# DOES FEEDBACK ALIGNMENT WORK AT BIOLOGICAL TIMESCALES?

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Biological learning unfolds continuously in time, yet most algorithmic models of error propagation rely on discrete updates and separate inference and learning phases. We study a continuous-time neural model that encompasses several biologically plausible learning algorithms and removes the need for phase separation. Rules including stochastic gradient descent (SGD), feedback alignment (FA), direct feedback alignment (DFA), and Kolen–Pollack (KP) emerge naturally as limiting cases of the dynamics. Simulations show that these continuous-time networks stably learn at biological timescales, even under temporal mismatches and integration noise. Our results reveal that, in the absence of longer-range memory mechanisms, learning is constrained by the temporal overlap of inputs and errors. Robust learning requires potentiation timescales that outlast the stimulus window by at least an order of magnitude, placing the effective eligibility regime in the few-second range. More broadly, this identifies a unifying principle: *learning succeeds when input and error are temporally correlated at each synapse*, a rule that yields testable predictions for neuroscience and practical design guidance for analog hardware.

## 1 INTRODUCTION

Understanding how biological circuits learn has long been a central challenge at the interface of neuroscience and machine learning. Among the many proposals for biologically plausible learning, a prominent class consists of feedback-alignment-style, weight-transport-free algorithms such as feedback alignment (FA; Lillicrap et al. (2016)), direct feedback alignment (DFA; Nøkland (2016)), and Kolen–Pollack / weight mirrors (KP; Kolen & Pollack (1994); Akrout et al. (2019)). These algorithms aim to relax the strong requirements of backpropagation, such as exact weight transport, while preserving the ability to optimize deep networks. These models are attractive both as potential models of cortical learning and as candidates for training in neuromorphic or analog hardware.

Two key aspects of real biological systems, however, are conspicuously absent from most error propagation formulations: learning in biology unfolds continuously in time, and does not rely on separate learning and inference phases. Realistic neurons operate with finite conduction and integration times, so neither inference, error propagation, nor plasticity can be assumed to occur instantaneously or in synchronized steps (Kandel et al., 2000). Unlike digital learning algorithms, biological systems do not alternate between distinct inference and learning phases. By contrast, most existing algorithmic models of error propagation-driven learning are cast in discrete steps that alternate between inference and learning phases, effectively assuming that forward and backward signals are globally and instantaneously synchronized (Lillicrap et al., 2016; Nøkland, 2016; Akrout et al., 2019; Scellier & Bengio, 2017). This leaves a basic question unaddressed: *Do FA/KP/DFA-style learning rules still work when implemented as continuous-time processes with biologically realistic propagation and plasticity timescales?*

In this work, we address this gap directly. We construct a continuous-time model in which both neural states and synaptic weights evolve according to coupled first-order differential equations. Each neuron receives a feedforward drive and a modulatory error drive, and updates its synapses through a local two-signal rule. Crucially, inference and learning occur simultaneously: there is no global phase separation, as all state variables evolve under the same ODE. The model is governed by distinct time constants for fast signal propagation, intermediate potentiation, and slow synaptic decay, an ordering that mirrors the hierarchy of timescales in real neural tissue.

Our contributions are threefold:

1. **Experimentally tested continuous-time formulation of feedback alignment style algorithms.** We specify a fully dynamical architecture for feedback alignment style algorithms, without retaining algebraic error terms or instantaneous pathways. All computational variables, including error representations, are governed by continuous-time dynamics. Inference and learning run concurrently, and classical algorithms such as SGD, FA, DFA, and KP emerge as limit cases (Section 3).
2. **Temporal learning analysis.** We analyze the effect of timing mismatches and show that learning is determined by the temporal overlap between input and error. This predicts failure when delay approaches the sample duration, depth fragility from accumulated lags, and robustness patterns observed in simulation (Sections 4–5).
3. **Biological timescales.** We demonstrate effective learning at biologically realistic timescales and derive a quantitative constraint: potentiation windows must outlast the stimulus by an order of magnitude, placing the functional eligibility regime in the few-second range (Section 5.3).

Together, these results support the biological feasibility of rules like KP, FA, and DFA when cast in continuous time. They show that plasticity arises from temporal overlap between input and error, explaining the robustness of these algorithms and providing a physically grounded path toward continuous-time learning in both biological and artificial systems.

## 2 Related Work

Related families of learning algorithms, such as contrastive Hebbian learning (Xie & Seung, 2003), equilibrium propagation (Scellier & Bengio, 2017), and other energy-based formulations (Hopfield, 1982; Bengio & Fischer, 2015), also instantiate continuous-time learning rules. Latent-equilibrium networks and their extensions derive both neural and synaptic dynamics as gradient flows so that arbitrarily slow neurons can approximate backpropagation (and backpropagation through time) without explicit forward/backward phases (Haider et al., 2021; Ellenberger et al., 2024). These approaches differ from the heterosynaptic two-signal rules that are our focus here. While our model shares the spirit of framing learning as a dynamical process, we restrict our focus in this work to error-propagation style rules (FA, DFA, KP) and their continuous-time realizations.

Whittington and Bogacz (Whittington & Bogacz, 2019) review biologically plausible backpropagation schemes, but emphasize predictive-coding and dendritic-error frameworks rather than weight-transport-free methods such as feedback alignment or Kolen–Pollack. Predictive coding (Whittington & Bogacz, 2017) uses continuous-time ODEs for neuronal states but treats error terms and weight updates as instantaneous algebraic quantities. Similarly, dendritic error models (Sacramento et al., 2018) compute apical errors by algebraic functions of current somatic activity, which is distinct from feedback alignment. Our work is orthogonal to these directions: we analyze feedback alignment type rules (FA, DFA, and KP) in layered feedforward networks, showing that these discrete-time algorithms admit a continuous-time formulation in which neural states and synaptic weights coevolve under coupled ODEs.

Both neural ODEs (Chen et al., 2018) and our neural differential-equation model cast network computation as a continuous-time dynamical system, replacing discrete layers/updates with ODE flows over time. In standard neural ODEs, parameters are fixed during the forward solve, and gradients are typically recovered by integrating an adjoint ODE backward in time (Chen et al., 2018) or by differentiating through the solver (Baydin et al., 2018). By contrast, our model couples learning and inference in one forward-in-time system: both neural states and parameters evolve by ODEs, with weights updated online via locally computed and propagated error terms.

