RESONANCE NET: PERSISTENT STRUCTURED DYNAMICS FOR POST-INPUT AND MULTIMODAL COMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce RESONANCE NET, a recurrent architecture in which persistent complex-valued signals evolve on the edges of a sparse directed graph. Motivated by graph-based signal propagation and filtering, the architecture treats each recurrent tick as a native internal computation step over a persistent structured substrate. Each tick pools incoming state, applies learned complex transformations and edge-selective routing, and updates the substrate, allowing computation to continue after external input ends. We evaluate this property on CLRS-style graph problems by holding the encoded instance fixed while varying the number of post-input recurrent ticks. On average, deeper dependencies favor larger computation budgets: at the deepest evaluated settings, 48 post-input ticks improve mean pointer accuracy across five seeds from 69.5% to 83.9% on BFS and from 69.8% to 84.7% on Bellman–Ford. We next evaluate heterogeneous modality-specific interfaces through a shared core. Across five seeds, increasing the core edge budget from $0.5\times$ to $4\times$ raises joint multimodal utility from 0.862 to 0.869, while fixed, permuted, and shared ingress assignments perform similarly. Eight-seed mechanism ablations reveal task- and seed-dependent effects rather than uniformly necessary components: no individual removal produces a consistent cross-seed degradation in the primary dependency-by-tick surface AUC, while the combined edge-selectivity module increases the BFS benefit obtained from post-input computation across all eight paired seeds. Together, these results show that persistent structured dynamics can support extended internal computation and heterogeneous task interfaces, while also exhibiting substantial variation across learned recurrent solutions.

1 INTRODUCTION

Currently, most neural network architectures couple computational progress to either additional external inputs or additional layers in a fixed feed-forward stack. While recurrent and state-space models relax the latter constraint by maintaining a bounded state, the recurrent state is commonly advanced once per input step (Hochreiter & Schmidhuber, 1997; Gu et al., 2022; Gu & Dao, 2024). We instead ask a different architectural question: Can a compact structured state act as a persistent computational substrate whose internal dynamics remain useful after external input has stopped? A positive answer would separate the amount of internal computation from the amount of evidence presented to the model, while retaining a fixed-size recurrent substrate.

We approach this question with RESONANCE NET, a sparse directed graph with persistent state that lives on complex-valued edges. The design is motivated conceptually by graph-based signal propagation and filtering where encoded observations inject signals into designated input nodes, connectivity determines where those signals can propagate, and learned edge dynamics determine how they are mixed, phase-shifted, retained, and rewritten over repeated ticks as they propagate through the graph. In this view, a recurrent tick is not only an input-time update but also the model's native unit of time and internal computation over a persistent substrate. Unlike a conventional static graph filter, the signal being transformed is itself the persistent nonlinear state, so once the external drive is removed the graph can still evolve without receiving another observation input.

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.

Based on the motivation, the evaluation therefore centers on post-input computation: an operational test of whether the model's internal tick mechanism performs useful work after the available evidence is fixed. On controlled CLRS-style graph problems, we hold the encoded instance fixed and vary only the number of recurrent ticks that follow input termination. On average, the preferred compute budget moves from zero or small values on shallow dependencies toward substantially larger budgets on deeper instances. At the deepest evaluated settings, 48 post-input ticks raise mean pointer accuracy across five seeds from 69.5% to 83.9% on BFS and from 69.8% to 84.7% on Bellman–Ford.

We then ask whether the same structured substrate remains usable behind heterogeneous interfaces. In multimodal experiments, modality-specific peripherals for CMU-MOSEI and Vision & Touch communicate through the same six-port core. Joint training is viable, and increasing the shared core edge budget improves the shared core joint utility. Stronger transfer and no-core controls delimit this result: a universally reusable frozen interlingua is not established, but they do not change the positive shared-substrate compatibility and capacity result.

Additional studies characterize rather than define the major claim. Across eight paired CLRS seeds, mechanism ablations show that phase dynamics, geometry, edge-selective routing, and structural plasticity are not uniformly additive across tasks and learned regimes. A more specific result is that the combined edge-selectivity module consistently increases the BFS benefit of post-input computation. We interpret this as a complementary mechanism repertoire rather than as evidence of an explicit mechanism-switching controller. Long-context BABILong experiments are retained as a stress test of bounded persistence.

Contributions. Our contributions are threefold. First, we introduce a graph-signal-inspired recurrent substrate with persistent complex edge state, where repeated ticks provide a native internal computation mechanism whose budget can be extended after input ends. Second, we construct a controlled dependency-depth experiment that isolates additional computation from additional evidence and show substantial post-input gains on deep BFS and Bellman–Ford instances. Third, we demonstrate that heterogeneous modality-specific peripherals can operate through one common fixed-port core and characterize how joint multimodal performance changes with core edge budget, while multi-seed ablations show task- and seed-dependent rather than uniformly additive effects of the available mechanisms.

2 Related Work

Recurrent and state-space dynamics. LSTM introduced gated recurrent state for long dependencies (Hochreiter & Schmidhuber, 1997). HiPPO, S4, Mamba/Mamba-2, and xLSTM develop increasingly capable bounded-state sequence models (Gu et al., 2020; 2022; Gu & Dao, 2024; Dao & Gu, 2024; Beck et al., 2024). Transformer-XL and RMT instead extend attention through segment recurrence or persistent memory tokens (Dai et al., 2019; Bulatov et al., 2022). Resonance Net shares the use of a bounded recurrent state but makes that state a sparse interaction substrate with explicit internal transitions that do not need to be tied one-to-one to new input steps.

Internal computation budgets. Adaptive Computation Time and Universal Transformers already decouple the internal processing depth from input length (Graves, 2016; Dehghani et al., 2019). Our approach is the persistent complex edge-state substrate and its controlled depth-by-tick evaluation; we prescribe the tick budget rather than learn halting.

Graph filtering and structured dynamics. Graph signal processing generalizes filtering and frequency-domain reasoning to signals defined on irregular graphs (Shuman et al., 2013), while graph attention learns content-dependent local aggregation (Veličković et al., 2018). This provides useful context for the intuition behind Resonance Net: each recurrent tick performs a learned local graph transformation, but unlike a conventional graph filter the state is complex-valued, stored on edges, retained between ticks, and updated by a nonlinear oscillator-like rule. Neural ODEs similarly separate computation time from a fixed stack of layers (Chen et al., 2018), and complex-valued networks provide trainable amplitude- and phase-sensitive transformations (Trabelsi et al., 2018). Resonance Net combines these ingredients in a discrete recurrent substrate whose internal tick budget can vary independently of input length.

Multimodal shared representations. Multimodal systems commonly place modality-specific encoders around a shared representation and task-specific decoders (Liang et al., 2021). Our current question is narrower than building a multimodal foundation model. We test whether heterogeneous streams can coexist through a fixed-port instance of the same persistent recurrent substrate, and whether changing the shared core edge budget measurably changes joint performance. We separately test frozen transfer so that shared-core compatibility is not conflated with a claim of universal modality-independent representation learning.

