Stabilizing Continuous-Time Kolen–Pollack Learning with a Scale-Balance Condition

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2026

Abstract

Kolen–Pollack (KP) is a candidate biologically plausible learning rule whose continuous-time formulation eliminates phase separation but suffers from instability in deep networks. Existing discrete-time KP implementations rely on engineered stabilizers (batch normalization, regularization, gradient clipping, and adaptive optimizers) that do not translate directly to continuous-time biological or analog substrates. We analyze continuous-time KP through heterosynaptic plasticity (HSP) theory and identify a local scale-balance condition between the plasticity drive and the decay term that KP's alignment mechanism does not by itself enforce. We show that bounded activations restore this balance, and matches or exceeds layer-local normalization performance on 5 and 10 layer MLPs.

1. Introduction

Kolen–Pollack [3] is a learning rule for neural networks in which the forward and feedback weights are driven by paired Hebbian-plus-decay updates that contract the feedback path toward the transpose of the forward path. It is biologically attractive because these networks can be trained without the weight transport of vanilla backpropagation [1]. Standard formulations of KP, however, retain the discrete forward/backward phase separation of backpropagation, which is hard to reconcile with the continuous, asynchronous dynamics of a real neural circuit or an analog substrate. Bacvanski et al. [2] recently introduced a continuous-time formulation of the rule in which forward activities, backward errors, forward weights, and feedback weights evolve on time constants, eliminating phase separation and explicitly modeling the finite signal-propagation delays of a physical implementation.

Continuous-time KP is a natural model for a biological or analog account of credit assignment, but it exhibits significant instability with deeper networks. In prior implementations, instability is typically mitigated by engineered mechanisms including batch normalization, regularization, gradient clipping, optimizer choices, and learning-rate schedules [1, 4, 5, 7]. Each of those mechanisms relies on nonlocality in time and/or space, and is therefore not naturally available to a continuous-time substrate that processes a temporal sequence of inputs.

Using the lens of heterosynaptic plasticity (HSP) theory [8], we show that, while KP causes the forward and feedback weights to converge exponentially, it does not constrain the scale of the weights. The weight scale determines whether perturbations decay or amplify through the same-layer feedback loop, and deep-network stability rests on a local balance condition between the plasticity drive and the decay term. We propose using bounded activations to satisfy this condition, and we show that it matches or exceeds layer-local normalization on MLPs.

2. Continuous-time Kolen–Pollack learning

The continuous-time model of KP [2] replaces the settled forward and backward computations of the discrete rule with first-order dynamics on every state. Each state relaxes via a first-order low-pass filter, modelling membrane RC integration in a biological circuit or capacitive nodes in an analog substrate. The model has four state families, each with its own time constant:

1. forward activities $y_l(t) \in \mathbb{R}^{n_l}$ for $l = 0, \dots, L$, with $y_0 = x(t)$ the input;
2. backward error signals $z_l(t) \in \mathbb{R}^{n_l}$ for $l = 1, \dots, L$;
3. forward weights $W_l(t) \in \mathbb{R}^{n_l \times n_{l-1}}$ for $l = 1, \dots, L$;
4. feedback weights $V_l(t) \in \mathbb{R}^{n_l \times n_{l+1}}$ for $l = 1, \dots, L-1$, intended to approach $W_{l+1}^\top$.

Signal-propagation dynamics. Forward activities relax toward the layerwise nonlinearity applied to the preactivation on a time constant τ_{inf}:

$$\tau_{\text{inf}}\, \dot{y}_l \;=\; \phi_l(s_l) - y_l, \qquad s_l \;=\; W_l\, y_{l-1}, \tag{1}$$

where ϕ_l is a nonlinearity (we use ReLU). Backward error signals relax on a time constant τ_{err} toward the gated feedback signal:

$$\tau_{\text{err}}\, \dot{z}_l \;=\; \phi_l'(s_l) \odot \big(V_l\, z_{l+1}\big) - z_l, \tag{2}$$

with $V_l\, z_{l+1}$ replaced at the output layer by the loss derivative.