**Scope of our theoretical claims.** In our work, statements about SGD, FA, DFA, and KP as limiting cases of heterosynaptic two-signal rules are recapitulations of the discrete-time analysis of Ziyin

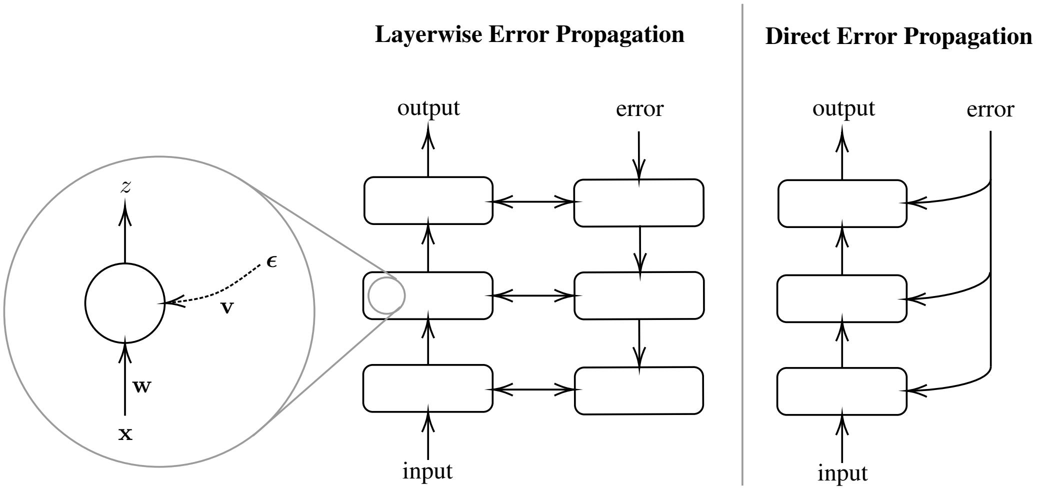

Figure 1: Neuron design with two archetypal topologies. At left, (zoom): continuous-time heterosynaptic neuron model whose weights evolve according to plasticity rules. Neurons receive forward input $\mathbf{x}$ and produce activated output $z$ with weights $\mathbf{w}$. Error signals $\boldsymbol{\epsilon}$ enter through modulatory weights $\mathbf{v}$ and drive plasticity of $\mathbf{w}$. In center: layerwise error propagation topology, where errors are propagated to previous layers via backward weights. At right: direct error propagation topology, where error signals are broadcast to previous layers directly.

et al. (2025), Lillicrap et al. (2016), Nøkland (2016), and Akrout et al. (2019). Our contribution lies in the continuous-time, delay-limited formulation and the analysis of temporal overlap and timescale constraints; we therefore do not restate or reprove the discrete-time equivalence results here.

## 3 A Continuous-Time Model of Feedback Alignment Error Propagation

Figure 1 illustrates the continuous-time heterosynaptic neuronal systems we study. The forward pathway, parameterized by $\mathbf{w}$ and with activation $\sigma$, maps inputs $\mathbf{x}$ to neuronal output $z$. The error pathway, parameterized by $\mathbf{v}$, receives an error signal $\boldsymbol{\epsilon}$ and provides a modulatory influence on learning. Learning is heterosynaptic: the forward weights $\mathbf{w}$ update in proportion to the modulatory drive $(\mathbf{v}^\top \boldsymbol{\epsilon})$, while the error weights $\mathbf{v}$ update in proportion to the forward drive $(\mathbf{w}^\top \mathbf{x})$. Let layer $l$ have width $d_l$. Stacking the per-neuron variables into matrices, we can use the following notation: $\mathbf{z}_{l-1}(t) \in \mathbb{R}^{d_{l-1}}$, $\mathbf{z}_l(t) \in \mathbb{R}^{d_l}$, $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$, and error source $\boldsymbol{\epsilon}_l(t)$. Define

$$\dot{\mathbf{z}}_l = \frac{-\mathbf{z}_l + \sigma_l(W_l^\top \mathbf{z}_{l-1})}{\tau_{\text{prop}}}, \quad \dot{W}_l = -\frac{W_l}{\tau_{\text{dec}}^W} + \frac{\mathbf{z}_{l-1}\,(V_l^\top \boldsymbol{\epsilon}_l)^\top}{\tau_{\text{pot}}^W}, \quad \dot{V}_l = -\frac{V_l}{\tau_{\text{dec}}^V} + \frac{(W_l^\top \mathbf{z}_{l-1})\,\boldsymbol{\epsilon}_l^\top}{\tau_{\text{pot}}^V}. \tag{1}$$

Here $\tau_{\text{prop}}$ is the neuronal propagation time constant, setting how quickly $\mathbf{z}_l$ relaxes to its driven input. The constants $\tau_{\text{pot}}^W, \tau_{\text{pot}}^V$ control how rapidly synaptic potentiation occurs when presynaptic and modulatory drives coincide. The constants $\tau_{\text{dec}}^W, \tau_{\text{dec}}^V$ govern passive weight decay, setting the forgetting timescale. For the biological interpretation of these timescales, see Section 5.3. These equations arise by stacking per-neuron heterosynaptic updates; see Appendix A for the per-neuron update rules.

We emphasize that these equations describe rate-based dynamics, not spiking dynamics: the neuronal state $\mathbf{z}(t)$ evolves continuously without threshold-triggered resets or discrete spikes (Hopfield, 1984; Wilson & Cowan, 1972). Abstracting away spikes allows focus on continuous-time error-propagation rules. Notably, this neuron model performs both learning and inference simultaneously. As neurons are connected, the result is a large coupled dynamical system. Unlike existing algorithms that explicitly separate inference and learning phases, these networks evolve in continuous time: outputs and weights change together in response to input and error signals.

These dynamics can be understood as a continuous-time realization of the heterosynaptic two-signal principle. Under separation of timescales, we may treat neuronal activity as equilibrated relative to the slower synaptic updates. In this quasi-stationary regime, the fast variables satisfy $\mathbf{z}_l = \sigma(W_l^\top \mathbf{z}_{l-1})$, and the weight dynamics reduce to the heterosynaptic updates analyzed in Ziyin et al. (2025), where local stability guarantees each neuron implements a gradient step and global consistency ensures alignment across layers. At full stationarity ($\dot{\mathbf{z}} = \dot{W} = \dot{V} = 0$) and with equal size, the fixed-point equations imply $W_l \propto \mathbf{z}_{l-1}(V_l^\top \boldsymbol{\epsilon}_l)^\top$ and $V_l \propto (W_l^\top \mathbf{z}_{l-1})\boldsymbol{\epsilon}_l^\top$, so that $W_l$ converges to a configuration consistent with $V_l^\top$. In this limit, the effective update to $W_l$ is the outer product of the input with the backpropagated error at that layer—recovering the canonical backpropagation rule as the stationary solution of the dynamics.

