khajehabdollahi et al., 2023). Three variants for CB-RNN models were evaluated. In the *simultaneous* variant, all modules trained concurrently. In the *full reservoir* variant, the recurrent core was frozen after training on $N = 2$, with all subsequent learning carried by the CB module and readout heads. In the *interleaved reservoir* variant, recurrent plasticity was additionally reinstated at fixed curriculum intervals: the RNN was trained whenever $(N-2) \bmod k = 0$ and refrozen once that level was solved, with $k = 10$ for DMS and $k = 5$ for Parity to account for their different curriculum ranges. The term reservoir here denotes these restricted-plasticity conditions rather than conventional reservoir computing. Although recurrent weights are frozen, the trainable CB pathway reads recurrent activity and feeds back into the recurrent update, such that the effective recurrent dynamics can still change through CB learning.

3.3.3 Training Regimes

Single-Task Training. In single-task training, each model was trained on one task at a time, with N increasing by one upon surpassing 98% evaluation accuracy. Training proceeded until $N = 100$ or the 1000-epoch limit was met. Performance was assessed using three metrics: the maximum N achieved within the training budget; the area under the curriculum progression curve (AUC), indicating cumulative learning progress; and the epochs needed to reach half of the global maximum N attained across all models ($N_{\text{global}}/2$), reflecting learning speed.

Multitask Training. In multitask training, models solved DMS and Parity simultaneously from shared input sequences, with separate task-specific readout heads each trained on a cross-entropy loss. Curriculum progression was synchronised across tasks, advancing only when both tasks exceeded the 98% threshold at the current N. The same metrics were used to summarise performance as in single-task training.

Task-Switch Training. Task-switching experiments alternated training between the DMS and Parity tasks across successive phases. In each phase, only the active task's readout head was trained, while both tasks were evaluated after every epoch. Task switches occurred when the active task reached the next multiple of $N_{\text{switch}} = 5$, with the incoming task resuming directly at that N rather than

restarting from $N = 2$. Consequently, this approach introduced the incoming task at a novel difficulty level, requiring adaptation without complete reacquisition. Curriculum progression ceased after approximately 600 epochs, and all analyses related to task switching were performed over the initial 650 epochs. AUC values were derived from task-specific curriculum trajectories and averaged across tasks. The speed of post-switch adaptation was measured as the mean number of epochs needed to progress from the phase-start N to $N + 1$ after a switch, averaged across DMS and Parity phases.

3.4 Activity Analysis

Cerebellar Ablation Analysis To test whether trained CB-RNN models relied on the CB pathway for task performance, the CB bias b_{cb} was silenced post-training without retraining any weights, and accuracy was re-evaluated across all curriculum levels N over 20 held-out batches of 64 sequences. The averaged effect of ablation was quantified as $\Delta\text{Acc} = \text{Acc}_{\text{full}} - \text{Acc}_{\text{ablated}}$, with positive values indicating performance loss following ablation of the CB pathway. To visualise task-relevant recurrent representations before and after ablation, final hidden-state activity was projected into three dimensions. A linear classifier fit to the output classes of the intact CB-RNN defined a class-discriminative axis, while PCA of activity orthogonal to this axis provided the remaining two dimensions. The projection was fit on intact activity and applied unchanged following CB ablation, providing a common coordinate system for comparison (Appendix I.1).

Task-dependent pathway mechanism analysis. To examine how the CB pathway contributes to recurrent dynamics, post-hoc analyses, tailored to the computation required by each task, were performed (full methods in Appendix J). For DMS, we measured one-step propagation of the task-relevant sample-memory direction using the local Jacobian $J_t = \partial h_t / \partial h_{t-1}$ and compared intact dynamics with dynamics in which the state dependence of the CB output was detached while preserving its forward value. For Parity, one-step causal interventions replaced the intact CB output with zero, or with CB outputs drawn from different transition conditions requiring either the same or opposite next parity, and measured the resulting movement along the next-parity representation axis.

4 Results

4.1 CB-RNNs outperform matched RNN-only baselines in efficiency and final performance

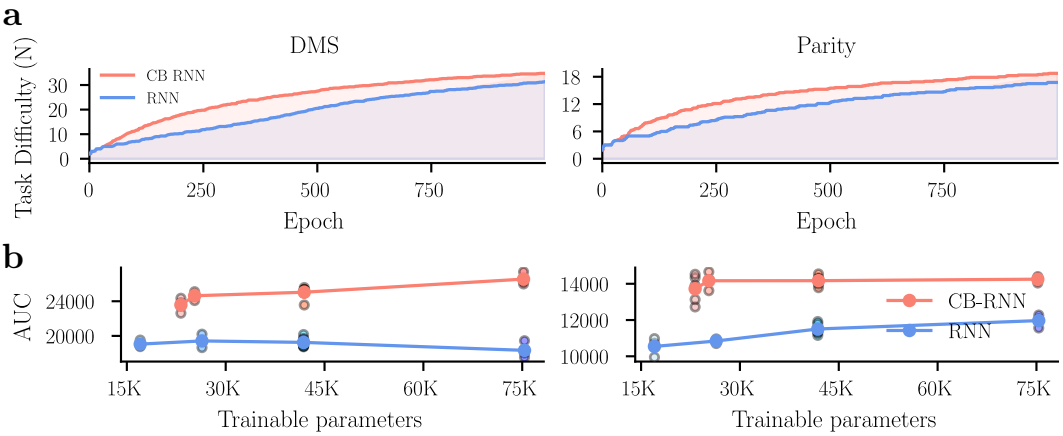


Figure 2: **CB-RNN modularity improves curriculum learning efficiency and final performance.** (a) Task difficulty N reached over training for DMS (left) and Parity (right); shading denotes AUC. (b) AUC across model scales. CB-RNN models consistently learned faster and reached higher difficulty levels than parameter-matched RNN-only baselines at all model sizes.

Across both tasks, CB-RNNs progressed through the curriculum more efficiently than parameter-matched RNN-only baselines (Fig. 2a), reaching higher final N values, accumulating larger AUCs, and approaching half of the global maximum N roughly twice as fast (see Appendix K). The magnitude of this advantage was not reproduced by replacing the feedforward CB pathway with a recurrent

module (Appendix E.2) indicating that generic modularity alone is insufficient. Furthermore, the CB-RNN advantage was also consistent across model scales: CB-RNNs achieved higher curriculum progression efficiency than matched RNN-only baselines at every model size examined, and smaller CB-RNNs matched or exceeded the performance of larger RNN-only models (Fig. 2b). Together, these results demonstrate that CB-inspired modularity improves both the speed and parameter efficiency with which recurrent networks acquire sequential tasks.

4.2 CB-RNN ADVANTAGE PERSISTS UNDER MULTITASK AND TASK-SWITCHING DEMANDS

Multitask. We next asked whether the CB-RNN advantage persisted when learning both tasks concurrently. Under multitask curriculum training, CB-RNNs again progressed faster, reached higher final N values, and accumulated larger AUCs than parameter-matched RNN-only baselines (Fig. 3a; Appendix K).

Task switching. To assess whether CB-RNNs could more flexibly adapt to a new task objective at an inherited difficulty level, models were trained in alternating phases on the tasks, with each switch presenting the incoming task at a previously unseen difficulty level. CB-RNNs accumulated greater cumulative learning progress than RNN-only baselines across the full switching experiment (AUC: $8{,}115 \pm 104$ vs. $6{,}393 \pm 268$; Fig. 3b), and adapted roughly twice as fast following each switch (35.19 ± 5.01 vs. 64.88 ± 6.57 epochs to advance one difficulty level from phase-start N). This suggests that in addition to improving learning progression speed, the CB module supports more efficient adaptation across task objectives.

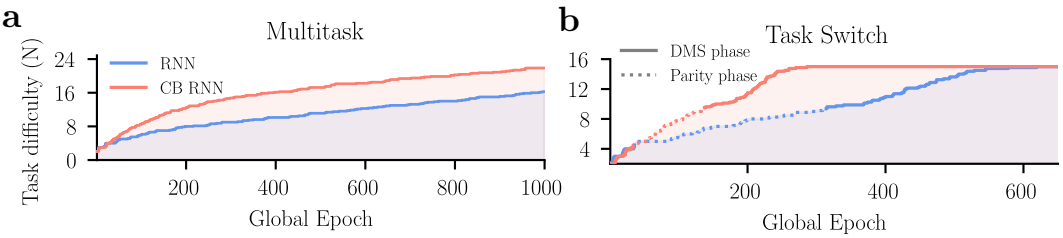


Figure 3: **CB-RNN models retain learning advantage under multitask and task-switching demands.** (a) Curriculum progression during multitask training; shading denotes AUC. (b) Averaged curriculum progression during task switching, dashed lines mark Parity epochs. CB-RNNs accumulated more learning progress and adapted faster following each switch than RNN-only baselines.

4.3 CB-RNN EFFICIENCY BENEFIT PERSISTS UNDER RESTRICTED RECURRENT PLASTICITY

To investigate whether the CB-RNN efficiency advantage depended on continuous optimisation of both the recurrent core and CB module, we trained two variants that restricted recurrent plasticity: a full reservoir model, in which the RNN was frozen after initial training, and an interleaved reservoir model, in which recurrent learning was periodically reinstated at fixed intervals. The term reservoir denotes restricted plasticity, rather than being analogous to conventional reservoir computing. Under single-task and multitask curricula, both reservoir variants outperformed the RNN-only baseline in AUC and speed of approach to the global half-maximum N, despite having a largely frozen recurrent core (Fig. 4a–b; Appendix K, Table 4). The full reservoir fell below the RNN-only baseline in final N reached for single-task training, consistent with the cost of extended recurrent freezing at high difficulty levels, while the interleaved reservoir model recovered this deficit on both tasks. The fully trained CB-RNN remained the strongest model on efficiency metrics, but was comparable to the interleaved variant in final N for DMS. The limitation of recurrent plasticity was accompanied by a reorganisation of population activity within the architecture (Appendices G–H). Representational dimensionality increasingly shifted towards CB populations at greater task difficulties, while the temporal structure across recurrent and CB populations also changed under restricted recurrent plasticity. Collectively, these findings suggest a redistribution of computational burden when recurrent-weight plasticity is limited.