3 RESONANCE NET

3.1 PERSISTENT STATE ON A SPARSE DIRECTED SUBSTRATE

Let $G = (V, E)$ be a sparse directed graph with input, hidden, and output nodes, $V = V_{\text{in}} \cup V_{\text{hid}} \cup V_{\text{out}}$. For a directed edge $e = (u \to v)$, the recurrent state is $z_e^{(t)} \in \mathbb{C}^S$. The full persistent state is therefore the edge tensor $Z^{(t)} = \{z_e^{(t)}\}_{e \in E}$; nodes are transient pooling and routing sites rather than a second long-lived recurrent state. Passing this edge tensor as the initial state of a later call carries information across input segments without an explicit cache of previous observations. Figure 1 summarizes the end-to-end interface and the separation between input-driven and post-input recurrent computation.

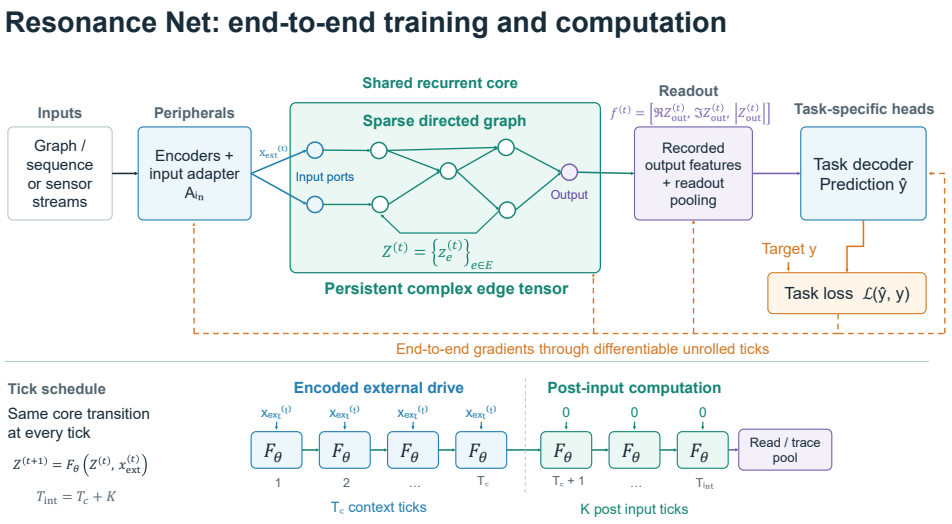

Figure 1: Overview of RESONANCE NET. Modality- or task-specific peripherals drive designated input ports of a shared sparse recurrent core whose persistent state lives on directed complex-valued edges. The same core transition is unrolled end-to-end during training. After T_c input-driven context ticks, the external drive is set to zero and the substrate may continue for K post-input recurrent ticks before readout, giving $T_{\text{int}} = T_c + K$.

This structure admits a useful connection to graph signal filtering. Classical graph filters transform signals according to graph connectivity and edge/vertex operators (Shuman et al., 2013). A RESONANCE NET tick can be read as a recurrent, state-rich generalization of that idea: local graph aggregation produces a node signal, edge-specific complex coefficients transform it, and the result is written back into a persistent edge signal that participates in subsequent filtering steps. The analogy is only conceptual though: the update below is nonlinear, state-rich, and can operate on a graph whose connectivity is itself adaptable.

Each node has a learned coordinate $p_v \in \mathbb{R}^{d_c}$, bounded by a $\tanh$ parameterization. For $e = (u \to v)$, the dimension-normalized geometric length is $\ell_e = \|p_v - p_u\|_2 / \sqrt{d_c}$. Geometry affects both the

phase of retained state and the attenuation of newly written signal, so coordinates participate directly in the dynamics rather than serving only as an embedding for visualization.

3.2 One core tick: pool, route, retain, and write

Destination-wise pooling. Let $x^{(t)}_{\mathrm{ext},v}$ denote the complex external drive supplied to an input node at tick t. Each edge has a trainable incoming-composition logit b_e. Weights are normalized separately for every destination, and the transient node signal is

$$\alpha_e = \frac{\exp b_e}{\sum_{e':\,\mathrm{dst}(e')=v} \exp b_{e'}}, \quad s^{(t)}_v = \sum_{e:\,\mathrm{dst}(e)=v} \alpha_e z^{(t)}_e + \mathbf{1}[v \in V_{\mathrm{in}}] x^{(t)}_{\mathrm{ext},v}. \tag{1}$$

Incoming edges therefore compete only with other edges entering the same node; the pooled node value is transient and is not an additional recurrent state.

Edge-specific complex routing. Every outgoing edge has a learned complex routing vector $q_e \in \mathbb{C}^S$. Routing is component-wise complex modulation followed by a complex ReLU,

$$\widetilde{s}^{(t)}_e = \mathrm{CReLU}\left(s^{(t)}_u \odot \overline{q}_e\right), \qquad \mathrm{CReLU}(a+ib) = \mathrm{ReLU}(a) + i\,\mathrm{ReLU}(b). \tag{2}$$

Edges sourced directly from input nodes bypass both modulation and activation and transmit the unmodified source signal. The routing vector can therefore selectively modulate complex signal components before they are written onto each outgoing edge.

Geometric oscillator update. For each edge, learned raw parameters determine phase increment ψ_e, retention γ_e, positive drive η_e, and positive amplitude ceiling ρ_e:

$$\begin{aligned} \psi_e &= \omega_e \Delta t + \phi_e + \beta \ell_e, & \gamma_e &= \gamma_{\min} + (\gamma_{\max} - \gamma_{\min})\sigma(d_e), \\ \eta_e &= \mathrm{softplus}(h_e), & \rho_e &= \mathrm{softplus}(r_e) + 10^{-6}. \end{aligned} \tag{3}$$

The pre-saturation state update is

$$\widehat{z}^{(t+1)}_e = \underbrace{\gamma_e e^{i\psi_e} z^{(t)}_e}_{\text{retain and rotate}} + \underbrace{\eta_e e^{-\ell_e/\lambda} \widetilde{s}^{(t)}_e}_{\text{write routed signal}}. \tag{4}$$

The first term preserves existing edge state while damping and phase-rotating it; the second injects newly routed information, with longer learned geometric distance reducing write strength. A radial limiter then bounds each complex component,

$$z^{(t+1)}_{e,j} = \widehat{z}^{(t+1)}_{e,j} \min\left(1, \frac{\rho_e}{\max(|\widehat{z}^{(t+1)}_{e,j}|, 10^{-8})}\right). \tag{5}$$

Because the rescaling is radial in the complex plane, magnitude is bounded without separately clipping real and imaginary parts. Equations 1–5 make the core tick a repeated *pool* $\rightarrow$ *route* $\rightarrow$ *retain/rotate* $\rightarrow$ *write* operation, with persistence located on directed edges.