The center and right panels of Figure 1 show how these neurons connect into multilayer networks. Under timescale separation $\tau_{\text{prop}} \ll \tau_{\text{pot}} \ll \tau_{\text{dec}}$, the activation states equilibrate quickly, and weights update quasi-statically. Outputs track their steady state $\mathbf{z}_l \approx \sigma_l(W_l^\top \mathbf{z}_{l-1})$ within a sample window. Averaging the weight ODE over one input data presentation gives a weight update $\Delta W_l \propto \mathbb{E}_{\text{window}}\left[\mathbf{z}_{l-1}(V_l^\top \boldsymbol{\epsilon}_l)^\top\right]$. The effective weight update rule corresponding to different learning algorithms depends only on the definitions of the feedback weights $V_l$ and error drive $\boldsymbol{\epsilon}_l$. The following discrete-time limits corresponding to SGD, FA, DFA, and KP follow directly from the updates analyzed by Ziyin et al. (2025). Given the error $\mathbf{e}_L$ as the gradient of the loss function with respect to the output layer:

**SGD.** Under the constraint $V_l = W_l^\top$, the error signals are equal to the true backpropagated error, $\boldsymbol{\epsilon}_l = \mathbf{e}_l$. Then, $\Delta W_l \propto \mathbb{E}[\mathbf{z}_{l-1}\mathbf{e}_l^\top]$, which is exactly the gradient of the loss.

**Feedback Alignment (FA).** Set $V_l$ to fixed random matrices that project the error at each layer backwards to the previous layer. Then, $\Delta W_l \propto \mathbb{E}[\mathbf{z}_{l-1}(V_l^\top \boldsymbol{\epsilon}_l)^\top]$, which is the standard FA update. Gradient alignment arises empirically (Lillicrap et al., 2016).

**Direct Feedback Alignment (DFA).** Set $V_l$ to fixed random matrices. Broadcast the output-layer error $\mathbf{e}_L$ to all hidden layers, such that $\boldsymbol{\epsilon}_l = \mathbf{e}_L$. Then, $\Delta W_l \propto \mathbb{E}[\mathbf{z}_{l-1}(V_l^\top \mathbf{e}_L)^\top]$, which is the DFA update. Each layer learns from a direct projection of the global error (Nøkland, 2016).

**Kolen-Pollack (KP) / Weight-Mirror Methods.** Allow $V_l$ to evolve under the full plasticity rules in equation 1. $V_l$ tracks the correlation between the forward drive $W_l^\top \mathbf{z}_{l-1}$ and the error $\boldsymbol{\epsilon}_l$. In expectation, this update combined with weight decay pushes $V_l$ towards $W_l^\top$. Thus KP can be viewed as a dynamical mechanism that drives $V_l \to W_l^\top$, in contrast to SGD which assumes this equality is enforced from the start (Akrout et al., 2019; Kolen & Pollack, 1994).

We simulate training and inference of these models using ODE solvers (Kidger, 2021). Inputs to the network are driven by the dataset's input signals, and the error at the output neurons in layer $L$ are $\mathbf{e} = \partial\mathcal{L}/\partial\mathbf{z}_L$. Figure 2 shows an example of the output-layer dynamics of early training. Each data point is presented to the network's input neurons for a fixed duration of time (sample time), with interpolation between the images done smoothly over a constant, much shorter duration (buffer time). Rather than impose the corresponding constraints and clamps on weights as traditional SGD/KP/FA/DFA prescribe, our experiments leave both $W$ and $V$ free to learn. We find that this makes our network architecture more general without sacrificing task performance.

During evaluation, weights are frozen and the model runs without error feedback. Only the inference dynamics of $\mathbf{z}$ are active, while $\mathbf{e}$ is clamped to 0 and the learning rules for $W$ and $V$ are disabled. The prediction is read out at the very end of the input sample time, right before it switches.

## 4 Timing Robustness

A central question is: *under what conditions does a synapse receive a correct update?* Two key considerations are (i) the temporal mismatch between input and error signals, and (ii) the rate at which the input changes. Figure 4 illustrates the dynamics of a final-layer weight when the error signal arrives earlier or later than the corresponding input. During the mismatch period, the instantaneous weight update is incorrect, leading to a cumulative update that deviates from the non-delayed case.

To analyze robustness, we isolate only the part of the weight update that is informative: that which is proportional to the correlation between presynaptic activity and the matching error drive. Vectorized,

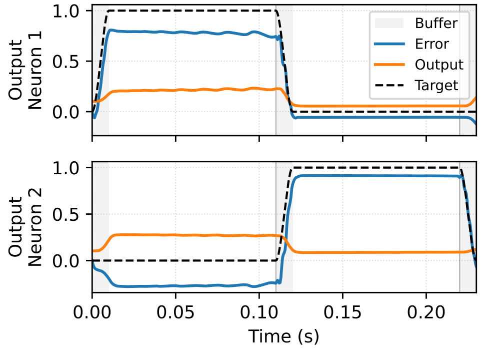

Figure 2: Zoomed-in view of output neuron dynamics during early training. The network input and corresponding error term changes during the *buffer* time periods. Over a single sample, the neuron outputs move only minutely.

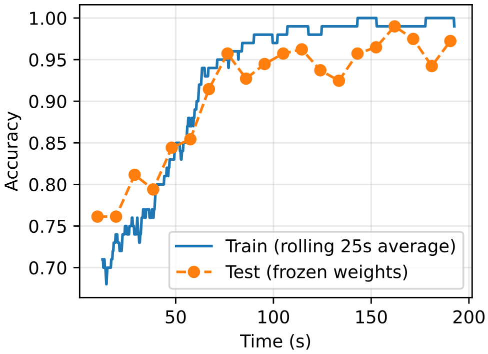

Figure 3: Sample train/test accuracy during training. Training set accuracy is measured as the moving average of instantaneous correct/incorrect predictions on the network input signals. Test set accuracy is measured by evaluating the network with frozen weight dynamics.

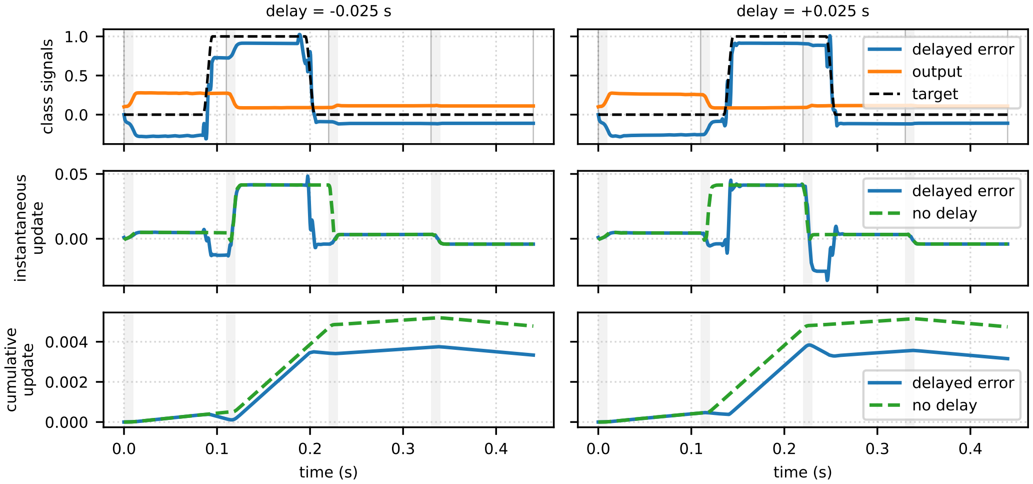

Figure 4: Single-neuron dynamics with different relative timings of the error and input signals. Left: error signal arrives early. Right: error signal is delayed. The bottom panel of both plots depicts the cumulative weight change and shows that in the presence of delay, the weight accumulates a biased gradient update compared to the case where there was no error delay.