In the task-switching paradigm, the full reservoir again performed above the RNN-only baseline but below the CB-RNN across all metrics (AUC: $7{,}772 \pm 99$ vs. $6{,}393 \pm 268$ and $8{,}115 \pm 104$; post-switch adaptation: 41.94 ± 4.09 vs. 64.88 ± 6.57 and 35.19 ± 5.01 epochs; Fig. 4c). Thus, recurrent

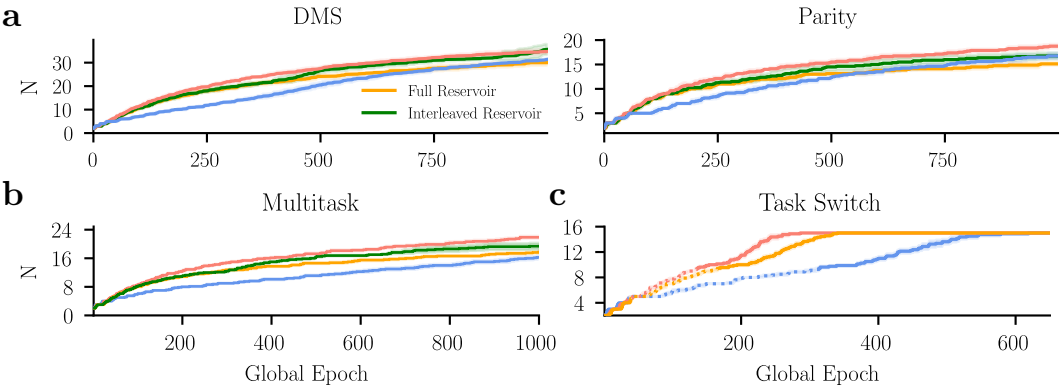


Figure 4: **Cerebellar module drives learning efficiency even when RNN plasticity is restricted.** (a) Single-task and (b) multitask curriculum progression; (c) task-switching trajectories. Reservoir variants retained most of the learning-speed advantage over the RNN-only baseline, showing that subsequent learning can be carried by the CB pathway despite restricted recurrent-weight plasticity.

plasticity contributes to performance, particularly at higher difficulty, but continuous recurrent-weight optimisation is not required for the CB-RNN learning-speed advantage.

4.4 Cerebellar ablation disrupts task performance and recurrent representations

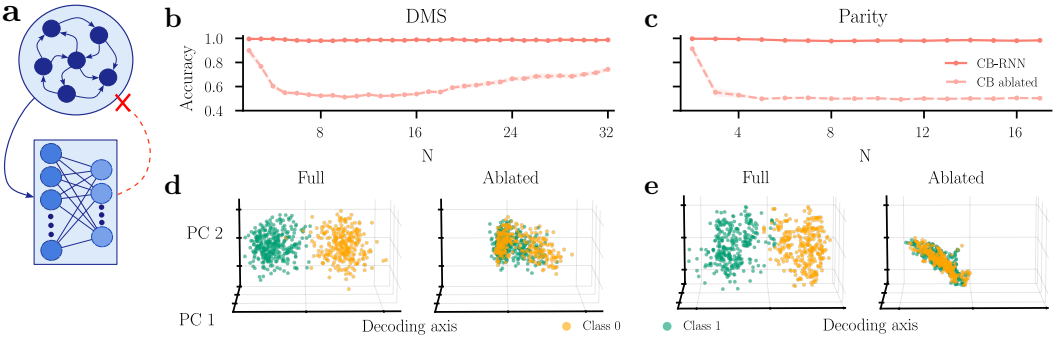


Figure 5: **Cerebellar bias supports task performance and class-discriminative recurrent representations.** (a) Ablation schematic. (b, c) Accuracy across N for intact and CB-ablated models on DMS and Parity. CB ablation substantially reduced accuracy, with partial recovery on DMS but not Parity. Lines show mean across runs; shading indicates SD over $n = 8$ seeds. (d, e) Three-dimensional projections of recurrent hidden-state activity at representative N values, defined by a class-discriminative axis and two orthogonal principal components fit to the intact CB-RNN and applied unchanged following ablation. CB ablation reduced class separation.

To test whether trained CB-RNN models relied on the CB pathway, we silenced b_{cb} post-training without retraining any weights and re-evaluated performance across curriculum levels (Fig. 5a–c). Ablation substantially impaired DMS (mean $\Delta\text{Acc} = 0.37 \pm 0.09$), although performance partially recovered at higher Ns (Fig. 5b). In contrast, Parity accuracy remained near chance following ablation across N (mean $\Delta\text{Acc} = 0.45 \pm 0.10$; Fig. 5c). Retraining the readout following ablation did not restore performance, excluding a simple readout–recurrent mismatch (Appendix I.3).

Ablation also disrupted task-relevant recurrent representations. Three-dimensional projections defined from the intact CB-RNN showed reduced class separation following ablation (Fig. 5d–e), consistent with decreases in linear separability measured by receiver operating characteristic area under the curve (ROC-AUC; Appendices I.1– I.2). Together, these results show that trained recurrent computation relies on the CB pathway, but reveal a task-dependent difference whereby CB-reliance partially weakens across the DMS curriculum, while Parity remains strongly dependent on CB output.

4.5 CEREBELLAR CONTRIBUTIONS TO RECURRENT DYNAMICS ARE TASK DEPENDENT

The contrasting effects of CB ablation suggested that the CB pathway may contribute differently to the computations required by DMS and Parity. We therefore analysed its causal contribution to local recurrent-state dynamics (Fig. 6; full methods and analyses in Appendix J).

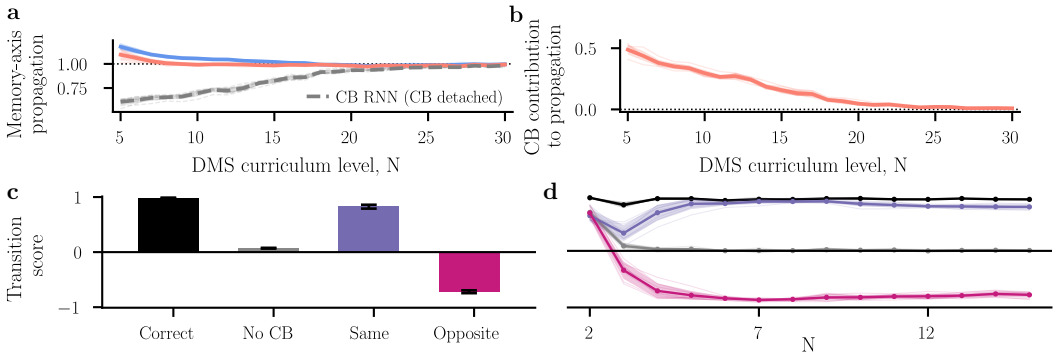


Figure 6: **The CB pathway makes task-dependent contributions to recurrent dynamics.** (a) DMS propagation of the task-relevant sample-memory direction across successive delay transitions for RNN-only networks, the intact CB-RNN, and the CB-RNN after detaching the state dependence of the CB pathway. (b) State-dependent CB contribution to DMS memory propagation, $Q_{\mathrm{CB}} = Q_{\mathrm{full}} - Q_{\mathrm{det}}$, which decreases across the curriculum. (c) Normalised Parity transition scores after replay with the original CB output (Correct), no CB output (No CB), an output from a different condition requiring the same next parity (Same), or an output requiring the opposite next parity (Opposite). (d) Transition scores across Parity curriculum levels; the transition-specific ordering is maintained from $N = 3$ onwards. Lines show means across $n = 8$ networks and error bars/shading indicates SD. Full analysis details are provided in Appendix J.

DMS. In DMS, the intact CB-RNN robustly propagated the task-relevant sample-memory direction across curriculum levels (Fig. 6a). At lower N, detaching the state dependence of the CB output substantially reduced this propagation, but the difference progressively diminished with increasing task difficulty (Fig. 6a–b). By approximately $N = 25$, the direct state-dependent CB contribution was close to zero and propagation was predominantly carried by the recurrent term. Because this recurrent term is evaluated at operating points partly determined by the forward CB output, the analysis indicates a decreasing state-dependent CB contribution rather than complete independence from the CB pathway.

Parity. For Parity, the intact CB output drove recurrent activity towards the representation required at the next timestep, whereas removing the CB output almost abolished this transition (Fig. 6c). Substituting a CB output from another condition requiring the same next parity largely restored the correct state update, while an output associated with the opposite next parity redirected activity towards the incorrect representation. This ordering persisted from $N = 3$ onwards (Fig. 6d), showing that the effect depends on transition-specific structure in the CB signal rather than simply its presence or mean magnitude. Furthermore, removing task input from the CB pathway abolishes the learning advantage on Parity but largely preserves it on DMS (Appendix E.1), consistent with the CB output specifying the required next state by combining the current input with state information. Together, these analyses reveal a task-dependent division of labour: the state-dependent CB contribution to memory propagation decreases across learning in DMS, whereas Parity remains reliant on CB-mediated signals that specify the required recurrent-state transition.

5 DISCUSSION

Cortico-cerebellar modularity as an inductive bias for learning. The CB-RNN advantage reflects not simply increased capacity or generic modularity, but the heterogeneous organisation of recurrent and feedforward computation. CB-RNNs outperformed parameter-matched RNNs in learning efficiency and performance, and non-heterogeneous modular architectures did not confer the

same advantage, suggesting that combining recurrent dynamics with a feedforward auxiliary pathway provides a useful architectural inductive bias (Marblestone et al., 2016; Salatiello, 2026; Goyal & Bengio, 2022). Reservoir experiments further showed that the CB pathway could support efficient learning with a fixed recurrent core, while periodic recurrent plasticity provided an additional benefit. This is consistent with a functional division in which the CB supports rapid adaptation alongside slower recurrent optimisation (Caligiore et al., 2017; Tanaka et al., 2020; Agueci & Cayco-Gajic, 2025), and with prior work on learned biases in ANNs (Mason et al., 2026; Williams et al., 2024). Finally, advantages in multitask and task-switching settings suggest that this organisation facilitates adaptation under competing or changing task demands, consistent with proposed roles of cortico-CB interactions in flexible behaviour (Müller et al., 2023; Verpeut et al., 2023; Peterburs et al., 2018; Stoodley et al., 2017).