Plasticity dynamics. Each weight is driven by the outer product of its pre- and post-synaptic activities (Hebbian potentiation, on time constant τ_{pot}) minus a linear decay (on time constant τ_{dec}):

$$\tau_{\text{pot}}\, \dot{W}_l \;=\; z_l\, y_{l-1}^\top \;-\; \tfrac{\tau_{\text{pot}}}{\tau_{\text{dec}}}\, W_l, \qquad \tau_{\text{pot}}\, \dot{V}_l \;=\; y_l\, z_{l+1}^\top \;-\; \tfrac{\tau_{\text{pot}}}{\tau_{\text{dec}}}\, V_l, \tag{3}$$

matching the discrete KP rule. The four time constants satisfy

$$\tau_{\text{inf}}, \tau_{\text{err}} \;\ll\; T_{\text{sample}} \;\ll\; \tau_{\text{pot}} \;\ll\; \tau_{\text{dec}}, \tag{4}$$

with $\tau_{\text{inf}} = \tau_{\text{err}} = 1\,\text{ms}$, $T_{\text{sample}} = 50\,\text{ms}$, $\tau_{\text{pot}} = 10\,\text{s}$, $\tau_{\text{dec}} = 10^3\,\text{s}$. Each example is held constant for T_{sample} with a smooth ramp between examples, and the four state families evolve according to Eqs. (3) throughout, without phase separation.

3. Stability of KP in continuous time

KP supplies a local mechanism by which the feedback path becomes aligned with the transpose of the forward path, but it does not by itself control the scale of either weight matrix. For the reciprocal pair (W_{l+1}, V_l) the forward-weight update and the transpose of the feedback-weight update (3) share the same potentiation term $z_{l+1}\, y_l^\top$, since both rules read the same pre- and post-synaptic signals on the reciprocal connection. Decomposing the pair into the transpose mismatch $\Delta_l := V_l^\top - W_{l+1}$ and the weight scale $A_l := \frac{1}{2}(V_l^\top + W_{l+1})$, subtracting and adding the two equations gives

$$\dot{\Delta}_l \;=\; -\tfrac{1}{\tau_{\text{dec}}}\, \Delta_l, \qquad \tau_{\text{pot}}\, \dot{A}_l \;=\; z_{l+1}\, y_l^\top \;-\; \tfrac{\tau_{\text{pot}}}{\tau_{\text{dec}}}\, A_l. \tag{5}$$

The mismatch contracts exponentially on timescale τ_{dec}, independent of the activity, error signals, input distribution, and network depth; this is KP's central success. The weight scale, however, obeys the same Hebbian-plus-decay form as a single weight matrix and has no equivalent contraction: whether A_l remains bounded depends on the local plasticity drive $z_{l+1}\, y_l^\top$ remaining comparable to $(\tau_{\text{pot}}/\tau_{\text{dec}})\, \|A_l\|$, a condition not enforced by KP.

3.1. The HSP scale-balance condition

Heterosynaptic plasticity (HSP) theory [8] states that a two-signal rule $\dot{\bar{V}} = \eta\, p\, h^\top - \gamma\, \bar{V}$ behaves as a gradient-like learning rule when the local update is near balance, $\eta\, p\, h^\top \approx \gamma\, \bar{V}$. Identifying $\eta_l = 1/\tau_{\text{pot}}$ and $\gamma_l = 1/\tau_{\text{dec}}$, KP's forward-weight update (3) fits this template with $p = z_l$, $h = y_{l-1}$, $\bar{V} = W_l$, and the corresponding HSP balance condition is

$$\eta_l\, z_l\, y_{l-1}^\top \;\approx\; \gamma_l\, W_l. \tag{6}$$

This is the scale condition that the weight-scale dynamics in Eq. (5) can violate: KP can drive $\Delta_l \to 0$ while Eq. (6) fails, because the local plasticity drive and the weight scale grow together. Vanilla continuous-time KP solves the alignment component but does not satisfy the stability requirement for a heterosynaptic two-signal rule to behave as a gradient-like learning rule.