the informative weight change accumulated over a presentation window of length $T$ is

$$\Delta W_l \;\propto\; \int_0^T \mathbf{z}_{l-1}(t)\,\big(V_l^\top \boldsymbol{\epsilon}_l(t)\big)^\top k_{\tau_{\text{pot}}}(t)\,dt, \qquad k_{\tau_{\text{pot}}}(t) = \exp\big(-(T-t)/\tau_{\text{pot}}\big), \tag{2}$$

where the causal exponential kernel arises from the low-pass potentiation dynamics and weights more recent coincidence more strongly. For a single synapse $(i \to j)$, this reduces to the scalar form

$$\Delta(W_l)_{ij} \;\propto\; \int_0^T (\mathbf{z}_{l-1}(t))_i\,\big((V_l)_{:,j}^\top \boldsymbol{\epsilon}_l(t)\big)\,k_{\tau_{\text{pot}}}(t)\,dt. \tag{3}$$

This makes clear that learning depends on the temporal cross-correlation between the presynaptic drive $(\mathbf{z}_{l-1})_i$ and the local modulatory/error drive at neuron $j$.

**Piecewise-constant inputs with delay.** Assume $(\mathbf{z}_{l-1})_i(t)$ is active on $[0, T]$ and the error drive is active on $[\Delta, \Delta + T]$ (same duration, delayed by $\Delta$). In the fast-propagation limit $\tau_{\text{prop}} \ll \tau_{\text{pot}}$, the expected update becomes

$$\mathbb{E}[\Delta(W_l)_{ij}] \propto \int_{t_0}^{t_1} \exp\big((t-T)/\tau_{\text{pot}}\big)\, dt$$

$$= \tau_{\text{pot}}\Big(e^{-(T-t_1)/\tau_{\text{pot}}} - e^{-(T-t_0)/\tau_{\text{pot}}}\Big) = \tau_{\text{pot}}\, e^{-(T-t_1)/\tau_{\text{pot}}}\Big(1 - e^{-L/\tau_{\text{pot}}}\Big)\,, \quad (4)$$

where $t_0 = \max(0, \Delta)$, $t_1 = \min(T, \Delta + T)$, and $L = t_1 - t_0 = (T - |\Delta|)_+$.

**Flat-kernel limit and prediction.** When $T \ll \tau_{\text{pot}}$ (the regime standard in our experiments), $k_{\tau_{\text{pot}}}$ is approximately constant over $[0, T]$, so equation 4 reduces to the symmetric triangular law

$$\mathbb{E}[\Delta(W_l)_{ij}] \;\propto\; (T - |\Delta|)_+. \quad (5)$$

Thus learning succeeds if and only if input and error *overlap in time*, and it degrades sharply as $|\Delta| \to T$. When $T$ approaches $\tau_{\text{pot}}$, the exact expression equation 4 predicts a mildly *skewed triangle* that up-weights late-arriving errors (positive $\Delta$) relative to equally early ones; the skew vanishes continuously as $\tau_{\text{pot}}/T \to \infty$.

**A fixed overlap budget.** For delay $\Delta$, define the correct-overlap set $C(\Delta) = [0, T] \cap [\Delta, \Delta + T]$ of length $L = (T - |\Delta|)_+$ and mismatched set $I(\Delta) = [0, T] \setminus C(\Delta)$. With potentiation kernel $k(t)$, let

$$K_C(\Delta) = \int_{C(\Delta)} k(t)\, dt, \quad K_I(\Delta) = \int_{I(\Delta)} k(t)\, dt,$$

so that $K_C(\Delta) + K_I(\Delta) = K_T := \int_0^T k(t)\, dt$. Thus overlap and mismatch trade off under a fixed budget. In the flat-kernel case $k \equiv 1$, this reduces to $L + (T - L) = T$, yielding $\mathbb{E}[\Delta(W_l)_{ij}] \propto L = (T - |\Delta|)_+$, i.e. accuracy improves with either larger $T$ or smaller $|\Delta|$.

Our analysis predicts and experiments confirm near-symmetry between early and late error in the flat-kernel regime. By contrast, biological plasticity is causally gated: synaptic activity first writes a short-lived eligibility trace, and only subsequently arriving modulatory/error signals consolidate it into weight change (Yagishita et al., 2014). Thus, early error fails to drive learning even though late error can—an asymmetry our simplified model compresses into a symmetric overlap law. Incorporating an explicit eligibility gate would recover this asymmetry without altering our conclusions about temporal overlap and timescale separation; we leave such extensions for future work.

# 5 Experiments

In this section, we evaluate three experimental regimes. Section 5.1 demonstrates that networks can robustly learn so long as error signals overlap with inputs. Section 5.2 demonstrate that deeper networks accumulate propagation lag, making them less tolerant to error delays and requiring longer sample times for stable learning. Finally, Section 5.3 shows that our framework operates effectively under synaptic, plasticity, and decay timescales that align with known cortical physiology.

## 5.1 Direct Error Routing

The direct error routing topology sets the local error at each layer to be the global error (from the output layer), $\boldsymbol{\epsilon}_l = \mathbf{e}_L$. This corresponds to a Direct Feedback Alignment scheme where the backward weights $V_l$ are also learned. Figure 5 examines the accuracy of the direct error routing topology on the $7 \times 7$ downsampled MNIST dataset as the error-signal delay and sample duration are varied.

In the direct error routing topology, each layer receives a direct copy of the error signal. As a result, all neurons in the network receive the error signal with a propagation delay of one synaptic length. Learning fails when the error signal delay is approximately equal to or longer than the sample duration, since the error signal has zero overlap with its corresponding input. In these regions (accuracy $\sim 10\%$), weights only change due to spurious correlations between mismatched input-label pairs. Transitional regions ($\sim 70\%$ accuracy) arise when error and input partially overlap: some useful updates occur, but they are mixed with incorrect updates from mismatched periods. These networks have a fixed $\tau_{\text{pot}} = 10$s while the sample time $T$ lies in 0.01-0.25s, putting them primarily in the flat-kernel learning limit. As a result, learning responds fairly symmetrically to both early and late error signals.