Architectural modularity enables an emergent division of labour. A central feature of the CB-RNN is that functional specialisation between the recurrent and CB pathways is not predefined; both pathways are trained end-to-end based on the same task objective, without specific losses or roles. However, the findings reveal distinct roles that emerge through learning. Limiting recurrent plasticity retains much of the learning efficiency while reorganising the dimensionality and temporal structure of activity (Appendices G–H). Architectural heterogeneity thus enables various computational regimes for learning solutions, allowing computation allocation to adapt to available plasticity and task requirements. This aligns with broader views of modularity as an inductive bias that structures learned computation without enforcing a fixed decomposition (Marblestone et al., 2016; Goyal & Bengio, 2022; Salatiello, 2026). Crucially, this flexibility does not correspond to independent modular processing, as CB ablation disrupted task-relevant recurrent representations, indicating that the recurrent solution develops learned dependence on CB input. This aligns more with an interacting-systems perspective of cortico-cerebellar computation than with a model where the CB performs a distinct computation downstream of the cortex (Gao et al., 2018; Israely et al., 2025; Tanaka et al., 2020). The observed task-dependent mechanisms indicate that the CB pathway does not confer a single functional role; the same architectural asymmetry supported various forms of recurrent–feedforward interaction across tasks. This contrasts modular approaches in which specialised functions are assigned to modules a priori. Here, the architecture dictates the computational resources and interactions available for learning, while optimisation shapes their utilisation. This perspective may also reconcile differing proposed roles for cortico-cerebellar interaction. CB contributions have been characterised as both continuously shaping cortical dynamics and diminishing as learned behaviours increasingly rely on cortical circuitry (Herzfeld et al., 2014; Pemberton et al., 2024; Tanaka et al., 2020; Agueci & Cayco-Gajic, 2025). The present results imply these principles need not be mutually exclusive. A heterogeneous recurrent–feedforward system can facilitate both persistent online interaction and computation redistribution between pathways, contingent on task demands. Thus, the advantages of heterogeneous modularity may stem not from assigning fixed functions to modules, but from broadening the ways learning can organise and redistribute computation across interacting pathways.

Limitations. Various modelling choices influence the interpretation and scope of these findings. The model is a coarse abstraction that omits biological details such as CB-specific teaching signals and subcortical relays via the pons and thalamus (Roš et al., 2025; Hoang et al., 2023; Habas et al., 2019; Ito et al., 1982). The tasks and recurrent core are deliberately simple, GRU controls showed an attenuated CB benefit (Appendix C), and the use of backpropagation through time lacks biological plausibility (Lillicrap et al., 2020), limiting generalisation to richer settings. Future work should test whether the observed efficiency gains persist with greater biological realism, including CB-specific plasticity, more complex models, and more plausible learning rules (Murray, 2019; Bellec et al., 2020; Agueci & Cayco-Gajic, 2025).

Future directions. These findings suggest several directions for future work. The mechanistic results predict that cortical dependence on CB input for memory maintenance should decline with training on delay-like tasks, whereas during continual-updating computations CB output should carry information about the upcoming task-relevant cortical state beyond the current input condition.

AI USE STATEMENT

In this work, we used AI to help implement standard components of our code, and to provide feedback on the manuscript for clarity and readability.

REPRODUCIBILITY STATEMENT

Model architectures, tasks, and training procedures are specified in text, with parameter matching in Appendix B, task generation and the training algorithm in Appendix F.1, and analysis procedures in the corresponding appendices. The number of random seeds is stated for each result. Code implementing the model architectures, tasks, training procedures, and analyses reported in this work are available in an anonymised repository at anonymous.4open.science/r/Cortico-cerebellar_RNNs.

REFERENCES

Leonardo Agueci and N. Alex Cayco-Gajic. Distributed learning across fast and slow neural systems supports efficient motor adaptation, June 2025. URL https://www.biorxiv.org/content/10.1101/2025.06.01.657238v1. Pages: 2025.06.01.657238 Section: New Results.

Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications*, 11(1):3625, July 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-17236-y. URL https://www.nature.com/articles/s41467-020-17236-y.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, June 2009. Association for Computing Machinery. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553380. URL https://dl.acm.org/doi/10.1145/1553374.1553380.

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, January 2023. ISSN 2041-1723. doi: 10.1038/s41467-022-35658-8. URL https://www.nature.com/articles/s41467-022-35658-8.

Daniele Caligiore, Giovanni Pezzulo, Gianluca Baldassarre, Andreea C. Bostan, Peter L. Strick, Kenji Doya, Rick C. Helmich, Michiel Dirkx, James Houk, Henrik Jörntell, Angel Lago-Rodriguez, Joseph M. Galea, R. Chris Miall, Traian Popa, Asha Kishore, Paul F. M. J. Verschure, Riccardo Zucca, and Ivan Herreros. Consensus Paper: Towards a Systems-Level View of Cerebellar Function: the Interplay Between Cerebellum, Basal Ganglia, and Cortex. *The Cerebellum*, 16(1):203–229, February 2017. ISSN 1473-4230. doi: 10.1007/s12311-016-0763-3. URL https://doi.org/10.1007/s12311-016-0763-3.

N. Alex Cayco-Gajic and R. Angus Silver. Re-evaluating Circuit Mechanisms Underlying Pattern Separation. *Neuron*, 101(4):584–602, February 2019. ISSN 1097-4199. doi: 10.1016/j.neuron.2019.01.044.

Geoffroy Delamare, Douglas Feitosa Tomé, and Claudia Clopath. Intrinsic neural excitability biases allocation and overlap of memory engrams. *The Journal of Neuroscience*, 44(21):e0846232024, 2024.

Ben Deverett, Mikhail Kislin, David W. Tank, and Samuel S.-H. Wang. Cerebellar disruption impairs working memory during evidence accumulation. *Nature Communications*, 10(1):3128, July 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-11050-x. URL https://www.nature.com/articles/s41467-019-11050-x.

Jeffrey L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, March 1990. ISSN 0364-0213, 1551-6709. doi: 10.1207/s15516709cog1402_1. URL https://onlinelibrary.wiley.com/doi/10.1207/s15516709cog1402_1.

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, November 2018. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-018-0633-x. URL https://www.nature.com/articles/s41586-018-0633-x.

Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478 (2266):20210068, October 2022. ISSN 1364-5021. doi: 10.1098/rspa.2021.0068. URL https://doi.org/10.1098/rspa.2021.0068.

Christophe Habas, Mario Manto, and Pierre Cabaraux. The Cerebellar Thalamus. *The Cerebellum*, March 2019. ISSN 1473-4222, 1473-4230. doi: 10.1007/s12311-019-01019-3. URL http://link.springer.com/10.1007/s12311-019-01019-3.

David J. Herzfeld, Damien Pastor, Adrian M. Haith, Yves Rossetti, Reza Shadmehr, and Jacinta O'Shea. Contributions of the cerebellum and the motor cortex to acquisition and retention of motor memories. *NeuroImage*, 98:147–158, September 2014. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2014.04.076. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC4099269/.

Huu Hoang, Shinichiro Tsutsumi, Masanori Matsuzaki, Masanobu Kano, Mitsuo Kawato, Kazuo Kitamura, and Keisuke Toyama. Dynamic organization of cerebellar climbing fiber response and synchrony in multiple functional components reduces dimensions for reinforcement learning. *eLife*, 12:e86340, September 2023. ISSN 2050-084X. doi: 10.7554/eLife.86340. URL https://elifesciences.org/articles/86340.

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, March 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-57832-4. URL https://www.nature.com/articles/s41467-025-57832-4.

Masao Ito, Masaki Sakurai, and Pavich Tongroach. Climbing fibre induced depression of both mossy fibre responsiveness and glutamate sensitivity of cerebellar Purkinje cells. *The Journal of Physiology*, 324(1):113–134, March 1982. ISSN 0022-3751, 1469-7793. doi: 10.1113/jphysiol.1982.sp014103. URL https://physoc.onlinelibrary.wiley.com/doi/10.1113/jphysiol.1982.sp014103.

Fredrik Johansson, Dan-Anders Jirenhed, Anders Rasmussen, Riccardo Zucca, and Germund Hesslow. Memory trace and timing mechanism localized to cerebellar Purkinje cells. *Proceedings of the National Academy of Sciences of the United States of America*, 111(41):14930–14934, October 2014. ISSN 1091-6490. doi: 10.1073/pnas.1415371111.

Sina Khajehabdollahi, Roxana Zeraati, Emmanouil Giannakakis, Tim Jakob Schäfer, Georg Martius, and Anna Levina. Emergent mechanisms for long timescales depend on training curriculum and affect performance in memory tasks. *arXiv preprint arXiv:2309.12927*, 2023. doi: 10.48550/ARXIV.2309.12927. URL https://arxiv.org/abs/2309.12927. Version Number: 3.

Dhireesha Kudithipudi, Mario Aguilar-Simon, Jonathan Babb, Maxim Bazhenov, Douglas Blackiston, Josh Bongard, Andrew P. Brna, Suraj Chakravarthi Raja, Nick Cheney, Jeff Clune, Anurag Daram, Stefano Fusi, Peter Helfer, Leslie Kay, Nicholas Ketz, Zsolt Kira, Soheil Kolouri, Jeffrey L. Krichmar, Sam Kriegman, Michael Levin, Sandeep Madireddy, Santosh Manicka, Ali Marjaninejad, Bruce McNaughton, Risto Miikkulainen, Zaneta Navratilova, Tej Pandit, Alice Parker, Praveen K. Pilly, Sebastian Risi, Terrence J. Sejnowski, Andrea Soltoggio, Nicholas Soures, Andreas S. Tolias, Darío Urbina-Meléndez, Francisco J. Valero-Cuevas, Gido M. van de Ven, Joshua T. Vogelstein, Felix Wang, Ron Weiss, Angel Yanguas-Gil, Xinyun Zou, and Hava Siegelmann. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence*, 4(3): 196–210, March 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00452-0. URL https://www.nature.com/articles/s42256-022-00452-0.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253,

2017. ISSN 0140-525X, 1469-1825. doi: 10.1017/S0140525X16001837. URL https://www.cambridge.org/core/product/identifier/S0140525X16001837/type/journal_article.

Timothy P. Lillicrap, Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, June 2020. ISSN 1471-003X, 1471-0048. doi: 10.1038/s41583-020-0277-3. URL https://www.nature.com/articles/s41583-020-0277-3.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning*, volume 30, Atlanta, Georgia, 2013.

Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, November 2013. ISSN 1476-4687. doi: 10.1038/nature12742. URL https://www.nature.com/articles/nature12742.

Adam H. Marblestone, Greg Wayne, and Konrad P. Kording. Toward an Integration of Deep Learning and Neuroscience. *Frontiers in Computational Neuroscience*, 10, September 2016. ISSN 1662-5188. doi: 10.3389/fncom.2016.00094. URL https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2016.00094/full.