3.2. Depth and continuous time amplify the imbalance

In the aligned limit, $z_{l+1}\, y_l^\top$ is the standard backpropagated loss gradient with respect to W_{l+1}, so the classical exploding/vanishing-gradient analysis [6] applies: for a depth-L ReLU chain with layerwise gain A, $\|z_{l+1}\, y_l^\top\| \sim A^{L-1}$, overwhelming the linear decay term in $\dot{A}_l$. Continuous-time KP additionally closes the depth-amplifying loop within a single layer: a perturbation δW_l enlarges y_l, the Hebbian update enlarges V_l, which amplifies z_l, which feeds back into δW_l. In chain-rule backpropagation the analogous loop runs through the entire downstream network.

3.3. Empirical depth instability of vanilla continuous-time KP

Figure 1 illustrates the depth-amplified instability of continuous-time KP networks trained on UCI digits-8×8. The training curves demonstrate that even as $V^\top$ aligns to W_{l+1} (guaranteed by KP), the scale of the weights A_l can grow without bound, causing the network to diverge. At $L = 2$ the dynamics are well-behaved: test accuracy rises smoothly to ≈ 0.96, test loss decays monotonically, and the per-layer weight scale remains bounded. At $L = 5$ training initially climbs to ≈ 0.80 accuracy, but per-layer weight scales begin to grow around $\sim$3,000 samples; the run then diverges as the weights explode. At $L = 8$ the failure happens earlier and harder: the weight scale grows immediately, and the run diverges within the first few thousand samples.

4. Satisfying the scale-balance condition

Practical KP-like and asymmetric-feedback implementations [1, 4, 5, 7] consistently pair the KP alignment mechanism with BatchNorm, weight decay, feedback-norm clipping, and/or a separate (slower) feedback learning rate. The scale-balance condition of Eq. (6) explains why: KP supplies alignment but leaves the LHS unconstrained, so each implementation has had to use an external scale regulator to satisfy it. None of these techniques transfer naturally to a continuous-time substrate.

Taking norms of Eq. (6) gives the scalar consequence $\eta_l\, \|z_l\|\, \|y_{l-1}\| \approx \gamma_l\, \|W_l\|$, which exposes two routes to local control: (a) bound the LHS by bounding the activity and error magnitudes, or (b) bound the RHS by regulating the weight magnitude directly. Route (a) is the more direct intervention because the relevant quantities are the signals already propagating through the layer at run time. The simplest neuron-local way to bound $\|y_l\|$ is to bound each $y_{l,i}$ individually by reshaping the activation function. We replace the ReLU activation at every hidden neuron by its clipped variant

$$\phi_l^{\text{sat}}(s) \;=\; \text{clip}\big(\phi_l(s),\, -c,\, +c\big), \tag{7}$$