3.3 External input ticks and input-free computation

Suppose an example supplies T_c encoded packets. External drive is present while those packets are consumed and zero afterward,

$$x^{(t)}_{\mathrm{ext}} = \begin{cases} A_{\mathrm{in}}(x_t), & t \le T_c, \\ 0, & t > T_c, \end{cases} \tag{6}$$

where A_{in} is the experiment-facing input adapter. The same recurrent operator may then continue for a total budget of K *post-input recurrent ticks*, separating exposure to evidence from amount of computation. In the algorithmic experiments, varying K therefore changes post-input recurrent processing while leaving the encoded problem instance fixed. Experiment adapters may internally decompose this total post-input budget into an initial propagation ("flush") portion and optional extra ticks. The CLRS computation sweep sets the separate flush to zero, so its reported K is exactly the total post-input budget shown in Figure 1.

The implementation used in all reported experiments is tick-synchronous: node signals are composed from the state at the start of a tick, after which the active edges update together from that shared snapshot. This preserves a clear interpretation of a tick as one recurrent transition of the persistent substrate.

3.4 OUTPUT BRANCHES AND TASK READOUT

At recorded ticks, output-node complex states are converted to real-valued features by concatenating real part, imaginary part, and magnitude,

$$f^{(t)} = [\Re Z_{\text{out}}^{(t)},\ \Im Z_{\text{out}}^{(t)},\ |Z_{\text{out}}^{(t)}|]. \tag{7}$$

Experiment-owned readouts pool these features and task-specific heads decode the requested target. This boundary keeps modality and task semantics primarily in peripherals while the central core operates on a common signal format. In the multimodal experiments, a learned query aggregates the small set of output branches per recorded tick and a second single-query temporal attention readout summarizes the resulting within-example internal trace. This readout is not a persistent history bank; recurrent state across calls remains the carried edge tensor.

3.5 TOPOLOGY AND STRUCTURAL PLASTICITY

RESONANCE NET uses a sparse *maximum-depth-budget* topology. Nominal graph depth is a maximum forward-path budget rather than a requirement that every route traverse all strata: skip edges create shorter propagation paths while a topology guard ensures that at least one route realizes the full budget. In our experiments the maximum forward depth is 12, skip probability decays with factor 0.35, and a repair rule restores a full-depth route if structural adaptation removes all such paths. Recurrent and lateral edges are outside the acyclic forward-depth budget and provide additional circulation paths.

Structural plasticity is optional and is activated only in experiments that test it or benefit from it. Activity statistics identify underused eligible capacity, selected edge slots are recycled, and new endpoints are chosen subject to the same topological constraints. Reinitialized slots also receive fresh oscillator and routing parameters. Because plasticity can alter both connectivity and local edge dynamics, its contribution is evaluated explicitly rather than inferred from late-training behavior. In the paper-scale CLRS configuration, automatic plasticity begins after 5,000 optimizer steps, is eligible every 1,000 steps, recycles two nodes per event, and selects activity using the top 24 active edges. The automatic trigger uses a 1,000-step patience window with a minimum validation improvement of 10^{-4} before another structural event is permitted.

3.6 MULTIMODAL PERIPHERALS

Heterogeneous observations are handled with modality-specific peripherals around an unchanged core. For modality m,

$$x_t^{(m)} \xrightarrow{E_m} p_t^{(m)} \xrightarrow{\text{fixed port } \pi(m)} \text{RESONANCE NET}, \tag{8}$$

where E_m is a local encoder and $\pi(m)$ is a stable physical input-port assignment. The reported multimodal configuration uses fixed ports rather than a learned modality router, preserving source identity at the interface while requiring cross-modal interaction to occur through the shared substrate. Lightweight, local, and stronger diagnostic encoder families are screened before choosing one setting per modality.

We do not interpret this interface as evidence of a universally reusable latent interlingua. Frozen-core and no-core controls show that substantial isolated-task performance can be carried by selected peripherals and readout. The supported interpretation is narrower: heterogeneous streams can coexist through the same persistent core, and joint performance exhibits measurable dependence on central graph capacity.

3.7 CONTROLLED AND ARCHITECTURE-NATIVE BASELINES

Where the scientific question is the temporal substrate itself, recurrent, attention-based, and state-space baselines share experiment encoders, task heads, and data processing, with hidden widths chosen to approximately match trainable parameter counts. Where the question concerns whether an architecture is being used through an appropriate interface, we additionally permit compact architecture-native peripherals and report those comparisons separately. This two-track design avoids conflating a poor interface match with the behavior of the central recurrent architecture.

4 EVALUATION

We organize evaluation around properties of the recurrent substrate. Architecture and protocol choices are fixed within each reported campaign. Key optimization, seed, and parameter-count settings are summarized in Appendix B; the supplementary code and data packages retain the complete run configurations and seed-level outputs.

Dependency-sensitive post-input computation. The primary study uses custom BFS and Bellman–Ford probes inspired by CLRS (Veličković et al., 2022), with one queried parent per layered graph and candidates restricted to the query's neighbors plus itself. These are not official CLRS benchmark scores. A fixed encoded instance is followed by post-input ticks $K \in \{0, 1, 2, 4, 6, 8, 12, 16, 24, 32, 48, 64\}$. Dependency depth is varied independently of the post-input budget: BFS uses depths $\{1, 2, 3, 4, 6, 8, 12, 16, 24, 32\}$ and Bellman–Ford uses $\{1, 2, 3, 4, 6, 8, 12, 16\}$. We report five independently trained seeds of the main RESONANCE NET configuration, with 128 fixed probes per depth and seed. Controlled adapter-only, GRU (Cho et al., 2014), and message-passing (Gilmer et al., 2017) baselines separate extra recurrent computation from task-adapter effects. The major statistic in this experiment is the interaction between dependency depth and effective post-input compute.

Heterogeneous multimodal compatibility. The multimodal study combines CMU-MOSEI (Bagher Zadeh et al., 2018) and Vision & Touch (Lee et al., 2020) using MultiBench-derived loaders. Final metrics use the MOSEI test split and the V&T validation split, which also supports model selection. Joint utility averages clipped $1 - \mathrm{MAE}/3$ and V&T balanced accuracy. Modality-specific encoders map six streams to stable physical ingress ports around one common core. We compare approximately parameter-matched RESONANCE NET, Transformer (Vaswani et al., 2017), GRU, and RSSM-style (Hafner et al., 2019) variants across five seeds. Resonance uses a graph-depth (12-tick) transit flush plus two explicit extra ticks, whereas the baseline wrappers use two post-input slots/steps; these cross-architecture controls are therefore not matched on raw transition count. We also evaluate isolated and joint training, vary the shared core edge budget over $0.5\times$, $1\times$, $2\times$, and $4\times$, and perturb ingress assignments. A final five-seed control suite compares source-trained frozen cores with core-randomized and fully random controls, and includes a peripheral-only no-core bypass. The focused question here is shared-core joint compatibility and capacity behavior.

