Extended Abstract Track

Time-Resolved Circuit Discovery in RNNs via Windowed Causal Interventions and Local Linearization

Editors: List of editors' names

Abstract

Recurrent neural networks (RNNs) have been widely adopted as models of cortical computation, yet their utility for understanding neural mechanisms and explicit structure-function relationships has been limited by their opacity. Recent advances in mechanistic interpretability offer new hope for opening these black boxes, moving beyond correlation-based analyses to causal understanding. Building on these developments, we present a time-resolved circuit discovery method that reveals how RNNs implement computations through dynamically coordinated subcircuits. Specifically, we combine windowed causal interventions with time-resolved linearization to identify task-critical neurons and visualize the dynamic reconfiguration of effective connectivity, exposing the temporal orchestration of information flow. We demonstrate our approach on RNNs trained with Dale's law constraints to perform a context-dependent flip-flop task, identifying distinct circuits for memory maintenance, state switching, and context-gated control. We find that excitatory and inhibitory neurons show consistent functional specialization: memory circuits are dominated by recurrent excitation, while switching circuits recruit inhibitory neurons at transition points. Critically, our time-resolved analysis reveals that during context switches, the memory circuit remains stable while a separate gating circuit dynamically reconfigures; a temporal dissociation invisible to static analyses. These findings demonstrate that mechanistic interpretability can bridge the gap between artificial and biological neural networks, transforming RNNs from black-box function approximators into white-box models of neural computation. We hope that our work encourages further development of such tools that bear the promise to advance our understanding of both, artificial and biological intelligence.

Keywords: RNNs, Mechanistic Interpretability, Circuit Visualization

1. Introduction

Recurrent neural networks (RNNs) are now a common substrate of choice for modeling cortical computation (Mante et al., 2013), supporting memory (Barak, 2017), dynamics (Barak et al., 2013), and feedback (Balwani et al., 2025a) – all hallmarks of neural circuits. Yet, their internal mechanisms remain difficult to understand from simply their weights or average activations, thereby limiting their applicability for the purposes of mechanistically finding neural circuits. Standard correlation-based analyses often reveal *which* units correlate with a variable but not *how* computations are implemented or *when* specific subcircuits are engaged, thereby leaving much to be desired.

Encouragingly, recent progress in mechanistic interpretability suggests a path forward (Wang et al., 2022; Conmy et al., 2023). Coupling causal interventions on internal states with time-local connectivity analysis, we expose the flow of influence during task execution and develop a compact, general-purpose pipeline for **time-resolved neural circuit discovery** in task- or data-trained RNNs. The central idea is to recover dynamically coordinated subcircuits – sets of neurons and connections that are *causally necessary or sufficient*

within short temporal windows – and to visualize how their effective connectivity reconfigures around key task events. To this end, we use (i) **windowed causal interventions** to score neurons by how much swapping or ablating their hidden traces within a brief window changes the per-step decision margin, and (ii) **time-resolved linearization** to render the "used" recurrent connectivity at each timestep by combining the hidden state Jacobian with current presynaptic activity, giving us a sense for the network's dynamics projected onto the local tangent space of the neural manifold at each timestep. Together, the two steps yield both, evidence of causality and an interpretable picture of how that causal effect is transmitted locally in the first order.

We demonstrate our approach on single-layer RNNs trained to respect Dale's law to perform a context-dependent flip–flop task. This task disentangles three computational demands that commonly co-occur in cortical settings: (1) memory maintenance, (2) state switching in response to a relevant pulse, and (3) context-gated control that suppresses distractors and adapts when the context cue flips. Using diagnostic stimuli that isolate each demand, our method recovers distinct but interacting subcircuits specialized for memory, switching, and cue-switching. Across trained models we observe a robust division of labor: memory circuits are dominated by recurrent excitation, switching circuits recruit inhibitory neurons at transition points, and context switches are mediated by a gating circuit whose connectivity reconfigures rapidly while the memory circuit remains stable – a temporal dissociation that would be invisible to simply static summaries of weights or activity.

Altogether, our results illustrate that mechanistic interpretability can indeed turn RNNs from black-box function approximators into more transparent models of neural computation, acting as a lens for biological circuits.

2. Experimental Setup

Task. We study a context-dependent flip-flop task (Sussillo and Barak, 2013) in which the network must maintain a latent state and update it only when a relevant input pulse arrives. At each timestep t, the input x_t contains two pulse channels (A,B) and a context bit $c_t \in \{\mathrm{A}, \mathrm{B}\}$ indicating which channel is currently relevant; distractor pulses may occur on the irrelevant channel. The target $y_t \in \{-1, 0, 1\}$ is the current flip–flop state: an in-context "set" pulse switches the state to 1, an in-context "reset" pulse switches it to -1, and in the absence of a relevant pulse the state persists. Context can change mid-sequence, after which relevance swaps across channels; the output should immediately reflect the correct state under the new context.