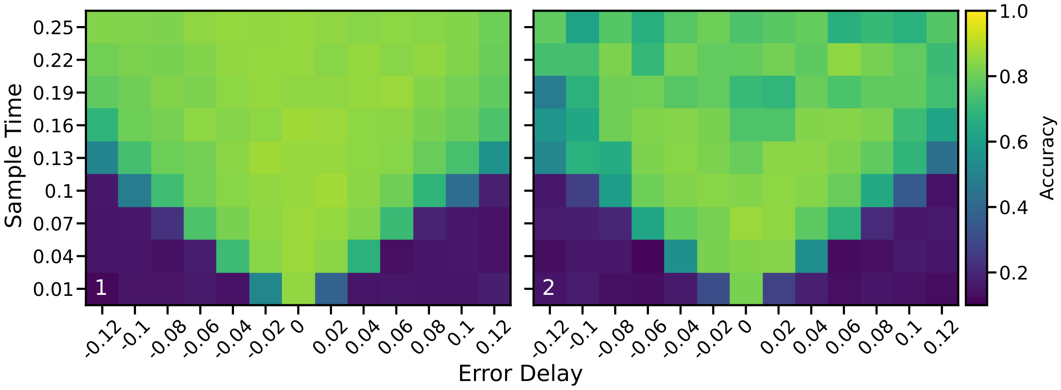

Figure 5: Evaluation of the direct error routing topology on the $7 \times 7$ downsampled MNIST dataset. The $x$-axis denotes the temporal delay between input signal and label: negative values indicate the label arrives before the input, while positive values indicate the label arrives after the input. At left is a network with 1 hidden layer of 49 neurons, and at right is a network with 2 hidden layers of 49 and 32 neurons. The learning is fairly robust until the delay exceeds the sample time.

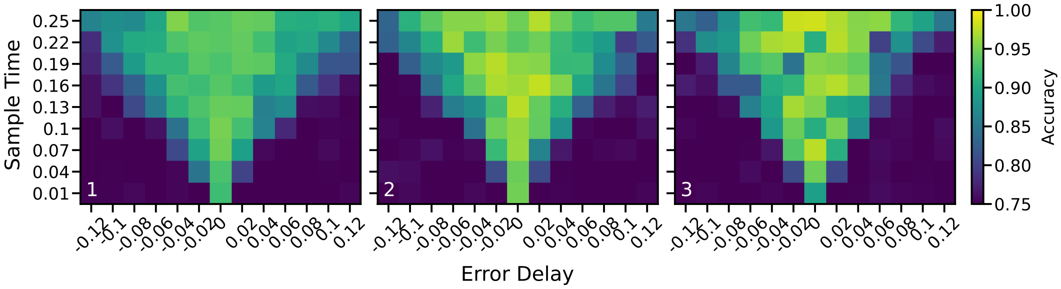

Figure 6: Evaluation of the layerwise error routing topology on the circle dataset as error signal delay and sample duration are swept. The network topologies increase in the number of hidden layers, where each added hidden layer has 24 neurons. From left to right, the networks have one hidden layer of 24 ReLU neurons, two hidden layers of 24 ReLU neurons, and three hidden layers of 24 ReLU neurons. Layerwise error routing imposes stricter requirements on delay due to longer error propagation paths.

### 5.2 Layerwise Error Routing

The layerwise error routing topology propagates error from the output layer step by step to previous layers via $V_l^\top \boldsymbol{\epsilon}_l$. This corresponds to a KP or weight mirroring scheme where the backward weights $V_l$ are learned and converge to transposes of the forward weights. Figure 6 examines the accuracy of the layerwise error routing topology on a synthetic two-dimensional circles dataset (see Appendix B). This task is more nonlinear than the $7 \times 7$ downsampled MNIST task, and requires deeper ReLU networks to represent an accurate decision boundary.

Unlike the direct error routing topology, the layerwise error routing topology suffers more from finite propagation times: not only does the input signal have to propagate to the output layer, but the error signal must also propagate backward from the output layer back to the input layer. The heatmaps show that these deeper networks require longer sample times and are more sensitive to delay in the error signal. Finite propagation speed means that deeper networks accumulate lag, degrading the learning signal especially in early layers. Longer sample durations mitigate this effect by increasing the window where inputs and their true errors coincide. In the language of Section 4, this increases $T$ relative to a fixed $\Delta$, resulting in a greater correct update $\mathbb{E}[\Delta W]$. In the lower triangular regions of the heatmap (large absolute delays), overlap is insufficient and learning fails entirely.

### 5.3 Biological Timescales

A central motivation for these algorithms is their relevance to biological learning. Real neural circuits operate on multiple nested timescales, from millisecond synaptic conductances to second-scale plasticity windows and slower homeostatic processes. Yet most algorithmic work has abstracted away these temporal constraints. Here, we parameterize our continuous-time networks with biologically motivated constants and show that they learn effectively on timescales observed in the brain. Of the three time constants relevant in our model, $\tau_{\text{pot}}$ is the least constrained biologically. Our results therefore provide a new theoretical prediction that narrows its plausible functional range.

In our model, $\tau_{\text{prop}}$ corresponds to the dominant synaptic conductance time constant, which is set primarily by the decay of the postsynaptic current.We interpret the potentiation constant $\tau_{\text{pot}}$ as the biochemical induction gate during which coincident presynaptic drive and modulatory/error input can trigger plasticity via second-messenger cascades. Finally, we assign the synaptic weight decay time constant $\tau_{\text{dec}}$ to be $\sim$ 20 minutes, capturing the gradual decay of synaptic efficiency in the absence of reinforcement. As long as $\tau_{\text{dec}}$ is significantly larger than the propagation and potentiation constants, it primarily sets a slow baseline for weight decay rather than shaping the learning dynamics. We find in simulation that once $\tau_{\text{dec}} \gg \tau_{\text{prop}}, \tau_{\text{pot}}$, its precise value has little effect on performance or dynamics for the tasks we study. Table 1 details the corresponding biophysical processes for each of these time constants.

| Constant | Corresponding Biophysical Process | Typical Range |
|---|---|---|
| $\tau_{\text{prop}}$ | Fast synaptic transmission determined by receptor/channel kinetics and membrane RC filtering. Dominated by AMPA/GABA$_A$ receptor conductance decay after vesicular glutamate/GABA release (Destexhe et al., 1998; O'Brien et al., 1998). | 2–30 ms in cortex; sub-ms in auditory brainstem synapses. |
| $\tau_{\text{pot}}$ | Coincidence-gated plasticity via second-messenger cascades (dopamine D1/D2→cAMP/PKA, Ca$^{2+}$ →CaMKII), regulating AMPAR phosphorylation and trafficking. Defines the biochemical "induction gate" during which pre/post and modulatory signals interact (Yagishita et al., 2014; Gerstner et al., 2018). | ~0.3–10 s depending on circuit (striatal vs. cortical/hippocampal). |
| $\tau_{\text{dec}}$ | Slow synaptic weakening via protein turnover, phosphatase activity, and homeostatic scaling (e.g., AMPAR endocytosis, transcriptional regulation). Provides a gradual baseline decay of synaptic efficacy in the absence of reinforcement (Turrigiano et al., 1998). | Minutes to tens of minutes. |

Table 1: Biophysical timescales in continuous-time network models. Time constants correspond to distinct molecular/cellular mechanisms: rapid receptor/channel kinetics ($\tau_{\text{prop}}$), seconds-scale intracellular signaling and receptor trafficking ($\tau_{\text{pot}}$), and slow homeostatic regulation ($\tau_{\text{dec}}$).