Mechanism ablations. We remove phase rotation, geometric effects, the combined edge-selectivity module, or structural plasticity under the same CLRS training protocol. The selectivity ablation also bypasses CReLU; Appendix E specifies each intervention. The expanded study uses eight paired training seeds and the same 30k-step horizon. Each trained model is evaluated on three common fixed-probe replicates with 128 examples per dependency depth and the full post-input tick grid. Probe replicates are averaged within training seed; all confidence intervals and paired randomization statistics use training seed as the replication unit. The primary statistic is normalized dependency-by-tick surface AUC, supplemented by targeted measurements of how much post-input recurrence improves over the $K = 0$ condition.

Long-context persistence stress test. BABILong (Kuratov et al., 2024) QA1–QA5 evaluates whether a bounded carried state remains useful as the evidence–query gap grows beyond training lengths. RESONANCE NET carries recurrent state across segments while truncating gradients rather than resetting the forward state. The completed study is retained because it exposes an important boundary of the present architecture; aggregate long-context tables and the causal state intervention summary are shown in the appendix; complete per-task and per-seed tables are retained in the supplementary data.

5 RESULTS

5.1 DEEPER DEPENDENCIES BENEFIT FROM POST-INPUT COMPUTATION

The aggregate curves show a depth-by-compute pattern. For the main RESONANCE NET model, the preferred post-input budget moves toward larger values as dependency depth grows. Bellman–Ford

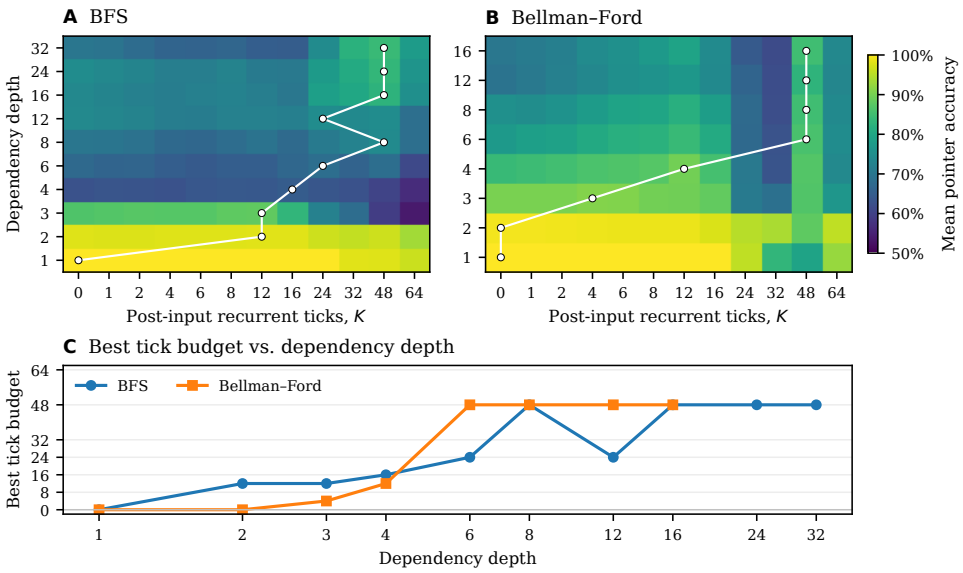

Figure 2: Dependency-sensitive post-input computation on BFS and Bellman–Ford. Panels A–B show mean pointer accuracy over the dependency-depth by post-input-tick sweep; the white trace marks the aggregate best tick budget at each depth. Panel C summarizes how the preferred budget shifts as dependency depth increases. The encoded instance is fixed throughout each K sweep.

is particularly clear: the aggregate best budget is 0 ticks at depths 1–2, 4 ticks at depth 3, 12 ticks at depth 4, and 48 ticks from depth 6 through the deepest evaluated depth 16. BFS is less monotone but shows the same overall frontier, reaching a 48-tick optimum at depths 8, 16, 24, and 32.

Figure 2 shows the full depth-by-compute interaction; at the deepest conditions, the numerical effect is especially clear. On BFS, 48 post-input ticks improve mean pointer accuracy from 69.5% to 83.9%; on Bellman–Ford the corresponding change is 69.8% to 84.7%. Because the encoded problem instance is held fixed during this sweep, the improvement cannot be attributed to seeing more input. It is evidence that recurrent evolution of the persistent substrate can perform useful additional computation after input ends. The effect is not unique to RESONANCE NET: a matched MPNN also gains at the deepest Bellman–Ford condition (Appendix Table A4), so we do not claim exclusive ownership of test-time compute scaling.

The mean curves also reveal a stability problem that is important for interpreting this result. At the deepest BFS condition, four of five seeds improve between $K = 0$ and $K = 48$, but one drops from 85.9% to 30.5%. Bellman–Ford likewise has one seed that drops, from 46.9% to 27.3%, while the remaining four improve. The paired five-seed 95% Student-t intervals for the deepest-condition gains include zero (BFS: $[-45.8, 74.6]$ pp; Bellman–Ford: $[-21.3, 51.0]$ pp). The mean gains are therefore descriptive, and extra recurrence is not uniformly safe. The frontier selects its best budget on the displayed evaluation sweep.

5.2 A COMMON CORE SUPPORTS HETEROGENEOUS MULTIMODAL INTERFACES

The multimodal experiments give a complementary breadth result. A single six-port RESONANCE NET core can be jointly trained with modality-specific peripherals across CMU-MOSEI and Vision & Touch, despite the different sensor structures and task heads. Figure 3 reports the completed five-seed capacity sweep. Increasing the shared core edge budget from $0.5\times$ to $4\times$ raises Resonance joint utility monotonically from 0.862 to 0.869. Most of the change comes from MOSEI, whose MAE improves from 0.649 to 0.613, while Vision & Touch remains near its already high performance band.

At low capacity, Resonance has a statistically detectable joint-utility gap to the matched Transformer and RSSM controls; by $4\times$, the paired confidence intervals against Transformer, GRU, and RSSM all include zero. Fixed, permuted, and shared ingress assignments are also all viable (Appendix Fig-

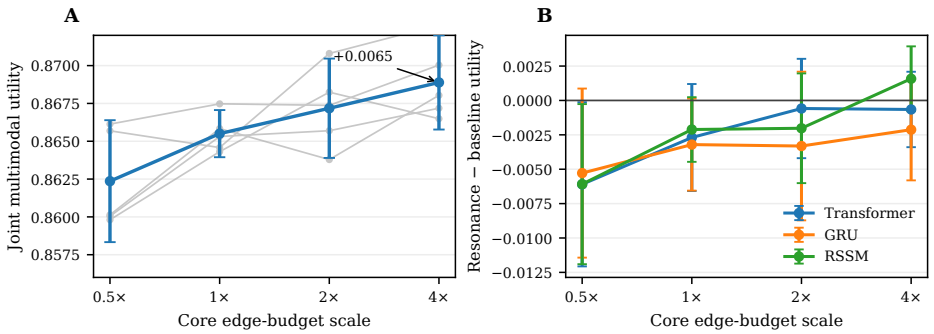

Figure 3: Multimodal capacity scaling across five seeds. Panel A shows Resonance joint multimodal utility as the shared core edge budget increases from $0.5\times$ to $4\times$; faint lines show individual seeds and the emphasized line shows the aggregate trend. Panel B reports paired Resonance-minus-baseline joint-utility gaps against approximately parameter-matched Transformer, GRU, and RSSM controls; raw internal transition counts are architecture-specific.