David Marr. A theory of cerebellar cortex. *The Journal of Physiology*, 202(2):437–470, June 1969. ISSN 0022-3751, 1469-7793. doi: 10.1113/jphysiol.1969.sp008820. URL https://physoc.onlinelibrary.wiley.com/doi/10.1113/jphysiol.1969.sp008820.

Kai Mason, Navid Akbari, Aaron Gruber, and Wilten Nicola. Versatile Learning without Synaptic Plasticity in a Spiking Neural Network. *bioRxiv*, January 2026. doi: 10.64898/2026.01.01.697291. URL http://biorxiv.org/content/early/2026/01/02/2026.01.01.697291.abstract.

Frank A. Middleton and Peter L. Strick. Cerebellar Projections to the Prefrontal Cortex of the Primate. *The Journal of Neuroscience*, 21(2):700–712, January 2001. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.21-02-00700.2001. URL https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.21-02-00700.2001.

Earl K. Miller and Jonathan D. Cohen. An Integrative Theory of Prefrontal Cortex Function. *Annual Review of Neuroscience*, 24(1):167–202, March 2001. ISSN 0147-006X, 1545-4126. doi: 10.1146/annurev.neuro.24.1.167. URL https://www.annualreviews.org/doi/10.1146/annurev.neuro.24.1.167.

James M Murray. Local online learning in recurrent networks with random feedback. *eLife*, 8:e43299, May 2019. ISSN 2050-084X. doi: 10.7554/eLife.43299. URL https://elifesciences.org/articles/43299.

Eli J. Müller, Fulvia Palesi, Kevin Y. Hou, Joshua Tan, Thomas Close, Claudia A. M. Gandini Wheeler-Kingschott, Egidio D'Angelo, Fernando Calamante, and James M. Shine. Parallel processing relies on a distributed, low-dimensional cortico-cerebellar architecture. *Network Neuroscience*, 7(2):844–863, June 2023. ISSN 2472-1751. doi: 10.1162/netn_a_00308. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10312290/.

Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to Construct Deep Recurrent Neural Networks, April 2014. URL http://arxiv.org/abs/1312.6026. arXiv:1312.6026 [cs].

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.

Nicolas Perez-Nieves, Vincent C. H. Leung, Pier Luigi Dragotti, and Dan F. M. Goodman. Neural heterogeneity promotes robust learning. *Nature Communications*, 12(1):5791, October 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-26022-3. URL https://www.nature.com/articles/s41467-021-26022-3.

Abigail L. Person. Corollary Discharge Signals in the Cerebellum. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, 4(9):813–819, September 2019. ISSN 2451-9022. doi: 10.1016/j.bpsc.2019.04.010. URL https://www.sciencedirect.com/science/article/pii/S2451902219301065.

Jutta Peterburs, David Hofmann, Michael P. I. Becker, Alexander M. Nitsch, Wolfgang H. R. Miltner, and Thomas Straube. The role of the cerebellum for feedback processing and behavioral switching in a reversal-learning task. *Brain and Cognition*, 125:142–148, August 2018. ISSN 0278-2626. doi: 10.1016/j.bandc.2018.07.001. URL https://www.sciencedirect.com/science/article/pii/S0278262618302100.

Ede A. Rancz, Taro Ishikawa, Ian Duguid, Paul Chadderton, Séverine Mahon, and Michael Häusser. High-fidelity transmission of sensory information by single cerebellar mossy fibre boutons. *Nature*, 450(7173):1245–1248, December 2007. ISSN 0028-0836. doi: 10.1038/nature05995. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC5881887/.

Hana Roš, Yizhou Xie, Sadra Sadeh, and R. Angus Silver. Population activity of mossy fibre axon input to the cerebellar cortex during behaviours, March 2025. URL https://www.biorxiv.org/content/10.1101/2025.03.23.644738v1. Pages: 2025.03.23.644738 Section: New Results.

Alessandro Salatiello. Modularity is the Bedrock of Natural and Artificial Intelligence, February 2026. URL http://arxiv.org/abs/2602.18960. arXiv:2602.18960 [cs] version: 1.

Wayne W.M. Soo, Vishwa Goudar, and Xiao-Jing Wang. Training biologically plausible recurrent neural networks on cognitive tasks with long-term dependencies, October 2023. URL http://biorxiv.org/lookup/doi/10.1101/2023.10.10.561588.

Catherine J. Stoodley, Anila M. D'Mello, Jacob Ellegood, Vikram Jakkamsetti, Pei Liu, Mary Beth Nebel, Jennifer M. Gibson, Elyza Kelly, Fantao Meng, Christopher A. Cano, Juan M. Pascual, Stewart H. Mostofsky, Jason P. Lerch, and Peter T. Tsai. Altered cerebellar connectivity in autism and cerebellar-mediated rescue of autism-related behaviors in mice. *Nature Neuroscience*, 20 (12):1744–1751, December 2017. ISSN 1546-1726. doi: 10.1038/s41593-017-0004-1. URL https://www.nature.com/articles/s41593-017-0004-1.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

Hirokazu Tanaka, Takahiro Ishikawa, Jongho Lee, and Shinji Kakei. The Cerebro-Cerebellum as a Locus of Forward Model: A Review. *Frontiers in Systems Neuroscience*, 14:19, April 2020. ISSN 1662-5137. doi: 10.3389/fnsys.2020.00019. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC7160920/.

Jessica L. Verpeut, Silke Bergeler, Mikhail Kislin, F. William Townes, Ugne Klibaite, Zahra M. Dhanerawala, Austin Hoag, Sanjeev Janarthanan, Caroline Jung, Junuk Lee, Thomas J. Pisano, Kelly M. Seagraves, Joshua W. Shaevitz, and Samuel S.-H. Wang. Cerebellar contributions to a brainwide network for flexible behavior in mice. *Communications Biology*, 6(1):605, June 2023. ISSN 2399-3642. doi: 10.1038/s42003-023-04920-0. URL https://www.nature.com/articles/s42003-023-04920-0.

Ezekiel Williams, Alexandre Payeur, Avery Hee-Woon Ryoo, Thomas Jiralerspong, Matthew G. Perich, Luca Mazzucato, and Guillaume Lajoie. Expressivity of Neural Networks with Random Weights and Learned Biases, 2024. URL https://arxiv.org/abs/2407.00957. Version Number: 3.

Roxana Zeraati, Tatiana A. Engel, and Anna Levina. A flexible Bayesian framework for unbiased estimation of timescales. *Nature Computational Science*, 2(3):193–204, March 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00214-3. URL https://www.nature.com/articles/s43588-022-00214-3.

A COMPUTE RESOURCES.

All experiments were run on a local workstation equipped with three NVIDIA GPUs: one GeForce GTX TITAN X and two GeForce RTX 2080 Ti GPUs. Experiments used CUDA 12.6. All simulations and analyses were run with Python 3.9.6, PyTorch 2.8.0, NumPy 2.0.2, scikit-learn 1.6.1, and Matplotlib 3.9.4.

B PARAMETER MATCHING ACROSS MODEL ARCHITECTURES

To ensure that performance differences between recurrent-only and CB-RNN models could not be attributed to different trainable parameter counts, recurrent-only baselines were matched to CB-RNN models by total trainable parameter count. Matching was performed separately for Elman and GRU architectures, as their recurrent cores differ in parameter count.

Elman RNN. For an Elman RNN with input size I and hidden size H, the recurrent core contains input-to-hidden and hidden-to-hidden weights and biases:

$$P_{\text{core}}^{\text{Elman}} = IH + H + H^2 + H.$$

RNN-RNN modular control. For the modular organisation control (Appendix E.2), the CB module was replaced by a second Elman RNN. For auxiliary hidden size H_{aux}, its parameter count was

$$P_{\text{aux}} = (H + I)H_{\text{aux}} + H_{\text{aux}} + H_{\text{aux}}^2 + H_{\text{aux}} + H_{\text{aux}}H + H.$$

GRU. For a GRU with input size I and hidden size H, the recurrent core contains update, reset, and candidate pathways, each with input weights, recurrent weights, and two bias vectors:

$$P_{\text{core}}^{\text{GRU}} = 3\big(IH + H^2 + 2H\big).$$

CB module and readout. For the reported CB-RNN models, the CB module received the concatenated input $c_t = [h_{t-1}; x_t]$ of dimension $H + I$. With GC dimension G and PC and CB output dimensions fixed to the hidden size H, the CB module contributed

$$P_{\text{CB}} = (H + I)G + G + GH + H + H^2 + H.$$

This corresponds to the GC transformation, the PC transformation, and the final CB output transformation back to hidden-state dimensionality.

The multi-head readout, with M heads and K classes, contributed

$$P_{\text{readout}} = M(HK + K),$$

including readout biases. Total CB-RNN parameters were therefore

$$P_{\text{CB-RNN}} = P_{\text{core}} + P_{\text{CB}} + P_{\text{readout}},$$

where P_{core} is the architecture-specific recurrent core term above.

For each CB-RNN configuration, the matched recurrent-only hidden size was selected as

$$H^* = \arg\min_H \left| P_{\text{recurrent-only}}(H) - P_{\text{CB-RNN}} \right|,$$

where

$$P_{\text{recurrent-only}}(H) = P_{\text{core}}(H) + M(HK + K).$$

This produced recurrent-only baselines with approximately matched trainable capacity, so comparisons reflected architectural organisation rather than parameter count alone.

Parameter Counts.

Table 1: **Model variants and approximate trainable parameter counts.**

Model	Trainable parameters
CB-RNN (H= 64, GC= 256)	41,922
RNN-only (H= 202)	41,816
Full Reservoir CB-RNN	37,634
GC = 64/128/512 → matched H*	128/160/272

C GRU-BASED CONTROLS

To test whether the CB-RNN advantage depended on the use of an Elman recurrent core, the single-task DMS and Parity experiments were repeated using a gated recurrent unit (GRU) architecture. The GRU models were implemented to match the Elman experiments as closely as possible, including the same curriculum-learning procedure, task generation, optimisation settings, multi-head readout structure, CB module design, and parameter-matched recurrent-only comparisons.