Model and training. We use a single-layer Elman RNN with tanh nonlinearity ϕ

$$h_{t+1} = \phi(W_{hh}h_t + W_{ih}x_t), \qquad z_t = W_o h_t, \qquad \hat{y}_t = \mathrm{softmax}(z_t),$$

and train it to classify the flip-flop state at every step using cross-entropy with Dale's backpropagation (Balwani et al., 2025b) which enforces all hidden neurons to satisfy Dale's law by projecting the recurrent weights onto the appropriate sign-constrained orthant after each gradient update. Specifics about the model and training are provided in Appendix A.

Diagnostic stimuli. To isolate computations and their associated subcircuits we leverage three controlled stimulus families that let us attribute causal influence to distinct, time-localized subcircuits for memory, switching, and cue-switching:

- **Memory probe:** Initialize the state with a single in-context pulse, then present a long delay with either no pulses or irrelevant distractors only.
- **State-switching probe:** In-context, deliver a pulse that flips the state (set→reset or reset→set) with no context change and no distractors.
- **Cue-switching probe:** Establish a state, then flip the context channel while injecting distractors on the now-irrelevant channel and occasional relevant pulses on the newly relevant channel.

3. Mechanistic Interpretability for Neural Circuit Discovery in RNNs

To turn trained RNNs into time-resolved circuit diagrams, we first *identify* task-critical units in short windows via *windowed causal interventions* after which we then *render* how these units interact at each timestep using a *time-resolved linearization* that converts activity and gains into an effective connectivity graph. Each of these steps is described intuitively as follows, with their associated details provided in the appendix.

Windowed Causal Interventions. We estimate causal importance within a window $W = [t_s{:}t_e)$ by measuring how small, targeted *soft ablations* and *trace patching* change the per–timestep decision margin m_t. For *necessity*, we attenuate selected hidden units only inside W with scale $\alpha \in [0, 1)$ and re-roll, computing $\Delta_{\text{abl}}(i; W)$ (Li and Janson, 2024) while for *sufficiency*, we patch a unit's trace from a donor run into a target run over W and use $\Delta_{\text{patch}}(i; W)$ (Meng et al., 2022; Zhang and Nanda, 2023). We select the top-k units per computation (memory/switch via patching; cue-switch/gating via soft ablation) to define the circuit. Additional details are given in Appendix B.

Time-Resolved Linearization. We turn momentary network activity into a *used* recurrent graph by scoring each directed edge as "*gate* × *wire* × *signal*." Intuitively, a postsynaptic unit contributes only if it is locally "open" (high gain), where the connection carries an effect (weight), and the presynaptic unit is actually sending activity at that moment. Concretely, let $g_i(t)$ be the postsynaptic gain (the local slope of the hidden nonlinearity), then the instantaneous influence from $j \to i$ is $S_{j\to i}(t) = g_i(t) \cdot W_{hh}[i, j] \cdot h_{t-1}[j]$ (gate × wire × signal). This is exactly a row of the hidden-state Jacobian $J_t^{(h)} = \partial h_t / \partial h_{t-1}$ applied to the current presynaptic activity, so it highlights edges the network is *actually using* at time t. For clarity of visualization we restrict ourselves to only the causally-selected top-k nodes, coloring edges by sign and scaling widths by $|S_{j\to i}(t)|$. Further details appear in Appendix C.

Results. The *memory circuit* (Fig. 1.A) manifests as mostly a tightly interconnected excitatory cluster whose node sizes and recurrent edges remain stable from $t = 9$ to $t = 11$, indicating persistent necessity of these units and steady effective influence for state maintenance; inhibitory nodes appear with fewer blue edges, consistent with a modulatory role that does not particularly drive the computation. However, when aligned to a *cue-switch* at $t = 10$ (Fig. 1.B), while the excitatory cluster largely preserves its size and connectivity (stable maintenance), but a distinct gating subcircuit transiently emerges: inhibitory nodes expand in size precisely at $t = 10$ and blue edges thicken toward task-relevant excitatory