Table 1: Eight-seed mechanism ablation. Entries are paired full-minus-ablation changes in normalized dependency-by-tick surface AUC, with 95% Student-t confidence intervals over training seeds. Positive values favor the full model.

Removed mechanism	BFS Δ surface AUC	Bellman–Ford Δ surface AUC
Phase rotation	-0.001 $[-0.099, 0.097]$	-0.048 $[-0.256, 0.159]$
Learned geometry	$+0.007$ $[-0.092, 0.106]$	-0.132 $[-0.332, 0.068]$
Selectivity + CReLU	$+0.044$ $[-0.009, 0.097]$	-0.070 $[-0.206, 0.066]$
Structural plasticity	-0.016 $[-0.080, 0.048]$	$+0.014$ $[-0.192, 0.221]$

ure A2), indicating that the result does not depend on learning a special modality router or on one privileged port assignment.

The final transfer controls narrow, rather than reverse, this interpretation. Source-trained frozen Resonance cores do not reliably outperform randomized-core controls in either transfer direction (Appendix Figure A3). In isolated evaluation, a peripheral-only bypass also lies in the same performance band as the core models: MOSEI MAE is 0.611 ± 0.015 (95% CI half-width) for the bypass, while Vision & Touch balanced accuracy is 0.944 ± 0.003. These controls rule out a strong claim that the current frozen core has already learned a universally reusable multimodal interlingua. The supported result is instead that heterogeneous task interfaces can be trained through one shared recurrent substrate and that the quality of this joint setting responds to the shared core edge budget.

5.3 Mechanism utility is task- and regime-dependent

We next ask whether the core's phase dynamics, learned geometry, edge-selective routing, and structural plasticity behave as uniformly necessary modules. The expanded ablation study retrains the full model and each one-mechanism removal over eight paired seeds, then evaluates three common fixed-probe replicates per seed. Table 1 reports the designated primary metric (the corresponding effect-size visualization is Appendix Figure A4): the normalized dependency-by-tick surface AUC, expressed as paired full-minus-ablation differences. None of the four removals produces a seed-stable primary degradation on both BFS and Bellman–Ford; all Student-t confidence intervals include zero. This revises the much stronger effects visible in the original seed-0 screen and is compatible with compensation across learned solutions, but does not establish equivalence or identify a compensation mechanism.

The effects are nevertheless structured rather than uniformly null. On BFS, retaining the combined edge-selectivity module increases the mean post-input improvement relative to $K = 0$ by 0.106, with a 95% Student-t interval of $[0.038, 0.173]$ and the paired effect positive in all eight seeds (exact two-sided sign-flip $p = 0.0078$, unadjusted secondary comparison). The corresponding primary surface-AUC effect is smaller and less conclusive ($+0.044$, 6/8 positive seeds). We interpret this

pattern as consistent with a *complementary repertoire*: different tasks and optimization trajectories can rely on different subsets of phase, geometry, routing, and plasticity rather than requiring every mechanism to contribute additively on every task. This remains an interpretation rather than a demonstrated switching or compensation mechanism.

The accompanying reduced BABILong ablation is not used as mechanism evidence: its full carried-state condition underperformed a question-only intervention and did not exercise structural plasticity. The CLRS setting is the controlled mechanism study used for the conclusions above.

5.4 Bounded persistence does not by itself guarantee long-context retention

The completed BABILong campaign provides a useful negative stress test. Performance of the present bounded-state system degrades substantially as context length grows, and dedicated recurrent-memory baselines remain stronger at long contexts. Causal state interventions also do not support turning this experiment into a strong long-term-memory claim. We therefore use BABILong to characterize a limitation of the current substrate rather than as a headline result. Aggregate per-length and intervention tables are reported in Appendix F, while the supplementary data retain the complete per-task and per-seed results. The aggregate stress-test curves are shown in Appendix Figure A5.

6 Discussion and Limitations

Persistent complex edge state can support post-input computation and heterogeneous task interfaces. On the controlled graph probes, deeper dependencies tend to favor larger tick budgets in the aggregate, but the five-seed endpoint-gain intervals include zero. The matched MPNN also benefits from additional computation. To conclude, these experiments support the feasibility and a descriptive depth/compute pattern, rather than a general superiority or a reliable stopping claim.

The multimodal capacity trend is positive across the evaluated seeds. However, V&T is evaluated on validation data also used for model selection, and the cross-architecture controls are parameter-matched rather than compute-matched. Frozen-transfer and bypass results further show that strong peripherals account for much of isolated-task performance. Shared-core compatibility therefore does not establish a universally reusable interlingua.

No primary ablation interval excludes zero. Task- and seed-dependent effects are consistent with a complementary mechanism repertoire, but do not demonstrate compensation, equivalence, or learned switching. The targeted BFS result concerns a combined pooling, routing, and activation intervention and an unadjusted secondary statistic. More selective interventions and independent confirmation would clarify the mechanism.

Finally, long-context retention remains weak despite persistent state. Future work should test stability-aware stopping, improved memory protection, and adaptable capacity on independently held-out evaluations. The current results motivate those tests without establishing their effectiveness.

AI USE STATEMENT

In this work, we used generative AI tools to assist with experimental design, implementation and debugging of research code, literature search and summarization, organization of experimental results, and drafting of manuscript text. The authors determined the research questions, architecture, experimental protocols, and claims; executed the reported experiments; inspected the resulting artifacts and aggregate measurements; and verified quantitative statements against the saved experimental outputs. AI-assisted code was reviewed and executed before its results were used, and AI-assisted summaries were checked against the underlying experimental results and sources cited. No reported experimental measurements were generated by a large language model. We take responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Key architecture, optimization, evaluation, parameter-matching, and seed settings are summarized in the main paper and appendix. The supplementary code and data packages contain the frozen experiment scripts, configuration manifests, per-run measurements, and aggregate summaries used to audit the reported numbers; trained model weights and source benchmark datasets are not included. Experiments are implemented as resume-safe pipelines, and the principal quantitative claims in this manuscript can be traced from aggregate tables back to seed-level outputs.

REFERENCES

AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2236–2246, 2018. doi: 10.18653/v1/P18-1208.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended long short-term memory. In *Advances in Neural Information Processing Systems*, volume 37, pp. 107547–107603, 2024. doi: 10.52202/079017-3417.

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11079–11091, 2022. doi: 10.52202/068431-0805.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

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

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, 2019. doi: 10.18653/v1/P19-1285.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1807.03819.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272, 2017. URL https://proceedings.mlr.press/v70/gilmer17a.html.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016. URL https://arxiv.org/abs/1603.08983.