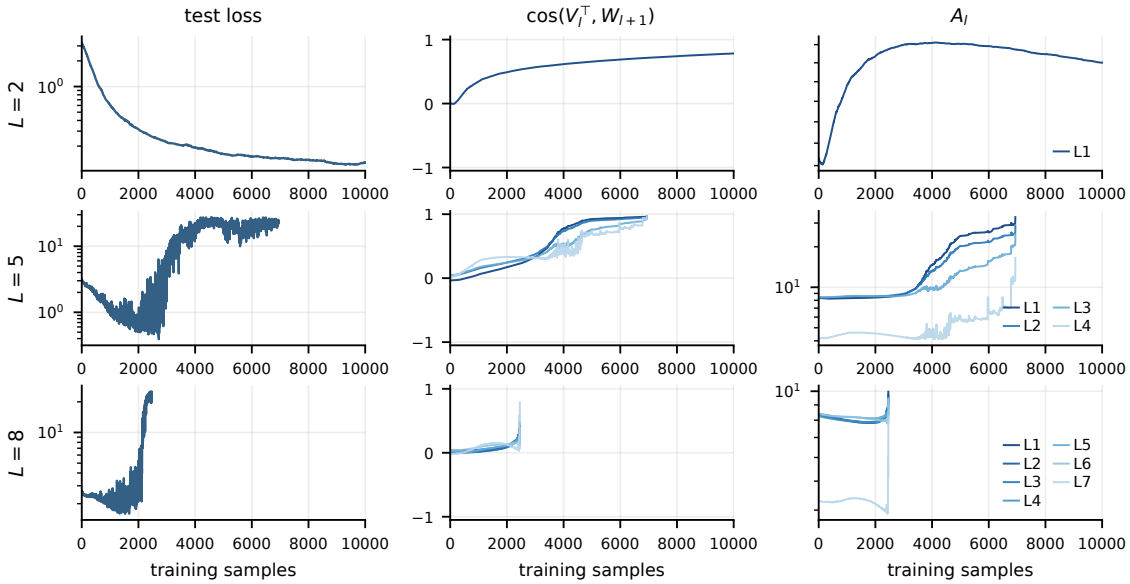


Figure 1: Continuous-time KP trained on UCI digits-8×8 exhibits instability dependent on network depth. Rows are network depths $L = 2, 5, 8$. Depth 2 trains stably; depth 5 trains briefly to ≈ 0.8 accuracy then diverges as per-layer weight scales grow without bound; depth 8 never escapes chance and diverges soon after the first few thousand samples.

with cap c shared across the network ($c = 1$ in our experiments), and use ϕ_l^{sat} in place of ϕ_l throughout Eqs. (1)–(2). With $|y_{l,i}| \le c$ at every hidden neuron, $\|y_l\|_2 \le c\sqrt{n_l}$. The backward derivative is gated by saturation: when $|y_{l,i}|$ is at the cap the activation derivative is zero, so the corresponding entries of $V_l\, z_{l+1}$ in Eq. (2) are killed, damping $\|z_l\|$ as well.

4.1. Bounded activations matches or beats RMSNorm at $L = 5, 10$

Figure 2 compares the two mechanisms on continuous-time KP trained on MNIST at depths $L = 5$ and $L = 10$, with the timescales of Eq. (4), Xavier initialization for both forward and feedback weights, two MNIST epochs (120,000 samples), and 100 seeds per condition. At $L = 5$, networks with bounded activations reach a final test accuracy of 0.941 ± 0.009 across seeds, comparable to a discrete-time SGD baseline of the same depth; RMSNorm exhibits much greater variance and significantly lower accuracies. At $L = 10$ both mechanisms underperform discrete-time SGD and exhibit large seed-to-seed variability, but bounded activations produces some seeds reaching as high as 0.73, while all RMSNorm networks plateau in the 0.2–0.4 range. Both mechanisms drive cosine alignment $\cos(V_l^\top, W_{l+1})$ to essentially 1 within roughly 25,000 samples and produce matched $\|W\|_F \approx \|V\|_F$ throughout training, in agreement with the contraction guarantee of §3.

5. Discussion

Our HSP scale-balance lens of §3.1 provides the missing condition for stability that KP alone does not enforce. The mechanisms found in KP-like implementations in the literature (batch normalization, weight decay, gradient clipping, adaptive optimizers) are not just training conveniences, but load-bearing mechanisms that satisfy the local balance condition Eq. (6). None of these mechanisms