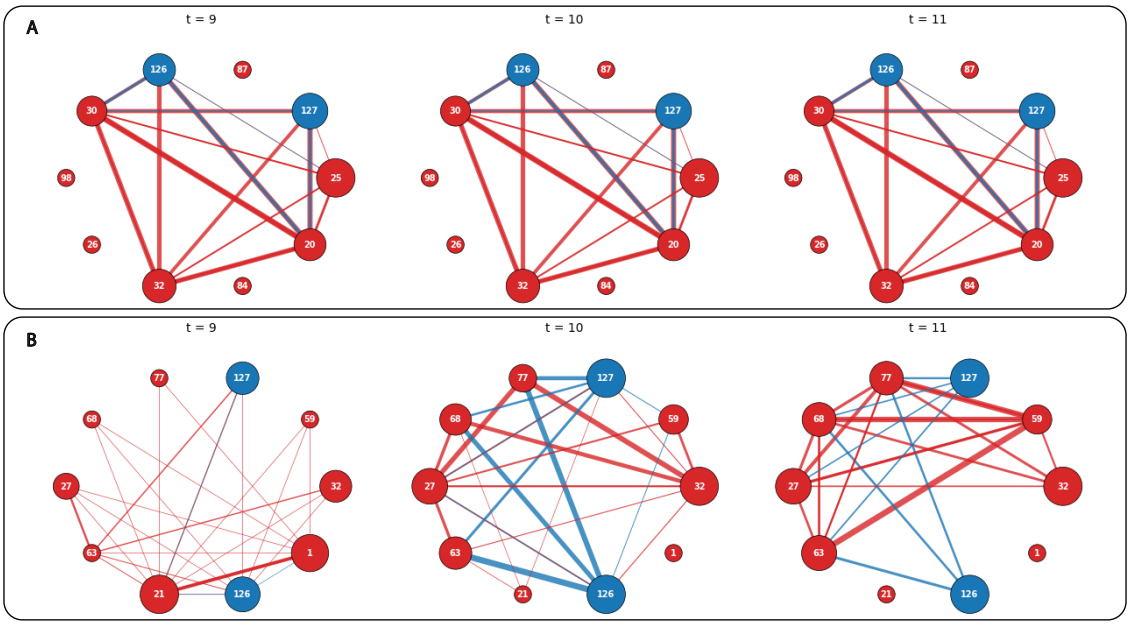

Figure 1: **Time-resolved circuits.** (A) Memory, (B) Cue-switching circuits at time steps t=9, 10, 11 around the switching event at $t = 10$. Only the top-10 neurons (by windowed causal score) are shown. Node color: excitatory (red), inhibitory (blue). Node size: proportional to the normalized hidden activity at time t. Edge color: sign of effective influence. Edge width: magnitude of instantaneous influence.

targets and readout pathways, then subside by t=11. This juxtaposition makes the temporal dissociation explicit: memory-related nodes and edges remain comparatively unchanged while cue-switching recruits a brief, reconfiguring inhibitory drive that reroutes input relevance without disturbing the stored state. For contrast, *state-switching* (Appendix D) displays a similar but slightly weaker and delayed inhibitory pattern, presumably reflecting the need to overcome the stable excitatory memory loop to flip the state.

4. Conclusion

Our time-resolved circuit discovery pipeline transforms RNNs from monolithic black boxes into dynamic assemblies of functionally specialized subcircuits, illuminating concrete architecture – behavior relationships within them. It also opens new avenues for understanding how various biophysical features may shape neural computation; by incorporating different anatomical motifs into our RNN models and interpreting them, we can systematically explore how certain structures or physiological phenomena constrain and enable specific computational strategies. Finally we note that our framework extends naturally to different granularities of analysis: the same windowed approach can be applied to individual neurons, entire E/I populations, layers, or functional modules, allowing one to match the scale of investigation to their hypotheses. By making RNN dynamics more mechanistically interpretable, we hope to accelerate neuroscientific discovery through intuitive and decipherable visualizations that translate neural computations into testable hypotheses.

References

Aishwarya Balwani, Suhee Cho, and Hannah Choi. Exploring the architectural biases of the cortical microcircuit. *Neural Computation*, 37(9):1551–1599, 2025a.

Aishwarya H Balwani, Alex Q Wang, Farzaneh Najafi, and Hannah Choi. Constructing biologically constrained rnns via dale's backprop and topologically-informed pruning. *bioRxiv*, 2025b.

Omri Barak. Recurrent neural networks as versatile tools of neuroscience research. *Current opinion in neurobiology*, 46:1–6, 2017.