Figure 7 evaluates the trained accuracy of two layerwise error routing networks on these biologically grounded timescales. We find a broad regime where learning is stable and accurate, despite finite propagation and potentiation times. This demonstrates that the timescales $\tau_{\text{prop}} \ll \tau_{\text{pot}} \ll \tau_{\text{dec}}$ characteristic of cortical tissue are sufficient to support effective learning in our continuous-time framework. Furthermore, our results predict that effective feedback-driven learning in cortical-like circuits requires potentiation windows that outlast the stimulus by at least an order of magnitude, placing $\tau_{\text{pot}}$ firmly in the few-second range—a regime that is both biologically plausible and experimentally testable.

## 6 Discussion

**Limitations.** Our empirical evaluation focuses on small datasets and simple tasks due to computational constraints: training of these systems requires explicit forward integration over long time periods of high-dimensional, coupled ODEs whose timescales span over 6 orders of magnitude. This is computationally intensive and numerically delicate (high stiffness, tight tolerances, long horizons), making large benchmarks prohibitive without extensive engineering. Additionally, our neurons are rate-based and simplified, and many biophysical phenomena (spikes, neuromodulatory heterogeneity, etc.) are not modeled. Despite these simplifications, the success of our continuous-

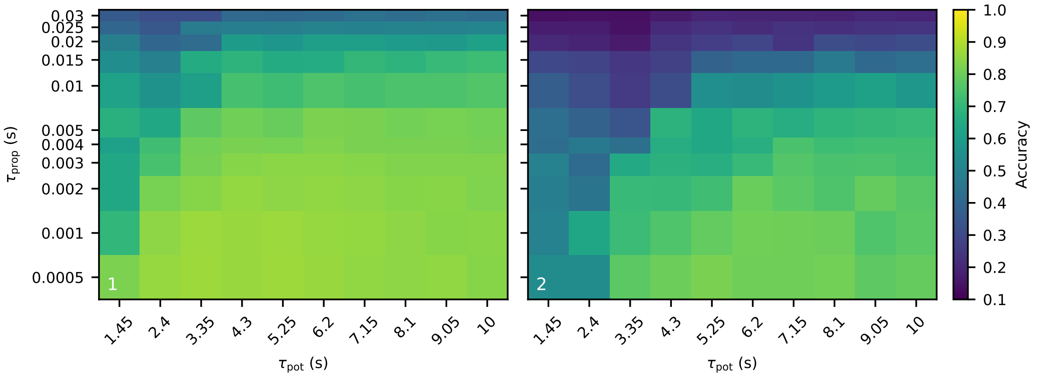

Figure 7: Evaluation of a layerwise error routing network on the $7 \times 7$ MNIST dataset. Left: 1 hidden layer (49 neurons). Right: 2 hidden layers (49 and 32 neurons). Learning is unstable when the potentiation timescale $\tau_{\text{pot}}$ is comparable to the presentation window ($T = 50\,\text{ms}$), but becomes robust only once $\tau_{\text{pot}}$ exceeds $\sim 2\,\text{s}$. This corresponds to $\tau_{\text{pot}}/T \approx 40$, highlighting the requirement that potentiation persist far longer than the input presentation time.

time, overlap-driven networks is promising, and future work may address incorporating more faithful biophysics.

**Temporal overlap as the operative constraint.** In our ODE formulation, the expected update of a synapse is proportional to the temporal cross-correlation between its presynaptic drive $\mathbf{x}$ and a locally projected error $V_l^\top \boldsymbol{\epsilon}_l$, low-pass filtered by the plasticity kernel with time constant $\tau_{\text{pot}}$. For piecewise-constant presentations, this yields the analytic overlap kernel $(T - |\Delta|)_+$. Failure bands seen at $|\Delta| \approx T$ confirm this prediction: deeper networks accumulate propagation delays that reduce overlap in early layers, degrading learning. This perspective unifies algorithms like FA, DFA, and KP under a single causal principle: a synapse can learn if and only if temporally overlapping input and error reach it. Weight symmetry or fixed feedback are not essential; overlap is.

**Potentiation timescale requirement.** From the overlap kernel analysis in Section 4, robust learning requires the potentiation kernel to be effectively flat over a presentation window $T$. Allowing at most a fractional attenuation $\eta$ across the window yields $\tau_{\text{pot}} \geq T/\ln\bigl(1/(1-\eta)\bigr)$. Empirically we observe a transition near $\tau_{\text{pot}}/T \approx 40$, i.e., $\eta \approx 1 - e^{-1/40} \approx 0.025$. With $T = 50\,\text{ms}$, this places the threshold at $\tau_{\text{pot}} \gtrsim 2\,\text{s}$. Thus, in our protocols, seconds-scale eligibility is sufficient and apparently necessary; deeper networks and finite propagation tighten this bound by further reducing effective overlap.

**Depth and architectural shortcuts.** Unlike digital networks, where error gradients are computed simultaneously for all layers, continuous-time networks suffer cumulative delay with depth. This makes learning increasingly fragile as networks grow deeper. Biological circuits appear to mitigate this through architectural shortcuts: skip and feedback pathways are implemented as interareal projections that bypass intermediate layers and, in the neocortex, create shortcuts across the cortical hierarchy (Douglas & Martin, 2004). These connections can provide faster routes for both inference and error signals, preserving temporal overlap in early layers. This interpretation links our results to HSP theory, which argues that arbitrary backward pathways can suffice for credit assignment—our analysis suggests that their role is to preserve overlap, not to enforce symmetry.

**Timescales as a design principle.** We find that biologically measured timescales are sufficient to support robust learning in our framework. The observed hierarchy $\tau_{\text{prop}} \ll \tau_{\text{pot}} \ll \tau_{\text{dec}}$ mirrors the organization of cortical tissue: millisecond synaptic currents, second-scale eligibility traces, and slow homeostatic decay. In simulation, this separation creates a broad regime where learning is stable and accurate, even in the presence of delays and noise. These results suggest that the brain's hierarchy of clocks may be a functional requirement for maintaining gradient-aligned learning in continuous time.