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

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/102f0bb6efb3a6128a3c750dd16729be-Abstract.html.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

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

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. In *Advances in Neural Information Processing Systems*, volume 37, pp. 106519–106554, 2024. doi: 10.52202/079017-3381.

Michelle A. Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 36(3):582–596, 2020. doi: 10.1109/TRO.2019.2959445.

Paul Pu Liang, Yiwei Lyu, Xiang Fan, Zetian Wu, Yun Cheng, Jason Wu, Leslie (Yufan) Chen, Peter Wu, Michelle A. Lee, Yuke Zhu, Ruslan Salakhutdinov, and Louis-Philippe Morency. MultiBench: Multiscale benchmarks for multimodal representation learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/37693cfc748049e45d87b8c7d8b9aacd-Abstract-round1.html.

David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013. doi: 10.1109/MSP.2012.2235192.

Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J. Pal. Deep complex networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1T2hmZAb.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The CLRS algorithmic reasoning benchmark. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22084–22102, 2022. URL https://proceedings.mlr.press/v162/velickovic22a.html.

A Architecture and Implementation Details

The reported core uses complex state on sparse directed edges, a maximum forward path budget of 12, dimension-normalized learned geometry, edge-selective incoming composition and outgoing routing, and optional automatic structural plasticity. In the paper-scale CLRS setting, auto-plasticity begins after 5,000 optimizer steps, is eligible every 1,000 steps, recycles two nodes per event, and uses the top 24 active edges in its activity statistic. The supplementary code and data packages contain the frozen configuration files, run manifests, and seed-level outputs used for the audits below.

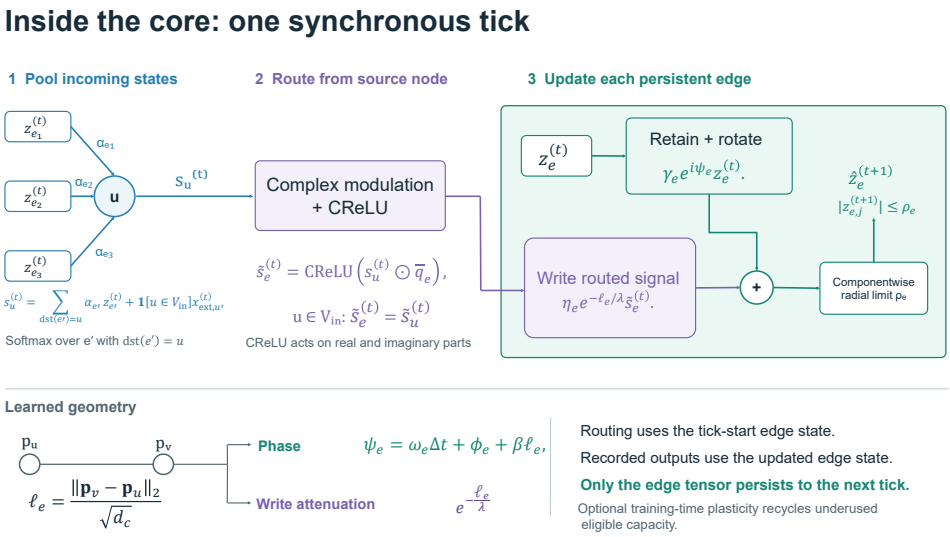

Figure A1: One synchronous Resonance Net tick. Incoming persistent edge states are pooled at each destination, the transient source-node signal is routed by edge-specific complex modulation, and every persistent edge combines retained, phase-rotated state with the routed write before componentwise radial limiting. Pooling and routing use the shared tick-start state; only the edge tensor persists to the next tick.

B Reproducibility Summary

Table A1 records the frozen campaign-level settings used for the principal results. The supplementary data retain the full per-run configurations, including seed-specific realized graph sizes, selected learning rates, checkpoint selection metadata, and evaluation rows; the supplementary code contains the scripts that generated and aggregated those records. Trained weight files and original benchmark datasets are not part of the supplied packages; evaluation-from-checkpoint requires regenerating or separately obtaining those artifacts.

Table A1: Frozen training and evaluation settings for the principal campaigns.

Campaign	Training seeds	Training budget	Optimizer and principal evaluation
CLRS post-input	0–4	30k steps, batch 128	AdamW, 3×10^{-4}, weight decay 10^{-2}, 1k-step warmup and cosine decay to $0.1\times$; train-time compute choices $\{2,4,6,8\}$; evaluate $K \in \{0,1,2,4,6,8,12,16,24,32,48,64\}$.
Multimodal	0–4	6k MOSEI / 12k V&T updates	AdamW with 5% warmup and cosine decay to $0.1\times$; LR selected from $\{10^{-4}, 3\times10^{-4}, 10^{-3}\}$; baselines approximately match seed-0 trainable parameters. Resonance uses a 12-tick graph-depth flush plus two extra ticks; baseline wrappers use two post-input slots/steps.
Mechanism ablation	0–7	30k steps	Same CLRS training protocol; three shared fixed-probe replicates of 128 examples per dependency depth are averaged within seed.
BABILong stress test	0–2	55k steps, batch 128	AdamW, 3×10^{-4}, weight decay 10^{-2} and 10% warmup; train only through 16k context, evaluate through 128k plus a 100-example-per-task tail at 256k–1M.

For the main CLRS configuration at seed 0, RESONANCE NET has 998,167 trainable parameters and 509 realized core edges. The matched MPNN has 998,231 trainable parameters, the GRU 1,011,801, and the adapter-only control 978,689. The persistent RESONANCE NET edge tensor occupies 65,152 bytes for batch size one in this configuration. In the $1\times$ multimodal capacity run, the five realized Resonance cores contain 472–529 edges (mean 506.6); total trainable parameters are approximately 1.02M and are used as the target for the matched baseline widths. These multimodal baselines are parameter-matched rather than raw-transition-count matched; the comparison therefore serves as a task-performance control, not a compute-efficiency comparison.

C DEPENDENCY-SENSITIVE COMPUTATION PROTOCOL AND ADDITIONAL RESULTS

The primary computation experiment trains five seeds for BFS and Bellman–Ford. Each model is evaluated with post-input ticks $K \in \{0,1,2,4,6,8,12,16,24,32,48,64\}$ while the encoded instance is held fixed. BFS probes dependency depths $\{1,2,3,4,6,8,12,16,24,32\}$; Bellman–Ford probes $\{1,2,3,4,6,8,12,16\}$; each depth uses 128 fixed examples per seed. The supplementary data contain the full per-seed depth-by-tick rows and the separate official-CLRS calibration outputs. Tables A2–A4 summarize the main deepest-condition results and controlled baselines.