Omri Barak, David Sussillo, Ranulfo Romo, Misha Tsodyks, and LF Abbott. From fixed points to chaos: three models of delayed discrimination. *Progress in neurobiology*, 103: 214–222, 2013.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

Maximilian Li and Lucas Janson. Optimal ablation for interpretability. *Advances in Neural Information Processing Systems*, 37:109233–109282, 2024.

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, 2013.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

David Sussillo and Omri Barak. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, 25(3):626–649, 2013.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*, 2023.

Appendix A. RNN Model Architecture and Training Details

Task and targets. Sequences have length T=20 and three input channels per step: two pulse channels $(x_t^{(1)}, x_t^{(2)})$ and a binary *context* $c_t \in \{-1, +1\}$ indicating which pulse is relevant. Training sequences are sampled i.i.d.; pulses on the two channels are drawn with low per-step probability and amplitudes ± 1; the context c_t flips with small probability (e.g., ≈ 0.05) to elicit cue-switch trials. The latent flip–flop state $s_t \in \{-1, 0, +1\}$ is initialized at 0 and updates only on *relevant* pulses (per the current context); otherwise it persists. For cross-entropy, the target at each step is the class index

$$y_t^{\mathrm{CE}} \;=\; \big(s_t + 1\big) \in \{0, 1, 2\} \quad (-1 \mapsto 0,\ 0 \mapsto 1,\ +1 \mapsto 2).$$

By default, steps before the first relevant pulse are supervised as the 0-class.

Network. We use a single-layer, Dale-constrained RNN with a tanh–then–rectify hidden nonlinearity constructed and trained as per Balwani et al. (2025b):

$$\begin{aligned} a_t &= W_{ih}\, x_t + W_{hh}\, h_{t-1} + b_h, \\ u_t &= \tanh(a_t), \\ h_t &= [u_t]_+ \;=\; \max(u_t, 0) \in \mathbb{R}^N, \\ z_t &= W_o\, h_t + b_o \in \mathbb{R}^3, \qquad \hat{y}_t = \mathrm{softmax}(z_t). \end{aligned}$$

Hidden units are partitioned into excitatory $\mathcal{E}$ and inhibitory $\mathcal{I}$ populations in the ratio $\mathcal{E} : \mathcal{I} = 8 : 2$ with 128 hidden neurons in total. Dale's law is enforced on the *recurrent* matrix W_{hh}; Input W_{ih} and readout W_o are unconstrained.

Training objective. We minimize stepwise cross-entropy over the three classes:

$$\mathcal{L} \;=\; \frac{1}{BT} \sum_{b=1}^{B} \sum_{t=1}^{T} \mathrm{CE}\big(z_t^{(b)},\, y_t^{\mathrm{CE},(b)}\big),$$

where B is the batch size and CE is the standard logits-based cross-entropy loss.

Optimization and Dale projection ("Dale's backprop"). Parameters are updated with Adam (learning rate 10^{-3}; other hyperparameters at PyTorch defaults), for 200 epochs with batch size 200. After each optimizer step, we *project* the recurrent matrix onto the sign-constrained set:

$$\big(W_{hh}\big)_{ij} \leftarrow \begin{cases} \max\{(W_{hh})_{ij},\, 0\}, & \text{if presynaptic } j \in \mathcal{E}, \\ \min\{(W_{hh})_{ij},\, 0\}, & \text{if presynaptic } j \in \mathcal{I}. \end{cases}$$

This is the closest sign-consistent update in the Frobenius sense and maintains Dale's law throughout training.

Nonlinearity and postsynaptic gain. With $h_t = [\tanh(a_t)]_+$, the postsynaptic gain used in our time-resolved linearization is

$$D_{\phi,t} = \mathrm{diag}\big((1 - u_t^2) \odot \mathbf{1}\{h_t > 0\}\big), \quad g_i(t) = D_{\phi,t}(i, i) \geq 0.$$

Under Dale's law, the sign of an instantaneous influence $S_{j \to i}(t) = g_i(t)\, W_{hh}[i, j]\, h_{t-1}[j]$ matches the synaptic sign, simplifying E/I interpretation in our circuit visualizations.

Appendix B. Windowed Causal Interventions

We estimate a neuron's causal contribution within a short window $W = [t_s : t_e)$ by intervening on its hidden *trace* and measuring the average change in the per–timestep decision margin. For a label $y_t^\star$, define

$$m_t(z_t, y_t^\star) \;=\; z_t[y_t^\star] \;-\; \max_{c \neq y_t^\star} z_t[c].$$

Soft ablation (necessity). Given a neuron set I and a scale $\alpha \in [0, 1)$, we attenuate their hidden states only inside W and then re-roll:

$$h_{t,i}^{\text{abl}} \;=\; \begin{cases} \alpha\, h_{t,i}, & i \in I,\;\; t \in W, \\ h_{t,i}, & \text{otherwise.} \end{cases} \qquad \Rightarrow \qquad \Delta_{\text{abl}}(I) = \frac{1}{|W|} \sum_{t \in W} \Big(m_t - \tilde{m}_t \Big),$$

where $\tilde{m}_t$ is the margin after ablation. Positive Δ_{abl} indicates necessity in W.