The baseline GRU used standard update and reset gates, with a candidate hidden state computed from the current input and reset-gated previous hidden state. In the CB-GRU model, the cerebellar module received the same input configuration as in the main CB-RNN experiments and produced a hidden-sized bias vector. This bias was injected only into the candidate hidden-state preactivation, rather than into the update or reset gates:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}),$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1}),$$
$$\tilde{h}_t = \phi(W_n x_t + U_n(r_t \odot h_{t-1}) + b_{\mathrm{cb},t}),$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t.$$

Here, ϕ is the candidate activation function and $b_{\mathrm{cb},t}$ is the cerebellar bias. The CB bias was restricted to the candidate pathway to preserve the GRU gating mechanism while providing the closest architecture-specific analogue of the additive recurrent preactivation bias used in the Elman CB-RNN. The GRU controls also used an additional norm cap on the CB bias. This was included because the bias was injected into the candidate-state pathway of a gated recurrent transition, where large additive signals could saturate the candidate activation and interact strongly with the update gate. The cap therefore kept the CB contribution on the same scale as the GRU candidate preactivation.

These controls tested whether CB-inspired biasing improved temporal learning beyond the specific dynamics of an Elman RNN. Performance was compared between CB-GRU models and parameter-matched GRU-only baselines under the same single-task curriculum procedure used for the Elman models.

Together, the GRU controls revealed an important boundary condition on the CB-RNN advantage (Fig. 7, Table 2).

Table 2: **Curriculum-learning performance for GRU-based single-task controls.** Values are mean ± SD across $n = 3$ seeds. Half-maximum epochs were measured relative to the global maximum N across models within each task.

Task	Model	Max N	Epochs to $N_{\mathrm{global}}/2$	AUC
DMS	CB-GRU	32.33 ± 0.58	141.67 ± 6.11	$11{,}009 \pm 264$
	GRU-only	33.67 ± 1.15	163.00 ± 15.10	$10{,}700 \pm 480$
Parity	CB-GRU	16.33 ± 0.58	163.00 ± 14.00	$5{,}389 \pm 227$
	GRU-only	13.67 ± 0.58	189.33 ± 26.39	$4{,}602 \pm 267$

On DMS, the CB-GRU reached the global half-maximum difficulty faster and showed a slightly higher AUC than the GRU-only baseline, but reached a slightly lower final difficulty, indicating that the strong CB benefit observed with Elman RNNs was substantially attenuated in a gated recurrent architecture. On Parity, however, CB-GRU outperformed the GRU-only baseline across all summary

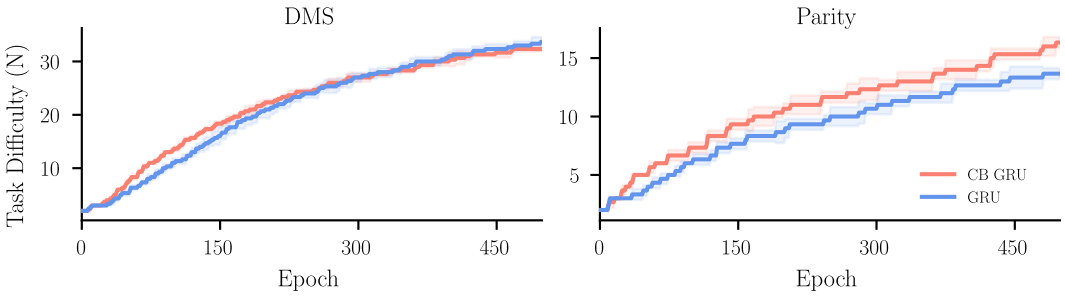


Figure 7: **GRU-based controls reveal an attenuated and task-dependent CB benefit.** Single-task DMS and Parity curriculum progression was compared between CB-GRU models and parameter-matched GRU-only baselines. On DMS, CB-GRU showed faster early curriculum progression but reached a similar final difficulty to the GRU-only baseline. On Parity, CB-GRU progressed faster and reached higher final task difficulty. Lines show means across random seed repeats; shaded regions show SD over $n = 3$ seeds.

metrics, reaching higher final N, progressing faster, and accumulating larger AUC. These results suggest that CB-inspired biasing can still improve temporal learning in GRUs, particularly under stronger nonlinear integration demands, but that the effect is not architecture-independent. Rather, the CB pathway may be most beneficial when the recurrent core lacks intrinsic gating mechanisms for regulating temporal integration.

D MODEL-SIZE VARIATION IN SINGLE-TASK TRAINING

Single-task training was repeated across CB-RNNs with GC-layer sizes of 64, 128, 256, and 512, using 256 in the main analysis, alongside parameter-matched RNN-only baselines. Across both DMS and Parity, CB-RNNs generally progressed faster through the curriculum and reached higher task difficulty than matched RNN-only models (Fig. 8). Thus, the learning advantage was preserved across model capacities and was not driven by a single choice of CB module size.

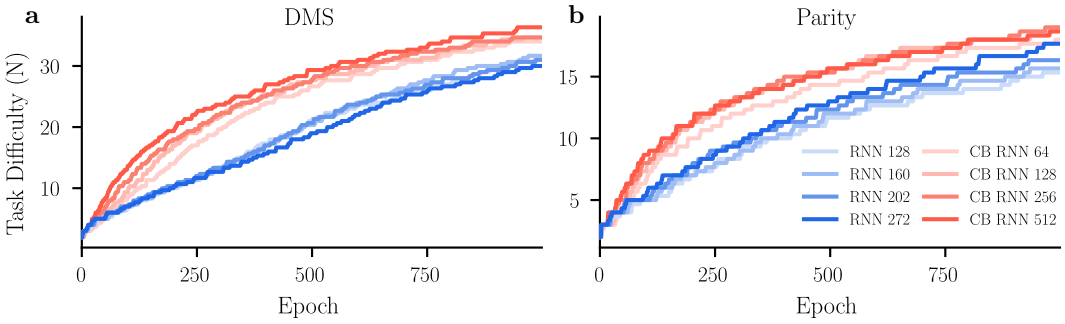


Figure 8: **CB-RNN performance advantage is preserved across model sizes.** Single-task curriculum progression for DMS (a) and Parity (b) across CB-RNNs with different GC-layer sizes and approximately parameter-matched RNN-only baselines. CB-RNNs generally progressed faster and reached higher task difficulty across model scales. Lines reflect mean averaged progression over $n = 3$ seeds.

To test whether GC-layer expansion contributed to the CB-RNN advantage, we compared the AUC difference between each CB-RNN and its parameter-matched RNN-only baseline for a non-expansive CB module ($GC = 64$) and the main expanded model ($GC = 256$). The AUC advantage was larger for the expanded GC model, particularly on DMS, suggesting that expansion in the CB pathway contributes to the performance benefit.

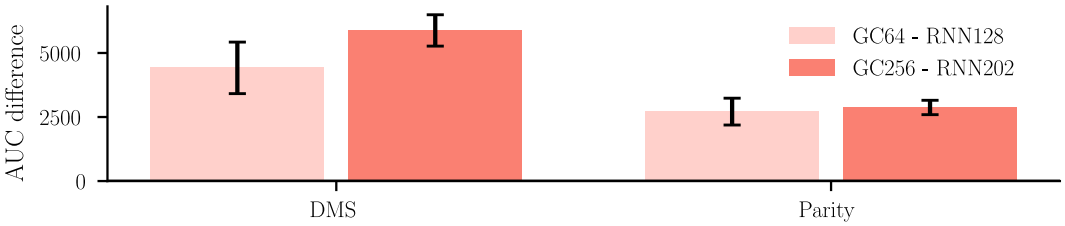


Figure 9: **GC-layer expansion increases the CB-RNN AUC advantage.** Difference in curriculum AUC between CB-RNNs and their approximately parameter-matched RNN-only baselines for non-expansive ($GC = 64$) and expanded ($GC = 256$) CB modules. The expanded GC model showed a larger AUC advantage, most clearly on DMS. Bars show mean AUC difference; error bars show SD across $n = 3$ seeds.

E Architectural Controls

E.1 Input Source Controls

To test whether CB-RNN performance depended on the information available to the CB module, single-task DMS and Parity runs were repeated using restricted CB input configurations. In the main analyses, the CB module received the concatenated recurrent state and task input, $[h_{t-1}; x_t]$. Control models instead received either the recurrent hidden state alone (h_{t-1} only; no direct task input) or the task input alone (x_t only; no recurrent state input).

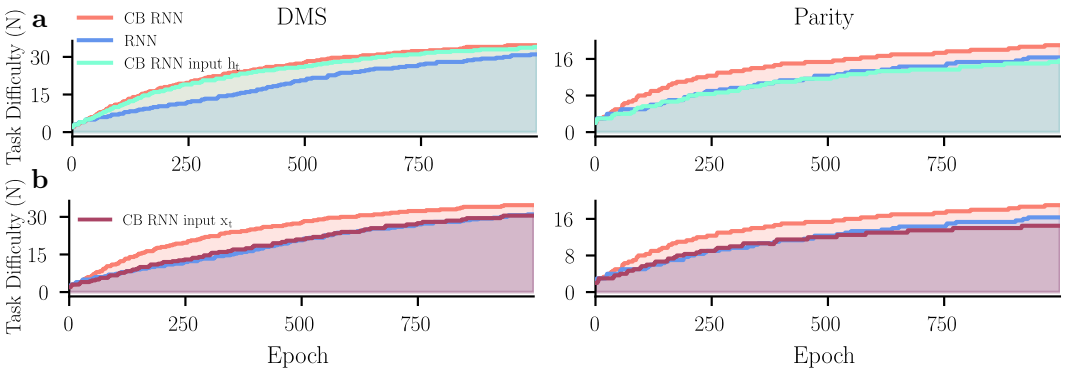


Figure 10: **CB-RNN performance depends on the information available to the cerebellar module.** Single-task DMS and Parity runs were repeated with restricted CB input configurations. (a) When CB input is only hidden state activity h_{t-1}, DMS performance is largely preserved but Parity performance falls to match baseline. (b) When CB input is restricted to task input x_t only, performance across DMS and Parity falls to match baseline. Lines reflect mean across $n = 3$ random seeds, shading indicates AUC.

The effect of removing CB access to hidden state input h_{t-1} was task-dependent, with performance largely preserved in DMS, but falling to match RNN-only in Parity (Fig. 10a). When removing task input x_t, performance across both tasks fell to be similar to RNN-only baseline, performing slightly worse than the baseline in Parity (Fig. 10b). These results suggest that recurrent state information is particularly important for the CB contribution to DMS, while Parity benefits from access to both recurrent state and the current task input.