**Implications for hardware.** Analog or neuromorphic implementations face the same challenges as biology: finite propagation speeds, integration imperfections, and heterogeneous device dynamics. Our results indicate that hardware does not need to enforce strict weight symmetry, but must preserve overlap between input and error signals across relevant timescales. Designing circuits with fast signal propagation, intermediate eligibility accumulation, and slow decay may therefore be a practical recipe for stable analog learning systems.

**Broader perspective.**

Our continuous-time formulation gives a model for biological and physical learning where inference and plasticity unfold together, constrained by finite propagation and integration dynamics. In this setting, the essential ingredient is not weight transport or symmetry, but the preservation of temporal overlap between inputs and errors within a hierarchy of timescales.

This perspective yields concrete predictions for biology. Deeper circuits are predicted to be more fragile to propagation delays, motivating anatomical shortcuts (skip and feedback pathways) that preserve overlap in early layers. More importantly, robust learning requires potentiation dynamics that outlast the stimulus window by at least an order of magnitude; with cortical integration times of tens of milliseconds, this places the effective eligibility regime in the seconds range. Thus, we predict that eligibility traces supporting error-driven learning must persist at least on the order of seconds—a regime that experimentalists can directly probe by manipulating trace duration or disrupting long-range feedback pathways.

More broadly, our analysis links cortical physiology, algorithmic proposals like FA/DFA/KP, and the practical requirements of analog hardware. It points to a unifying condition—*learning succeeds when input and error are temporally correlated at each synapse*—that provides both a mechanistic hypothesis for neuroscience and a design rule for neuromorphic systems.

## LLM Usage

LLMs (ChatGPT-5, accessed via web) were used to assist with drafting and refining phrasing of the manuscript. They were also employed to generate portions of the experimental code and plotting scripts. All outputs from the LLM were reviewed, verified, and, where necessary, modified by the authors.

## References

Mohamed Akrout, Collin Wilson, Peter Humphreys, Timothy Lillicrap, and Douglas B Tweed. Deep learning without weight transport. *Advances in neural information processing systems*, 32, 2019.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18(153):1–43, 2018.

Yoshua Bengio and Asja Fischer. Early inference in energy-based models approximates back-propagation. *arXiv preprint arXiv:1510.02777*, 2015.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Alain Destexhe, Zachary F Mainen, Terrence J Sejnowski, et al. Kinetic models of synaptic transmission. *Methods in neuronal modeling*, 2:1–25, 1998.

Rodney J Douglas and Kevan AC Martin. Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.*, 27(1):419–451, 2004.

Benjamin Ellenberger, Paul Haider, Jakob Jordan, Kevin Max, Ismael Jaras, Laura Kriener, Federico Benitez, and Mihai A Petrovici. Backpropagation through space, time, and the brain. *arXiv preprint arXiv:2403.16933*, 2024.

Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Frontiers in neural circuits*, 12:53, 2018.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323. JMLR Workshop and Conference Proceedings, 2011.

Paul Haider, Benjamin Ellenberger, Laura Kriener, Jakob Jordan, Walter Senn, and Mihai A Petrovici. Latent equilibrium: A unified learning theory for arbitrarily fast computation with arbitrarily slow neurons. *Advances in neural information processing systems*, 34:17839–17851, 2021.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

John J Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the national academy of sciences*, 81(10):3088–3092, 1984.

Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, A James Hudspeth, Sarah Mack, et al. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.

Patrick Kidger. *On Neural Differential Equations*. PhD thesis, University of Oxford, 2021.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

John F Kolen and Jordan B Pollack. Backpropagation without weight transport. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 3, pp. 1375–1380. IEEE, 1994.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1): 13276, 2016.

Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Richard J O'Brien, Sunjeev Kamboj, Michael D Ehlers, Kenneth R Rosen, Gerald D Fischbach, and Richard L Huganir. Activity-dependent modulation of synaptic ampa receptor accumulation. *Neuron*, 21(5):1067–1078, 1998.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

João Sacramento, Rui Ponte Costa, Yoshua Bengio, and Walter Senn. Dendritic cortical microcircuits approximate the backpropagation algorithm. *Advances in neural information processing systems*, 31, 2018.

Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017.

Ch Tsitouras. Runge–kutta pairs of order 5 (4) satisfying only the first column simplifying assumption. *Computers & mathematics with applications*, 62(2):770–775, 2011.

Gina G Turrigiano, Kenneth R Leslie, Niraj S Desai, Lana C Rutherford, and Sacha B Nelson. Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391(6670):892–896, 1998.

James CR Whittington and Rafal Bogacz. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation*, 29(5): 1229–1262, 2017.

James CR Whittington and Rafal Bogacz. Theories of error back-propagation in the brain. *Trends in cognitive sciences*, 23(3):235–250, 2019.

Hugh R Wilson and Jack D Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical journal*, 12(1):1–24, 1972.

Xiaohui Xie and H Sebastian Seung. Equivalence of backpropagation and contrastive hebbian learning in a layered network. *Neural computation*, 15(2):441–454, 2003.

Sho Yagishita, Akiko Hayashi-Takagi, Graham CR Ellis-Davies, Hidetoshi Urakubo, Shin Ishii, and Haruo Kasai. A critical time window for dopamine actions on the structural plasticity of dendritic spines. *Science*, 345(6204):1616–1620, 2014.

Liu Ziyin, Isaac Chuang, and Tomaso Poggio. Heterosynaptic circuits are universal gradient machines. *arXiv preprint arXiv:2505.02248*, 2025.

## A Dynamics of a Single Neuron

The single-neuron version of the dynamics presented in equation 1:

$$\dot{z} = \frac{-z + \sigma(\mathbf{w}^\top \mathbf{x})}{\tau_{\text{prop}}}, \quad \dot{\mathbf{w}} = -\frac{\mathbf{w}}{\tau^W_{\text{dec}}} + \frac{(\mathbf{v}^\top \boldsymbol{\epsilon})\,\mathbf{x}}{\tau^W_{\text{pot}}}, \quad \dot{\mathbf{v}} = -\frac{\mathbf{v}}{\tau^V_{\text{dec}}} + \frac{(\mathbf{w}^\top \mathbf{x})\,\boldsymbol{\epsilon}}{\tau^V_{\text{pot}}}.$$

## B Datasets & Methodology

We use two datasets in our evaluation of these models. The $7 \times 7$ downsampled MNIST dataset is the standard MNIST (LeCun, 1998) dataset that has been downsampled with $4 \times 4$ average pooling. The circles dataset is the `make_circles` dataset from scikit-learn (Pedregosa et al., 2011). It is well established that simple linear classifiers achieve surprisingly high accuracy on MNIST, with only modest improvements from deeper architectures (LeCun et al., 2002). We find a linear softmax regressor on $7 \times 7$ downsampled MNIST achieves over 89% test accuracy. In contrast, a logistic regressor on the 2-circle concentric-rings dataset achieves only 75% test accuracy, which corresponds to the class imbalance itself (75% outer ring, 25% inner ring).