Trace patching (sufficiency). Given a *source* run (e.g., with a switch) and a *target* run (e.g., hold), we copy a single neuron's trace in W and re-roll the target:

$$h_{t,i^\star}^{\text{patch}} \;\leftarrow\; h_{t,i^\star}^{\text{src}}, \quad t \in W, \qquad \Delta_{\text{patch}}(i^\star) = \frac{1}{|W|} \sum_{t \in W} \Big(\tilde{m}_t - m_t \Big).$$

Positive Δ_{patch} indicates the neuron is sufficiently helpful in W.

We test sufficiency for computations that write or restore state (memory/switch), and necessity for computations that filter or gate inputs (cue-switch).

Appendix C. Time-Resolved Linearization (TRL): activity-weighted Jacobian edges

Our analyzed RNN uses a tanh followed by a rectifying clamp:

$$a_t = W_{ih} x_t + W_{hh} h_{t-1} + b, \qquad u_t = \tanh(a_t), \qquad h_t = \big[u_t\big]_+, \qquad z_t = W_o h_t + b_o.$$

Define the *postsynaptic gain* as the diagonal Jacobian of the hidden nonlinearity,

$$D_{\phi,t} \;=\; \text{diag}\Big(\big(1 - u_t^2\big) \odot \mathbf{1}\{h_t > 0\} \Big),$$

so the one-step hidden Jacobian w.r.t. h_{t-1} is

$$\frac{\partial h_t}{\partial h_{t-1}} \;=\; J_t^{(h)} \;=\; D_{\phi,t}\, W_{hh}.$$

We score the *instantaneous recurrent influence* from presynaptic j to postsynaptic i at time t as

$$S_{j \to i}(t) \;=\; \underbrace{D_{\phi,t}(i,i)}_{\text{gate}} \cdot \underbrace{W_{hh}[i,j]}_{\text{wire}} \cdot \underbrace{h_{t-1}[j]}_{\text{signal}}.$$

This "gate × wire × signal" form equals a Jacobian row entry modulated by the *actual* presynaptic activity, highlighting edges that are used at that moment. Because $D_{\phi,t} \geq 0$ and (under Dale) $\text{sign}(W_{hh}[i,j])$ is fixed, the sign of $S_{j \to i}(t)$ is interpretable as excitatory (red) vs. inhibitory (blue). We restrict the rendered graph to the WCI-selected top-k nodes and draw edges with width $\propto |S_{j \to i}(t)|$.

Appendix D. State-Switching Circuit Visualization

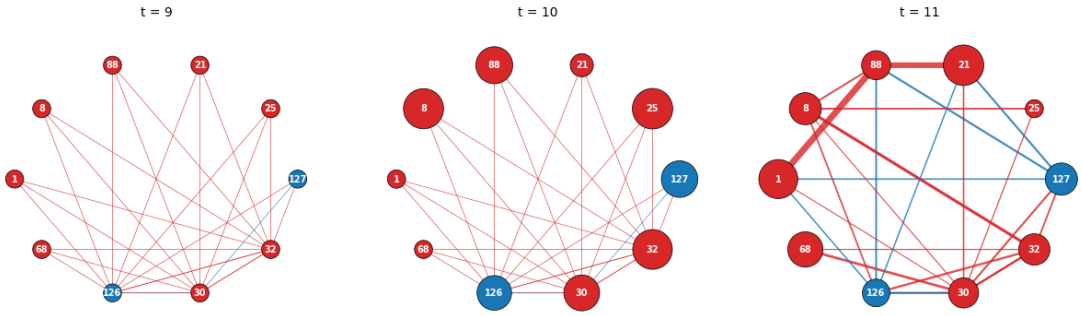

Figure 2: **Time-resolved switching circuit.** Switching circuit at time steps t=9, 10, 11 around the switching event at $t = 10$. Only the top-10 neurons (by windowed causal score) are shown. Node color: excitatory (red), inhibitory (blue). Node size: proportional to the normalized hidden activity at time t. Edge color: sign of effective influence. Edge width: magnitude of instantaneous influence.