E.2 Modular Organisation Control

To determine whether the learning advantage of the CB-RNN could be explained by modularity alone, an additional modular recurrent architecture was constructed in which the feedforward CB module was replaced by a second recurrent module (RNN+RNN, $H = 139$). The auxiliary module received the previous recurrent hidden state h_{t-1} and current task input x_t. Its recurrent state was passed

through a linear output layer, as in the CB-RNN, to generate a 64-dimensional signal that was added to the current recurrent pre-activation. It was implemented as a leaky Elman RNN with the same time constant and activation function as the main recurrent network. Its size was selected to approximately parameter match the CB-RNN and RNN-only controls, and learning rates and all other training and evaluation procedures were unchanged.

The RNN+RNN architecture achieved greater curriculum AUC than the RNN-only baseline but did not recover the learning efficiency of the CB-RNN on either DMS or Parity (Fig. 11). Thus, generic modularity contributes to learning efficiency but does not fully account for the CB-RNN advantage, suggesting that the computational organisation of the auxiliary pathway also matters.

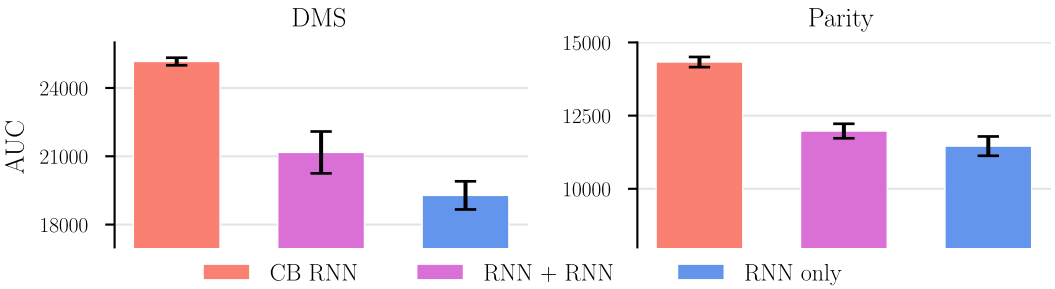


Figure 11: **Generic modularity does not fully account for the CB-RNN learning advantage.** Curriculum learning efficiency (AUC) for the CB-RNN, RNN-RNN modular control (RNN+RNN), and RNN only baseline on DMS and Parity. The RNN-RNN architecture achieved greater AUC than the RNN only model but did not recover the learning efficiency of the CB-RNN. Bars show mean and error bars show SD across $n = 3$ seeds.

F Initialisation and task generation details

F.1 Initialisation

RNN weights were initialised using Kaiming-uniform. The hidden state was initialised as $h_0 \sim \mathcal{U}(0, 0.1)$ per batch. For CB-RNN models, CB biases were initialised to zero; input→GC weights were sampled from $\mathcal{N}(0, 0.01^2)$, and GC→PC and PC→CB output layer weights from $\mathcal{N}(0, 0.1^2)$.

F.2 Task generation

Single task All tasks used binary input sequences $x_t \in \{0, 1\}$ generated online during training and evaluation. For each batch, a sequence length M was sampled as a function of the largest active curriculum level $N_{\max}$:

$$M \in \{N_{\max} + 2, \ldots, 4N_{\max} + 1\}.$$

Longer curriculum levels were thus paired with longer input sequences, ensuring the final target always had sufficient temporal context. Input bits were sampled independently from a balanced Bernoulli distribution, $x_t \sim \text{Bernoulli}(0.5)$, and each batch contained $B = 64$ independently generated sequences arranged as a tensor $X \in \mathbb{R}^{M \times B \times 1}$.

N-bit delayed match-to-sample. The label for a sequence of length M was a single binary value indicating whether the final input bit matched the bit $N - 1$ timesteps earlier:

$$y_N^{\text{DMS}} = [s_{t-N+1} = s_t],$$

where $t = M$ denotes the final timestep.

N-bit parity. The label was the parity of the final N input bits:

$$y_N^{\text{Parity}} = \left(\sum_{i=0}^{N-1} x_{M-i} \right) \bmod 2.$$

Batch outputs. Each task-generation call returned the input tensor $X \in \mathbb{R}^{M \times B \times 1}$ and a list of label tensors, one per active curriculum level N, each of shape B.

Algorithm 1: CB-RNN curriculum training protocol.

Data: Task $\mathcal{T} \in \{\mathrm{DMS}, \mathrm{Parity}\}$, initial difficulty $N = 2$, epoch budget $E_{\max} = 1000$, advancement threshold $\tau = 0.98$
Initialise: Initialise RNN weights, CB weights, readout heads, and optimiser
for *epoch* $e = 1, \ldots, E_{\max}$ **do**
  `// Training phase`
  for *gradient step* $g = 1, \ldots, 100$ **do**
    Sample sequence length M as a function of current difficulty N;
    Generate batch $X \in \{0, 1\}^{M \times B \times 1}$ with $B = 64$;
    Compute task labels y_N for the current difficulty level;
    for *timestep* $t = 1, \ldots, M$ **do**
      Compute CB bias $b_{\mathrm{cb},t} = \mathrm{CB}([h_{t-1}; x_t])$;
      Update recurrent state:

$$h_t = (1 - \alpha)h_{t-1} + \alpha[W_h h_{t-1} + W_x x_t + b + b_{\mathrm{cb},t}]_\gamma$$

    Compute readout prediction from final hidden state h_M;
    Compute cross-entropy loss $\mathcal{L}$;
    Backpropagate through the unrolled sequence;
    Clip gradients to ℓ_2 norm ≤ 7.5;
    Update trainable parameters with SGD and Nesterov momentum;
  `// Evaluation and curriculum advancement`
  Evaluate mean accuracy $\hat{a}$ on 50 newly generated batches;
  if $\hat{a} \geq \tau$ **then**
    Save checkpoint for current N;
    Increase difficulty: $N \leftarrow N + 1$;
  if $N > 100$ **then**
    break;

Multitask For multitask training, the same binary input batch was used for DMS and Parity. Separate labels were generated for each task at the current shared curriculum level, and separate task-specific readout heads produced the corresponding predictions. The training loss was the sum of the DMS and Parity cross-entropy losses. Curriculum advancement occurred only when both task accuracies exceeded 98% on evaluation batches.

Task-switching batches. In task-switching experiments, only the currently active task contributed to the training loss. The active task-specific readout head and shared network parameters were updated during that phase, while both task heads were evaluated after each epoch. Switches occurred when the active task reached the next multiple of $N_{\mathrm{switch}} = 5$, and the incoming task resumed at that inherited difficulty.

G Population timescale analysis

Intrinsic timescales were estimated from population autocorrelations under stationary random input. At each N checkpoint, the model was driven by $T = 20{,}000$ timesteps of i.i.d. Bernoulli(0.5) input across $S = 8$ independent streams, discarding the first 500 timesteps as burn-in. A population activity trace was formed per stream by summing across units, and autocorrelations were estimated using a Fast Fourier Transform (FFT)-based estimator, corrected for lag-dependent sample overlap, averaged across streams, and normalised by the zero-lag value. Population timescale τ_{pop} was defined as the first lag at which the normalised autocorrelation fell below e^{-1}, evaluated up to a maximum lag of 200 timesteps (Zeraati et al., 2022). Because the CB pathway contains no recurrent state of its own, temporal correlations in CB activity ultimately derive from the recurrent state and input it receives.

These measurements therefore characterise temporal structure expressed by each population rather than independently generated memory timescales.

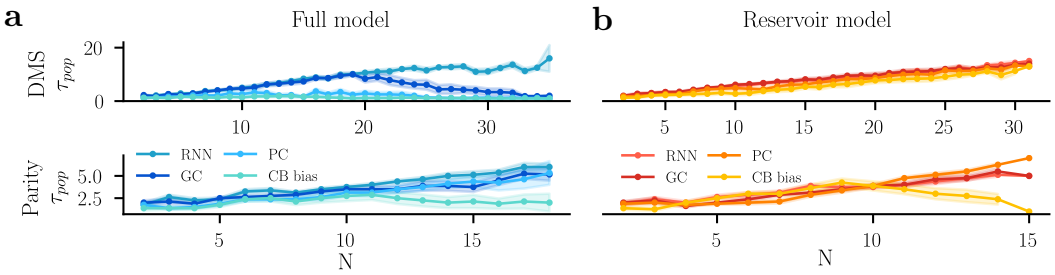


Figure 12: **Restricted recurrent plasticity alters the temporal structure of activity across recurrent and cerebellar populations.** Population autocorrelation timescale (τ_{pop}) across task difficulty N for DMS (upper) and Parity (lower), in fully plastic (a) and reservoir (b) models. In the fully plastic model, the recurrent hidden state shows the longest timescale at high N under restricted recurrent plasticity, the temporal structure expressed across populations is reorganised. Lines show means; shading indicates SD. These population timescales should not be interpreted as independent memory timescales of the feedforward CB pathway.

In the fully plastic CB-RNN on DMS, the recurrent hidden state and GC activity showed the longest population timescales, increasing together until approximately $N = 18$, after which CB population timescales declined while the recurrent timescale remained high (Fig. 12a). Parity showed a weaker version of this pattern. Under restricted recurrent plasticity, the profiles changed: DMS populations generally increased with N, whereas in Parity the PC population expressed the longest autocorrelation timescale over part of the curriculum. Because CB activity is a feedforward transformation of the recurrent state and current input, these changes are interpreted as altered readout and feedback of temporally structured recurrent modes, rather than memory stored independently within the CB module. The profiles complement the changes in representational dimensionality described in Appendix H.

H PCA FOR POPULATION DIMENSIONALITY ANALYSIS

Activity was extracted separately for each recorded population: the recurrent hidden state, GC layer, PC layer, and CB bias signal. For each population, activity was stored as a tensor of shape $T \times B \times D$, where T is sequence length, B is batch size, and D is the dimensionality of the population. Activity was flattened across the temporal and batch dimensions to produce a samples-by-units matrix $X \in \mathbb{R}^{TB \times D}$. Each matrix was mean-centred across samples before PCA was applied separately for each population, checkpoint, task, and random seed. PCA was fit without specifying a fixed number of components, so all available components up to $\min(TB, D)$ were retained before dimensionality metrics were computed.

Dimensionality was summarised as the fraction of available population dimensions required to explain 95% of the variance, $\tilde{d}_{95} = \frac{d_{95}}{D}$, where d_{95} is the smallest number of principal components required to explain 95% of the variance, and D is the dimensionality of the corresponding population. Thus, $\tilde{d}_{95}$ quantifies the proportion of the available dimensions occupied by the dominant population activity, with lower values indicating more low-dimensional activity and higher values indicating activity distributed across a larger fraction of the available dimensions.