$W$ are initialized according Xavier normalization (Glorot & Bengio, 2010). $V$ are initialized to a fixed constant 0.1. Classification decisions are read out from output neurons at the very end of each input sample's presentation. Evaluation is done on frozen $W$ and $V$ dynamics. Each heatmap data point is the average of 3-5 runs, depending on the experiment.

We integrate all continuous-time dynamics using Diffrax's (Kidger, 2021) Tsit5 solver (Tsitouras, 2011) (a fifth-order explicit Runge–Kutta method with an embedded fourth-order error estimate) equipped with a PID adaptive step-size controller ($\texttt{rtol} = 2 \times 10^{-3}$, $\texttt{atol} = 10^{-5}$). This setup allows the solver to take large steps during slowly varying segments of the dynamics while automatically refining steps around rapid transients induced by input switches and error onsets. Although the timescales of our systems span several orders of magnitude, we empirically find that they are not so stiff as to require an implicit method; Tsit5 remains stable and efficient under these tolerances. Compared to a fixed-step forward Euler integrator tuned to resolve the fastest timescale, this adaptive scheme reduces wall-clock time by several orders of magnitude while producing indistinguishable learning curves and dynamics.

## C Comparison to SGD

To ensure a fair comparison, we train baseline multi-layer perceptrons (MLPs) with the same hidden unit counts as our continuous-time networks with layerwise error propagation. Both models are exposed to the same training data samples in the same order. For the baselines, we use the Adam optimizer (Kingma, 2014) with an initial learning rate of 0.001, rather than vanilla SGD, to avoid penalizing the baseline with suboptimal hyperparameters. Training is performed with a batch size of 1, matching the presentation schedule used by the continuous-time networks. All networks use ReLU activations (Glorot et al., 2011) on hidden layers.

Table 2 shows accuracies on the 7 × 7 MNIST dataset. In shallow networks, our layerwise error propagation network matches the baseline on both training and test accuracy, but in deeper networks lags slightly. On simple and fairly linear tasks, this is expected: signal lag accumulates with depth (Section 4), and the dataset itself yields diminishing returns from additional layers.

| | **Train Accuracy** | | **Test Accuracy** | |
|---|---|---|---|---|
| **Model** | Discrete-time | Continuous-time | Discrete-time | Continuous-time |
| 1-layer | $0.8750 \pm 0.0054$ | $\mathbf{0.9318 \pm 0.0028}$ | $0.8634 \pm 0.0056$ | $\mathbf{0.9209 \pm 0.0011}$ |
| 2-layer | $0.8758 \pm 0.0220$ | $\mathbf{0.9318 \pm 0.0032}$ | $0.8673 \pm 0.0259$ | $\mathbf{0.8919 \pm 0.0050}$ |

Table 2: Accuracy (mean ± std) on the $7 \times 7$ MNIST dataset across 3 runs. We compare discrete-time MLP baselines (SGD with Adam, lr= $0.001$) to our continuous-time layerwise error propagation network across architectures with 1–2 hidden layers.

On the more nonlinear task of distinguishing concentric circles, additional depth provides a benefit (Table 3). Our continuous-time layerwise error propagation networks achieve comparable or higher test accuracies than the discrete-time baselines.

| | **Train Accuracy** | | **Test Accuracy** | |
|---|---|---|---|---|
| **Model** | Discrete-time | Continuous-time | Discrete-time | Continuous-time |
| 1-layer | $0.8318 \pm 0.0276$ | $\mathbf{0.9241 \pm 0.0133}$ | $0.8272 \pm 0.0282$ | $\mathbf{0.9470 \pm 0.0101}$ |
| 2-layer | $0.9520 \pm 0.0076$ | $\mathbf{0.9657 \pm 0.0082}$ | $0.9485 \pm 0.0066$ | $\mathbf{0.9683 \pm 0.0109}$ |
| 3-layer | $0.9322 \pm 0.0300$ | $\mathbf{0.9494 \pm 0.0588}$ | $0.9312 \pm 0.0218$ | $\mathbf{0.9568 \pm 0.0452}$ |

Table 3: Accuracy (mean ± std) on the concentric circles dataset across 3 runs. We compare discrete-time MLP baselines (SGD with Adam, lr = $0.001$) to our continuous-time layerwise error propagation network across architectures with 1–3 hidden layers.

We can also compare the resilience of our models versus standard SGD in the presence of label error. In our continuous time networks, this arises due to delays in the error signal, as discussed in Section 4. We can model the same error in standard discrete-time neural networks by presenting a standard neural network optimizer a corresponding fraction of mislabeled data. For example, a delay of half the sample time, $\Delta = \frac{1}{2}T$ in our continuous-time model, corresponds to presenting the correct data pairing $(X_i, Y_i)$ and then the mismatched label $(X_i, Y_{i+1})$, where $i$ is the index into the input and label datasets.

Figure 8 compares our continuous-time neural network with a standard discrete-time neural network trained with *label dithering*, which models the effect of delayed error signals without duplicating training data. Instead of splitting each sample into sub-intervals, we approximate the delay ratio $r = \tau_{\text{delay}}/T_{\text{sample}}$ by a rational fraction $p/q$, and then assign exactly $p$ out of every $q$ samples to use the *previous* label while the rest use the current label. This ensures that the fraction of "mismatched" updates matches the physical delay ratio in expectation, while keeping each sample presented only once. In this way, the training dynamics reflect the temporal overlap of input and delayed error, but the dataset size and number of SGD steps remain unchanged.

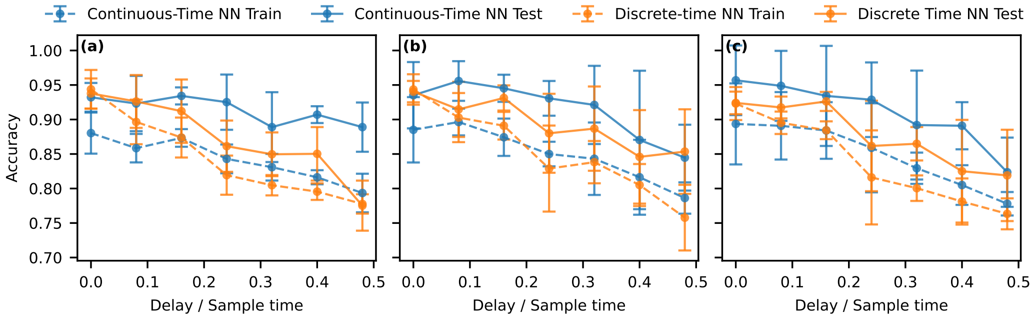

Figure 8: Delay ratio versus accuracy. Panels (a), (b), and (c) correspond to networks with 1, 2, and 3 hidden layers. Our continuous-time neural networks with the layerwise error propagation topology demonstrate comparable, if not slightly better, robustness to mismatch than corresponding discrete-time neural networks.