Training uses 16-node graphs; the depth probes use 64-node, undirected layered graphs with one designated query per example. Bellman–Ford backbone edges are cheap while added distractor edges are expensive. The query-conditioned adapter pools graph facts into 16 packets, and the decoder uses observable graph features and a neighbors-plus-self candidate mask. All controlled models share this mask; MPNN uses $1+K$ message-passing passes. Accuracy measures the selected parent, not exact recovery of the full predecessor tree. The budget frontier is selected from the evaluation grid and is descriptive, not a learned or held-out stopping rule. The paired $K=48$ minus $K=0$ confidence intervals use five training-seed differences, rather than treating individual queries as independent model replications.

Table A2: Post-input computation at the deepest evaluated dependency for each primary task. Accuracy is pooled over five trained seeds and 128 probes per seed (640 queries per cell). These are repeated evaluations of fixed probes, not 640 independent training replicates. No new evidence is supplied between $K=0$ and $K=48$.

Task	Depth	$K=0$	$K=48$	Change
BFS	32	69.5%	83.9%	+14.4 pp
Bellman–Ford	16	69.8%	84.7%	+14.8 pp

Table A3: Seed-level Resonance accuracy at the deepest condition. The aggregate gain is not uniformly stable: BFS seed 4 and Bellman–Ford seed 3 deteriorate at $K = 48$.

Task	Seed	$K = 0$	$K = 48$	Δ	Task	Seed	$K = 0$	$K = 48$	Δ
BFS	0	93.0	93.8	+0.8	BF	0	94.5	99.2	+4.7
BFS	1	36.7	97.7	+60.9	BF	1	38.3	97.7	+59.4
BFS	2	93.8	99.2	+5.5	BF	2	92.2	99.2	+7.0
BFS	3	38.3	98.4	+60.2	BF	3	46.9	27.3	-19.5
BFS	4	85.9	30.5	-55.5	BF	4	77.3	100.0	+22.7

Table A4: Controlled deepest-condition comparison, pooled over five seeds (percent pointer accuracy). The adapter-only control is insensitive to K by construction. The Bellman–Ford MPNN also benefits strongly from additional message-passing steps, so the post-input-compute effect is not claimed to be unique to Resonance.

	BFS, depth 32			Bellman–Ford, depth 16		
Model	$K = 0$	$K = 48$	Δ	$K = 0$	$K = 48$	Δ
Adapter only	58.0	58.0	0.0	67.8	67.8	0.0
GRU	77.7	67.2	-10.5	38.3	37.5	-0.8
MPNN	73.0	68.4	-4.5	61.4	94.1	+32.7
Resonance	69.5	83.9	+14.4	69.8	84.7	+14.8

D Multimodal Compatibility Protocol and Additional Results

CMU-MOSEI and Vision & Touch are trained with modality-specific encoders and task heads around a common six-port temporal core using MultiBench-derived loaders (Liang et al., 2021). The main benchmark and extension studies use five seeds where reported, compare Resonance Net with Transformer, GRU, and RSSM cores, and include isolated versus joint training, $0.5\times$–$4\times$ capacity scaling, ingress assignment controls, sequential training, and directional transfer. The final transfer-control campaign retrains fresh source models and compares source-trained frozen cores, randomized cores, fully random core/readout controls, and a no-core peripheral bypass. The appendix reports aggregate capacity, ingress, bypass, and transfer controls; complete per-seed rows are retained in the supplementary data. The main text uses the capacity curve as the positive result and transfer as a boundary control.

The reported joint score is $U = \frac{1}{2}[\mathrm{clip}(1 - \mathrm{MAE}_{\mathrm{MOSEI}}/3, 0, 1) + \mathrm{clip}(\mathrm{BA}_{\mathrm{V\&T}}, 0, 1)]$. MOSEI uses its test split; V&T has a train/validation split and its validation partition also informs learning-rate and checkpoint selection. V&T scores and joint utility are therefore not wholly independent held-out test estimates. The V&T loader deduplicates validation trajectories, uses the complementary training files, and drops the final incomplete batch. Final results use the validation-selected checkpoint, after a training budget of 6k MOSEI and 12k V&T updates. Across five paired seeds, the $4\times$ minus $0.5\times$ joint-utility change is 0.00652, with a 95% Student-t interval of $[0.00462, 0.00843]$; this interval captures training-seed variation on the fixed splits, not uncertainty from fresh splits or model selection.

Table A5: Five-seed Resonance capacity sweep. Joint utility is the campaign's normalized aggregate across the two domains; MOSEI MAE is lower-is-better and Vision & Touch balanced accuracy is higher-is-better.

Core edge budget	Mean edges	Joint utility	MOSEI MAE ↓	V&T bal. acc. ↑
$0.5\times$	254.2	0.8624	0.6490	0.9411
$1\times$	506.6	0.8655	0.6272	0.9401
$2\times$	1013.2	0.8672	0.6170	0.9400
$4\times$	2026.4	0.8689	0.6125	0.9419

Table A6: Five-seed multimodal controls. Ingress values are joint utility. Bypass values are the native task metrics with 95% Student-t intervals.

Control	Mean	95% CI
Fixed ingress joint utility	0.8667	[0.8647, 0.8687]
Permuted ingress joint utility	0.8677	[0.8659, 0.8694]
Shared ingress joint utility	0.8669	[0.8649, 0.8689]
No-core bypass, MOSEI MAE ↓	0.6108	[0.5955, 0.6262]
No-core bypass, V&T balanced accuracy ↑	0.9443	[0.9416, 0.9470]

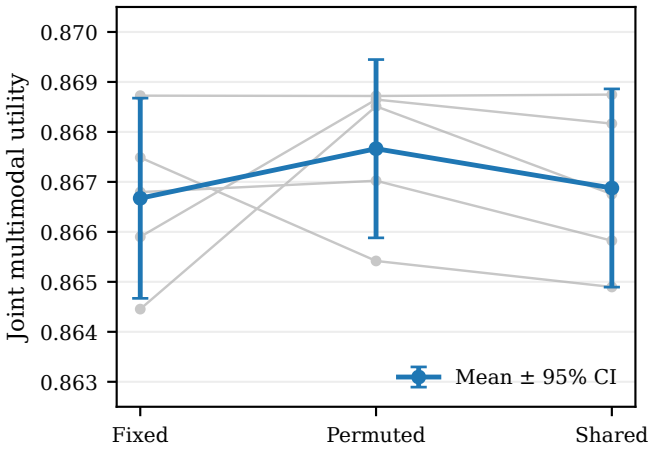

Figure A2: Multimodal ingress control. Joint multimodal utility is similar across fixed, permuted, and shared ingress assignments; points show the mean and 95% confidence interval, with faint lines connecting corresponding seeds.