Population dimensionality exhibited distinct trajectories across tasks and training regimes (Fig. 13). In the fully trainable CB-RNN, recurrent dimensionality increased with task difficulty for both DMS and Parity. Across both tasks, dimensionality within the CB pathway showed a non-monotonic profile, where the CB bias produced initially expanded before decreasing at higher N, despite continued expansion in the recurrent module. This compression was not observed when the recurrent network was held fixed in the reservoir model, where dimensionality followed largely monotonic trajectories across all populations.

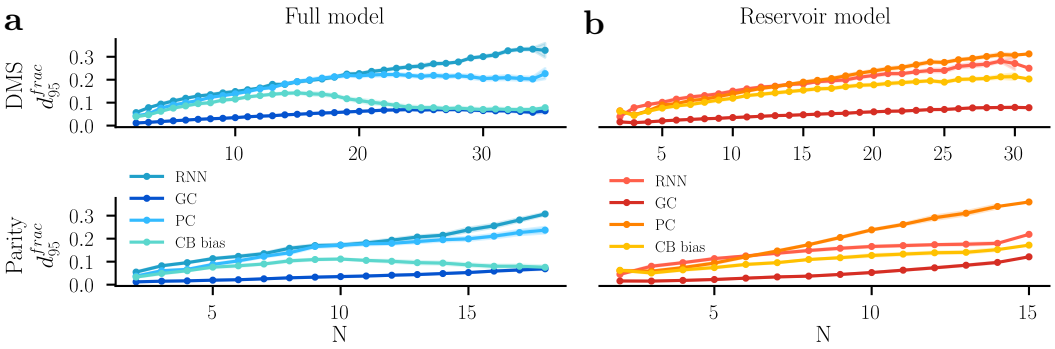


Figure 13: **Representational dimensionality depends on task and training regime.** Normalised population dimensionality, $\tilde{d}_{95} = \frac{d_{95}}{D}$ (denoted here as d_{95}^{frac}), across curriculum difficulty N for DMS (top) and Parity (bottom). (a) In the fully trainable CB-RNN, recurrent dimensionality increases with task difficulty, while DMS exhibits a non-monotonic change in CB-bias dimensionality at higher N. (b) When plasticity is restricted in the recurrent network (Reservoir model, see Section 3.3.2 for details), the compression of dimensionality in CB bias is absent, and population dimensionality follows predominantly monotonically increasing trajectories.

Together with the population-timescale analysis (Appendix G), these findings suggest that the architectural organisation biases the distribution of representational dimensionality across modules, but does not fix their computational roles. Alterations to plasticity in the network lead to redistributions of representational burden across the network, increasing the dimensionality of modules which, in the fully plastic network, would decrease their respective representational dimensionality.

I Cerebellar ablation analyses

I.1 Supervised projection of recurrent representations

To visualise task-relevant structure in the recurrent population, final-timestep hidden-state activity was projected into a three-dimensional linear subspace. Analyses were performed separately for each task. For each trial i, the final recurrent hidden state $\mathbf{h}_i \in \mathbb{R}^D$ was extracted together with its binary output class $y_i \in \{0, 1\}$, where D denotes the number of recurrent units. Intact and CB-ablated activity was evaluated on identical batches of trials.

The projection basis was estimated exclusively from intact CB-RNN activity. Hidden states from both conditions were first centred using the mean hidden state of the intact condition,

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{h}_i^{\text{Full}}, \tag{7}$$

such that

$$\tilde{\mathbf{h}}_i = \mathbf{h}_i - \boldsymbol{\mu}. \tag{8}$$

An ℓ_2-regularised logistic-regression classifier was then fit to predict the output class from centred intact hidden states. The normalised classifier weight vector,

$$\mathbf{v}_{\text{dec}} = \frac{\mathbf{w}}{\|\mathbf{w}\|_2}, \tag{9}$$

defined the first, task-discriminative decoding axis, where $\mathbf{w} \in \mathbb{R}^D$ denotes the fitted classifier weights.

To characterise variation orthogonal to the decoding axis, its contribution was removed from each centred intact hidden state:

$$\mathbf{r}_i = \left(\mathbf{I} - \mathbf{v}_{\text{dec}} \mathbf{v}_{\text{dec}}^{\top}\right) \tilde{\mathbf{h}}_i^{\text{Full}}, \tag{10}$$

where $\mathbf{I}$ is the $D \times D$ identity matrix. Singular value decomposition was applied to the resulting residual activity, and the first two right singular vectors were retained as the two leading principal-component directions, $\mathbf{v}_{\mathrm{PC1}}$ and $\mathbf{v}_{\mathrm{PC2}}$. The three-dimensional projection basis was therefore

$$\mathbf{B} = [\mathbf{v}_{\mathrm{dec}} \quad \mathbf{v}_{\mathrm{PC1}} \quad \mathbf{v}_{\mathrm{PC2}}] \in \mathbb{R}^{D\times 3}. \tag{11}$$

Final hidden states were projected as

$$\mathbf{z}_i = \mathbf{B}^\top (\mathbf{h}_i - \boldsymbol{\mu}) \in \mathbb{R}^3. \tag{12}$$

Thus, the first dimension captures class-discriminative activity identified by the linear decoder, while the remaining two dimensions capture the largest directions of intact recurrent-state variance orthogonal to this decoding axis.

To assess changes in recurrent geometry following CB ablation, the mean, decoding axis, and principal-component directions estimated from intact activity were held fixed and applied unchanged to CB-ablated activity. No projection parameters were refit following ablation. Intact and ablated recurrent states were therefore visualised within a common coordinate system defined by the intact task representation. Projection bases were estimated independently for each task. Because the projection was defined from intact activity, displacement of ablated states in this coordinate system can reflect both changes in class information and shifts in the operating point; the cross-validated decoders in Appendix I.2, refit within each condition, provide a complementary measure of class separability.

I.2 Quantification of class separability

To quantify the effect of CB ablation on class-discriminative recurrent representations, linear separability was measured directly in the full hidden-state space. For each independently trained network, a logistic-regression decoder was trained to classify output class from final-timestep hidden activity using stratified five-fold cross-validation. Performance was summarised as mean ROC-AUC across folds for each network and is reported as mean $\pm$ SD across $n = 8$ networks.

Table 3: **CB ablation reduces linear separability of recurrent representations.** ROC-AUC of linear decoders trained on final-timestep hidden activity using stratified five-fold cross-validation. Values are mean $\pm$ SD across $n = 8$ networks.

Task	N	Full model	CB ablated
DMS	24	0.999 ± 0.002	0.820 ± 0.030
Parity	8	0.999 ± 0.001	0.614 ± 0.065

CB ablation therefore reduced class-discriminative information in the recurrent state for both tasks, with a larger reduction for Parity.

I.3 Readout retraining during ablation

To test whether the performance loss following CB pathway ablation could be explained solely by a mismatch between recurrent representations and the trained output readout, we performed a readout-only recovery analysis. Ablations were repeated as in Section 3.4, but with readout training enabled. Performance was compared across three conditions: the intact CB-RNN, the CB-disabled model before readout retraining, and the CB-disabled model after readout-only retraining.

Across both DMS and parity, disabling the CB pathway caused a large drop in accuracy relative to the intact CB-RNN (Fig. 14). Readout-only retraining failed to restore performance to the intact model level. In DMS, readout retraining produced partial recovery at larger Ns, as expected given the findings with readout heads frozen, but a substantial gap from the full CB-RNN remained across task difficulties. In Parity, readout retraining provided little additional recovery beyond the zero-shot CB-disabled condition, with both remaining close to chance-level performance across most Ns. Thus, the behavioural impairment caused by CB pathway ablation cannot be explained simply by a poorly matched output readout.

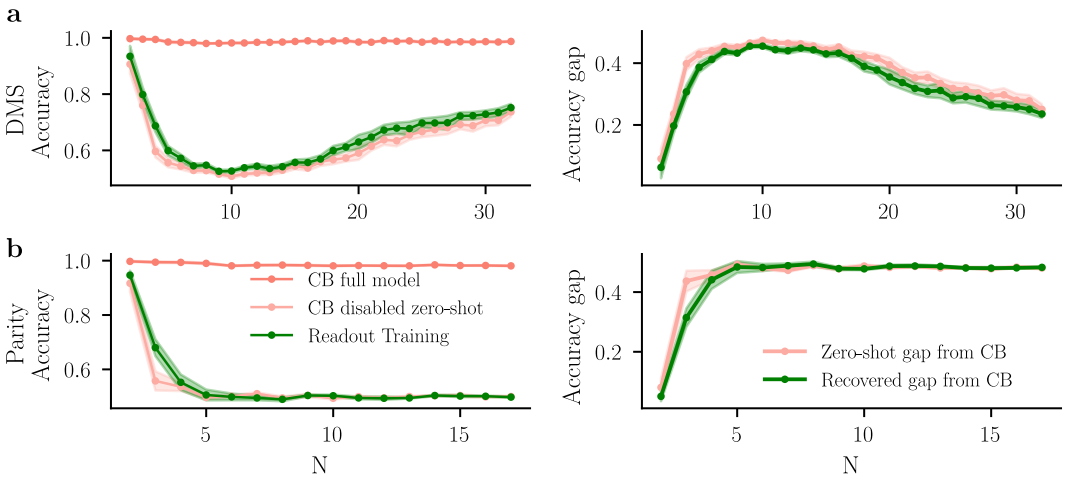


Figure 14: **Readout-only retraining does not rescue cerebellar pathway ablation.** (a) In DMS, disabling the CB pathway strongly reduced accuracy, and retraining only the output readout produced only partial recovery. (b) In Parity, readout retraining produced little recovery after CB pathway ablation. In both tasks, the residual gap from the intact CB-RNN remained substantial, indicating that the CB pathway supports task-relevant recurrent representations. Lines show mean across runs; shaded regions indicate SEM over $n = 8$ seeds.

J TASK-DEPENDENT MECHANISMS OF CEREBELLAR CONTRIBUTION

Additional post-hoc exploratory analyses were performed on trained DMS and Parity networks, designed to identify candidate mechanisms by which the CB bias supports task performance through modulating the RNN module. Because DMS and Parity have different computational demands, the contribution of the CB pathway was analysed separately for each task.

J.1 DMS

DMS requires information about the sample identity to be maintained in the recurrent state across a delay. Ablating the CB pathway disrupted the separation of recurrent representations of task variables (Fig. 5), suggesting that the CB pathway contributes to maintaining task-relevant representations. To investigate how this contribution arises from the local network dynamics, the propagation of the sample-memory representation across successive delay steps was analysed and decomposed into CB-mediated and remaining components.

