Recurrent Hamiltonian Echo Learning Enables Biologically Plausible Training of Recurrent Neural Networks

Motivation. Backpropagation-through-time (BPTT) is the best-known method for training recurrent neural networks (RNNs), but it is widely regarded as biologically implausible because it requires a backward phase separate from inference. Real-Time Recurrent Learning (RTRL) [1] computes exact gradients online without an explicit backward pass; however, its memory cost scales as $\mathcal{O}(N^3)$ in the number of units, making it impractical at scale. This limitation has motivated approximate variants [2–4].

Background. Recurrent Hamiltonian Echo Learning (RHEL) [5] is a recent training framework for Hamiltonian-based models with reversible dynamics. It exploits time-reversal symmetry: after a standard inference phase, the system is run backward in an "echo phase" where loss signals are injected into the state. This echo phase reuses the same neural substrate and encodes gradients in trajectory deviations rather than explicit Jacobians. The equivalence between RHEL and BPTT was established in [5].

Our Contributions. Our contributions are twofold: (i) *theoretical*: we show that applying RHEL in a Hopfield-inspired Hamiltonian network yields a contrastive Hebbian learning rule; and (ii) *empirical*: we demonstrate that such networks trained with RHEL achieve BPTT-level performance on temporal tasks where existing biologically motivated methods fall short.

Hamiltonian Network. Building on previous extension of the continuous Hopfield model [6], we derive a Hamiltonian-based RNN with symmetric connectivity ($W = W^\top$). The system has state variables (ϕ, π), and nonlinearity $\rho(\cdot)$. The dynamics are governed by the Hamiltonian function: $H(\pi, \phi) = \frac{1}{2}\|\pi\|^2 + \frac{\alpha}{2}\|\phi\|^2 + \frac{1}{2\beta}\rho(\phi)^\top W \rho(\phi) + b^\top \rho(\phi) + u^\top B \rho(\phi)$ with bias b, input u, and input map B. Unlike equilibrium propagation [6], which is restricted to static inputs, this reversible formulation in combination with RHEL enables *temporal credit assignment*.

Learning Rule. RHEL yields the local contrastive Hebbian rule: $\Delta W_{ij} \propto \int_0^{T_f} \Big[\rho(\phi_i)\rho(\phi_j) - \rho(\phi_i^e)\rho(\phi_j^e)\Big] dt$, where superscript e denotes the echo phase. Each synapse only requires two local accumulators, recording pre–post correlations during the inference and echo phases.

Benchmarks. We evaluate our RNN trained with RHEL on two temporal learning benchmarks: Heidelberg-Digits [7], a spoken-digit classification dataset with mel-spectrogram inputs; and Copy-Paste [8], where the network must recall 400 dt sequences of 3-bit patterns after a delay.

Performance. On compact architectures (64–128 units, <20K parameters), RHEL outperforms biologically motivated baselines such as e-prop [2] and truncated BPTT (2-steps), while matching full BPTT accuracy (Table 1). On Copy-Paste, all effective methods saturate near 100%, and RHEL achieves this while alternatives fall short.

Method	Heidelberg-Digits	Copy-Paste
Truncated	90.27 ± 0.80	55.44 ± 6.14
Eprop	94.95 ± 1.34	77.77 ± 0.25
Ours/Rhel	**97.61 ± 1.03**	**99.36 ± 0.02**
Bptt	97.56 ± 0.62	99.46 ± 0.05

Table 1: Mean test accuracy ± std. over 5 seeds.

Gradient Alignment. We assess gradient fidelity by measuring cosine similarity with BPTT updates during training (Fig. 1). RHEL maintains the closest agreement with BPTT among biologically plausible methods, supporting its theoretical gradient equivalence and explaining its strong performance.

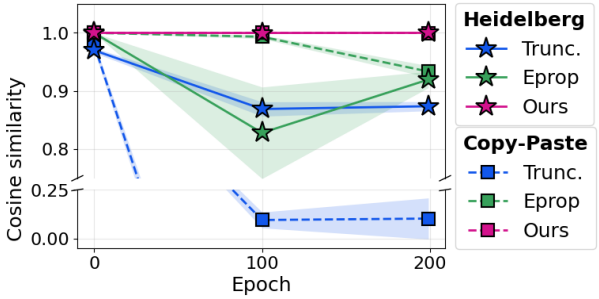

Figure 1: Cosine similarity with BPTT at epochs 0, 100, and 199. Shaded regions show the standard error of the mean (s.e.m.) over 5 seeds. The y-axis is broken for readability; the two panels use different scales.

Further Work. Future directions include scaling RHEL to larger networks, extending it to spiking models, and testing on more challenging sequential tasks.

References. [1] Williams, R. J., et al. (1989). *Neural Comput.* 1(2):270–280. [2] Bellec, G., et al. (2020). *Nat. Commun.* 11:3625. [3] Zucchet, N., et al. (2023). In *NeurIPS*. [4] Ellenberger, B., et al. (2024). *arXiv:2403.16933*. [5] Pourcel, G., et al. (2025). *arXiv:2506.05259*. [6] Scellier, B., et al. (2017). *Front. Comput. Neurosci.* 11. [7] Cramer, B., et al. (2022). *IEEE Trans. Neural Netw. Learn. Syst.* 33:2744–2757. [8] Hochreiter, S., et al. (1997). *Neural Comput.* 9(8):1735–1780.