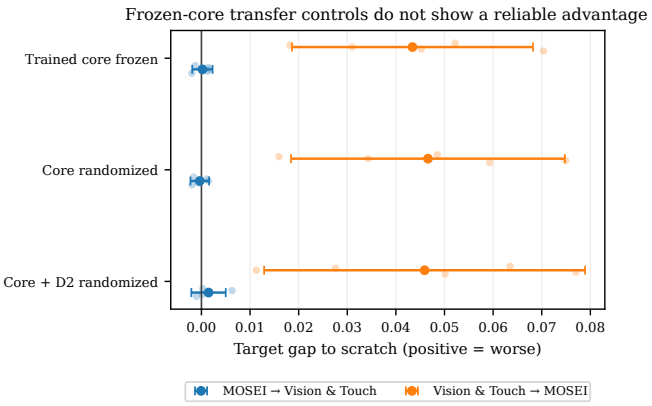

Figure A3: Frozen-core directional transfer controls. The horizontal axis is the target-performance gap to training from scratch, so positive values are worse. Source-trained frozen cores do not show a reliable advantage over core-randomized or core-plus-readout-randomized controls in either transfer direction.

E Mechanism-Ablation Protocol

The mechanism DOE compares the full model with removal of phase rotation, learned geometry, edge selectivity, or structural plasticity under the same 30k-step CLRS training horizon. The final confirmation campaign uses eight training seeds $\{0, \ldots, 7\}$ for both BFS and Bellman–Ford. Evaluation uses the same dependency and post-input tick grids as the primary computation study, with

three common fixed probe replicates of 128 examples per depth for each trained model (384 effective examples per depth after pooling within seed). The probe sets are shared across variants so that full-versus-ablation comparisons are paired.

The primary metric is normalized dependency-by-tick surface AUC: trapezoidal integration in $\log_2(K+1)$ and $\log_2(d+1)$ divided by the corresponding rectangular area. The reported weights are the best validation checkpoints from 30k-step runs, not necessarily the final checkpoints; selected steps range from 6k to 30k. Phase ablation sets the retained-state rotation to identity. Geometry ablation sets $\ell_e = 0$, removing both geometric phase and distance attenuation; it does not merely freeze coordinates. Selectivity ablation makes incoming pooling uniform and sends the unmodified source signal on outgoing edges, jointly removing complex modulation and CReLU. Plasticity ablation disables structural events. Thus the selectivity effect cannot isolate learned routing from its nonlinearity. We also report deep-dependency surface AUC, mean improvement over the zero-post-input-tick condition, best-tick gain on deep dependencies, and degradation at excessive tick budgets. Probe replicates are averaged within each training seed and are never counted as independent statistical replicates. Across seeds we report mean, standard deviation, median/MAD, Student-t and bootstrap 95% confidence intervals, paired full-minus-ablation effects, positive-seed counts, paired Cohen's d_z, and exact two-sided sign-flip tests. The highlighted post-input-gain test is a secondary, unadjusted comparison; the saved analysis reports four ablations, two tasks, and five metrics, so it should not be read as family-wise corrected evidence. The primary surface-AUC comparisons do not identify a single uniformly necessary mechanism; the clearest targeted effect is the BFS post-input-gain reduction when edge selectivity is removed.

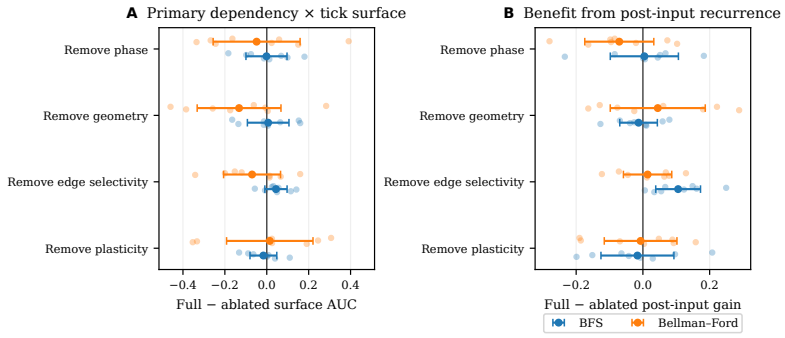

Figure A4: Paired full-minus-ablation effects across eight training seeds. Panel A shows changes in normalized dependency-by-tick surface AUC; Panel B shows changes in the benefit obtained from post-input recurrence. Positive values favor the full model. Effects vary by task and mechanism rather than showing one uniformly necessary component.

F Long-Context Persistence Stress Test

The production BABILong campaign covers QA1–QA5, trains only through 16k context, and evaluates out of distribution through 128k with three seeds for the principal Resonance setting and controlled recurrent/long-context baselines. These are compact local baseline implementations, not reproductions of the published pretrained BABILong leaderboard systems. The carried forward state remains bounded across segments. Because this experiment does not support a strong long-context advantage for the current model, the appendix reports it as a stress test. Tables A7 and A8 summarize the aggregate accuracy, state-size, and causal-intervention evidence; the supplementary data retain complete per-task, per-seed, checkpoint-sensitivity, and extended-tail rows.

Table A7: BABILong QA1–QA5 mean accuracy across three seeds at selected context lengths. State bytes are the mean persistent state footprint for batch size one when an explicit recurrent state is present.

Model	16k	128k	Persistent state bytes
Resonance + associative state	0.316	0.218	89,003
Resonance state only	0.297	0.220	64,043
RMT	0.352	0.280	90,112
Mamba	0.399	0.312	–
Transformer	0.496	0.435	–
GRU	0.430	0.373	6,144
LSTM	0.448	0.386	10,240

Table A8: QA5 at 128k: three-seed persistent-state intervention means. Erasing or shuffling the carried states does not reliably reduce accuracy; zeroing all carried state is numerically higher than the intact condition in this small intervention set, arguing against a strong causal long-memory interpretation.

Condition	Mean accuracy	Condition	Mean accuracy
Intact	0.307	Zero associative state	0.318
Shuffle associative state	0.308	Zero core state	0.307
Shuffle core state	0.305	Zero all persistent state	0.340

The extended 100-example-per-task tail remains low: Resonance + associative state averages 0.213, 0.220, and 0.201 at 256k, 512k, and 1M context respectively; RMT averages 0.253, 0.239, and 0.229 on the same tail.

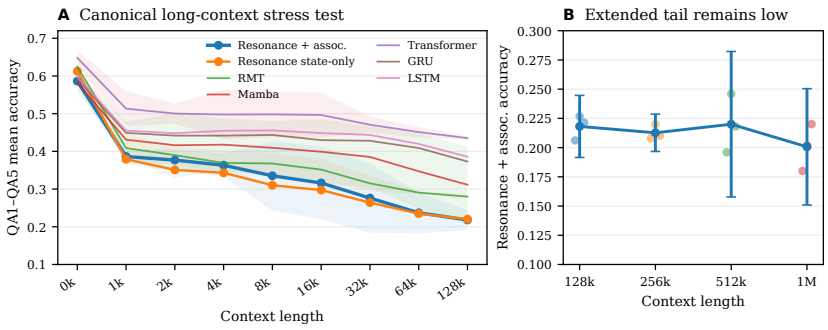

Figure A5: BABILong persistence stress test. Panel A compares QA1–QA5 mean accuracy through 128k context against recurrent and long-context baselines; Panel B extends the Resonance associative-readout condition through 1M context. The current bounded persistent state does not maintain a strong long-context advantage.