Defining the task-relevant memory direction. Analyses were performed on curriculum checkpoints $N \in \{5, \ldots, 30\}$ from 8 independently trained networks. For each network and checkpoint, 2048 fixed trials were divided equally into independent direction-estimation and evaluation sets. At each aligned delay timestep t, recurrent states in the direction-estimation set were grouped according to the sample identity that had to be retained. The task-relevant memory direction was defined as the unit-normalised difference between the corresponding condition-averaged recurrent states,

$$\mathbf{m}_t = \frac{\boldsymbol{\mu}_{A,t} - \boldsymbol{\mu}_{B,t}}{\|\boldsymbol{\mu}_{A,t} - \boldsymbol{\mu}_{B,t}\|_2}.$$

The direction was consistently oriented according to sample identity. Because $\mathbf{m}_t$ was estimated independently at each timestep, the task-relevant direction was allowed to change over the course of the delay.

Local propagation of the memory direction. The local dynamics around recurrent states from the independent evaluation set were characterised by the one-step Jacobian

$$\mathbf{J}_t = \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}}.$$

To quantify whether a perturbation along the task-relevant direction at time $t-1$ is locally preserved along the corresponding task-relevant direction at time t, memory-axis propagation was defined as

$$q(t) = \mathbf{m}_t^\top \mathbf{J}_t \mathbf{m}_{t-1}.$$

This quantity measures the local gain from the memory direction at time $t-1$ to the corresponding direction at time t. A value of $q(t) = 1$ indicates unit local propagation of this component, while $q(t) < 1$ and $q(t) > 1$ indicate local attenuation and amplification, respectively.

Decomposing CB-mediated and remaining local dynamics. In the CB-RNN, the recurrent update depends on the previous recurrent state both directly through the recurrent pathway and indirectly through the CB output. Consequently, the full Jacobian contains a term arising from the dependence of the CB output on the recurrent state. To isolate this term, a second Jacobian, $\mathbf{J}_t^{\text{det}}$, was computed with the CB output held at its value from the unmodified forward pass while its derivative with respect to $\mathbf{h}_{t-1}$ was detached. The forward CB output, and therefore the forward transition itself, was unchanged; only the state-dependent CB contribution to the local Jacobian was removed. Thus,

$$\mathbf{J}_t^{\text{full}} - \mathbf{J}_t^{\text{det}} = \alpha \, \text{diag}[\phi'(\mathbf{a}_t)] \, \frac{\partial \mathbf{b}_{\text{CB},t}}{\partial \mathbf{h}_{t-1}}$$

where $\mathbf{a}_t$ denotes the full recurrent pre-activation, including the CB output.

Propagation was then evaluated separately using the full and detached Jacobians,

$$q_{\text{full}}(t) = \mathbf{m}_t^\top \mathbf{J}_t^{\text{full}} \mathbf{m}_{t-1}, \qquad q_{\text{det}}(t) = \mathbf{m}_t^\top \mathbf{J}_t^{\text{det}} \mathbf{m}_{t-1}.$$

Their difference isolates the signed state-dependent CB contribution to local propagation along the memory axis,

$$q_{\text{CB}}(t) = q_{\text{full}}(t) - q_{\text{det}}(t) = \mathbf{m}_t^\top \left(\mathbf{J}_t^{\text{full}} - \mathbf{J}_t^{\text{det}} \right) \mathbf{m}_{t-1}.$$

Importantly, $\mathbf{J}_t^{\text{det}}$ was evaluated at the same forward operating point as $\mathbf{J}_t^{\text{full}}$, which still includes the CB output. Thus, $q_{\text{CB}}(t)$ isolates the direct state-dependent contribution of the CB pathway to local propagation; it does not measure the difference between the intact network and a network in which CB output is absent.

For each curriculum checkpoint, these quantities were averaged over evaluation trials and across the $N-2$ delay transitions, excluding the comparison step, to obtain

$$Q_{\text{full}}(N) = \langle q_{\text{full}}(t) \rangle_{\text{trials,delay}}, Q_{\text{det}}(N) = \langle q_{\text{det}}(t) \rangle_{\text{trials,delay}}, Q_{\text{CB}}(N) = Q_{\text{full}}(N) - Q_{\text{det}}(N).$$

J.2 Parity

In contrast to DMS, Parity requires the network to continuously update a task-relevant state as new inputs arrive. For sliding-window Parity, the next parity changes according to the current parity π, incoming bit x^{in}, and outgoing bit x^{out}, such that

$$\pi' = \pi \oplus x^{\text{in}} \oplus x^{\text{out}}.$$

The eight possible combinations of $(\pi, x^{\text{in}}, x^{\text{out}})$ define eight transition conditions, with multiple conditions able to lead to the same required next parity. This structure makes it possible to ask whether the CB output contributes to coordinating recurrent transitions between task-relevant states. Specifically, if CB outputs contain information related to the required next state, outputs generated under different transition conditions should produce similar effects when those conditions require the same next parity, but different effects when they require opposite next parities.

Parity transition analysis. Analyses were performed on curriculum checkpoints $N \in \{2, \ldots, 15\}$ from eight trained CB-RNN networks. At each checkpoint, 8 batches of 64 trials were generated using a fixed evaluation seed, and all timesteps for which both current and subsequent sliding-window parity were defined were included. Transitions were pooled across positions within each sequence and assigned to one of the eight $(\pi, x^{\text{in}}, x^{\text{out}})$ conditions.

To quantify where recurrent activity landed following each intervention, a reference geometry was first constructed from unmodified network activity. For each network and checkpoint, transitions were randomly divided into a 70% direction-estimation set and a 30% evaluation set. Within the

direction-estimation set, post-transition recurrent states were grouped according to whether the correct next parity was even or odd and averaged to obtain the corresponding centroids, $\boldsymbol{\mu}_{\text{even}}$ and $\boldsymbol{\mu}_{\text{odd}}$. The direction connecting these centroids defines the next-parity axis:

$$\hat{\mathbf{m}} = \frac{\boldsymbol{\mu}_{\text{even}} - \boldsymbol{\mu}_{\text{odd}}}{\|\boldsymbol{\mu}_{\text{even}} - \boldsymbol{\mu}_{\text{odd}}\|_2}.$$

The midpoint between the centroids was defined as

$$\mathbf{m}_0 = \frac{\boldsymbol{\mu}_{\text{even}} + \boldsymbol{\mu}_{\text{odd}}}{2}.$$

Together, the axis and midpoint provide a coordinate space for investigating where recurrent states lie post-transition, relative to the even and odd representations.

The held out transitions were used for the CB interventions. Following each one-step replay, the resulting recurrent state $\mathbf{h}_{\text{after}}$ was projected onto the next-parity axis relative to its midpoint. The orientation was flipped for transitions requiring odd parity such that positive scores consistently indicated movement towards the correct next-parity representation:

$$s = \sigma \left(\mathbf{h}_{\text{after}} - \mathbf{m}_0\right)^\top \hat{\mathbf{m}}, \qquad \sigma = \begin{cases} +1, & \text{if the correct next parity is even,} \\ -1, & \text{if the correct next parity is odd.} \end{cases}$$

To account for differences in the separation of representations across networks and checkpoints, the score was normalised by half the distance between the centroids:

$$\tilde{s} = \frac{s}{\|\boldsymbol{\mu}_{\text{even}} - \boldsymbol{\mu}_{\text{odd}}\|_2 / 2}$$

to make the scale directly interpretable; a score of $\tilde{s} = +1$ reflects a state at the centroid of the correct next-parity representation, $\tilde{s} = 0$ means that it lies at the midpoint between the two representations, and $\tilde{s} = -1$ indicates a state lying at the centroid of the incorrect next-parity representation.

CB output interventions. Each held-out transition was replayed for a single step from the same pre-transition recurrent state and with the same incoming input, while manipulating only the CB output used in the recurrent update. Four conditions were compared. In *Correct*, the original CB output was retained. In *No CB*, the CB output was set to zero. In *Same*, the original CB output was replaced by CB output from a randomly sampled transition belonging to a different $(\pi, x^{\text{in}}, x^{\text{out}})$ condition that required the same parity. In *Opposite*, the output was instead sampled from a different condition requiring the opposite next parity. Donors were sampled with replacement from the evaluation transitions. Thus, the *Same* and *Opposite* interventions test whether CB outputs generated under different transition conditions produce similar effects on recurrent dynamics according to the next parity associated with the donor condition.

For each network and checkpoint, normalised transition scores were averaged across evaluation transitions. For the aggregate comparison, each network was averaged equally across checkpoints. Scores were additionally examined separately across curriculum levels.

K PERFORMANCE STATISTICS

Descriptive statistics were calculated across repeat runs with random seeds to demonstrate performance differences.

Table 4: **Curriculum-learning performance for model variants under single-task and multitask training.** Values are mean ± SD across $n = 8$ seeds. Epochs to $N_{\text{global}}/2$ is measured against the global maximum N across all models within each setting.

Setting	Model	Max N	Epochs to $N_{\text{global}}/2$	AUC
Single-task DMS	CB-RNN	34.75 ± 0.89	215.38 ± 11.31	$25{,}041 \pm 627$
	Interleaved Reservoir	35.63 ± 2.45	264.38 ± 17.34	$23{,}595 \pm 792$
	Full Reservoir	30.00 ± 1.31	275.00 ± 28.65	$21{,}689 \pm 710$
	RNN-only	31.38 ± 0.52	455.63 ± 28.98	$19{,}254 \pm 524$
Single-task Parity	CB-RNN	18.75 ± 0.46	153.13 ± 14.97	$14{,}170 \pm 232$
	Interleaved Reservoir	16.75 ± 0.71	166.25 ± 10.59	$12{,}996 \pm 571$
	Full Reservoir	15.13 ± 0.35	176.13 ± 8.92	$11{,}971 \pm 245$
	RNN-only	16.75 ± 0.71	312.13 ± 23.96	$11{,}505 \pm 252$
Multitask	CB-RNN	21.88 ± 0.99	168.88 ± 15.15	$16{,}082 \pm 571$
	Interleaved Reservoir	19.25 ± 2.43	218.75 ± 17.24	$14{,}594 \pm 759$
	Full Reservoir	17.75 ± 1.04	235.88 ± 25.85	$13{,}511 \pm 451$
	RNN-only	16.25 ± 1.39	550.25 ± 65.72	$10{,}857 \pm 610$