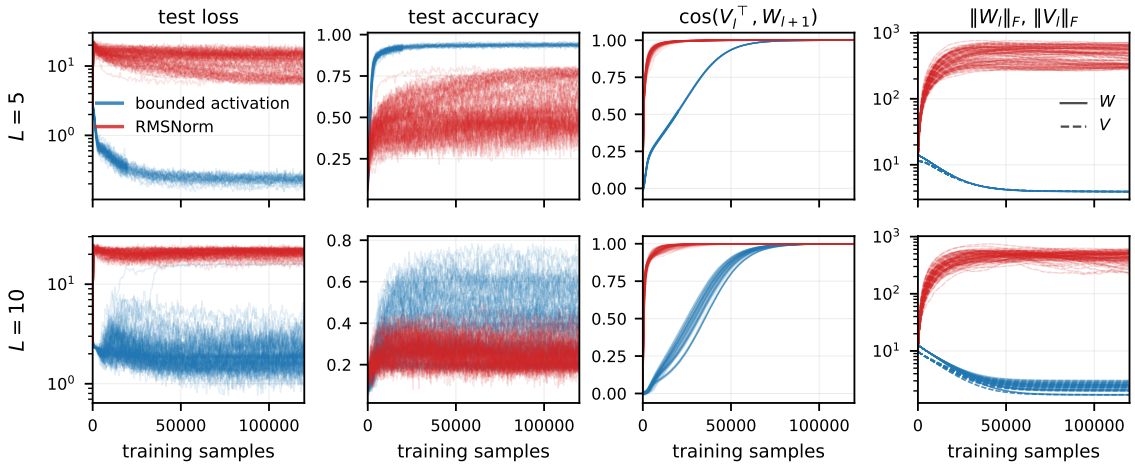


Figure 2: Continuous-time KP networks trained on MNIST with 5 hidden layers (top) and 10 hidden layers (bottom). Saturating activation networks (ϕ_l^{sat} at $c = 1.0$) are in blue and networks with RMSNorm ($\alpha = 0.5$) are in red; 100 seeds per network. At $L = 5$ per-neuron saturation reaches near-SGD accuracy (0.94), unlike RMSNorm despite also achieving perfect feedback alignment; at $L = 10$ per-neuron saturation results in some good seeds achieving high accuracies, while all RMSNorm networks plateau in the 0.2–0.4 range with near perfect alignment and stable W, V scale.

transfer naturally to a continuous-time substrate. We show that bounding hidden neuron activations is an alternative approach to satisfying the stability condition of KP, and trains continuous-time KP networks to near-SGD accuracy at $L = 5$ and to $L = 10$ accuracies that beat layer-local normalization on MNIST. Bounded activations are moreover native to both real neurons (firing-rate ceilings set by refractory dynamics) and analog hardware.

Our results suggest a remaining open question. At $L = 10$ per-neuron saturation produces a wide distribution of test accuracies across the 100-seed sweep, with the best seed reaching 0.73 and the median plateauing near 0.37, while at $L = 5$ the same mechanism is essentially deterministic across seeds. The maximum accuracy across seeds at each depth quantifies how strongly the accumulated signal-propagation lag through a deeper network bounds what is achievable [2]; what determines whether a given seed reaches that ceiling or stalls near chance is not captured by either the scale-balance condition or by the classical depth instability.

References

[1] Mohamed Akrout, Collin Wilson, Peter C. Humphreys, Timothy Lillicrap, and Douglas B. Tweed. Deep learning without weight transport. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1904.05391.

[2] Marc Gong Bacvanski, Liu Ziyin, and Tomaso Poggio. Does feedback alignment work at biological timescales?, 2026. URL https://arxiv.org/abs/2510.18808.

[3] John F. Kolen and Jordan B. Pollack. Backpropagation without weight transport. In *Proceedings of 1994 IEEE International Conference on Neural Networks*, volume 3, pages 1375–1380, 1994. doi: 10.1109/ICNN.1994.374486.

[4] Qianli Liao, Joel Z. Leibo, and Tomaso Poggio. How important is weight symmetry in backpropagation? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, pages 1837–1844, 2016. doi: 10.1609/aaai.v30i1.10279.

[5] Theodore H. Moskovitz, Ashok Litwin-Kumar, and L. F. Abbott. Feedback alignment in deep convolutional networks, 2019. URL https://arxiv.org/abs/1812.06488.

[6] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1310–1318, 2013. URL https://proceedings.mlr.press/v28/pascanu13.html.

[7] Matthew Bailey Webster, Jonghyun Choi, and Changwook Ahn. Learning the connections in direct feedback alignment, 2020. URL https://openreview.net/forum?id=zgGmAx9ZcY.

[8] Liu Ziyin, Isaac Chuang, and Tomaso Poggio. Heterosynaptic circuits are universal gradient machines, 2025. URL https://arxiv.org/abs/2505.02248.

Appendix A. Adiabatic reduction in the aligned limit

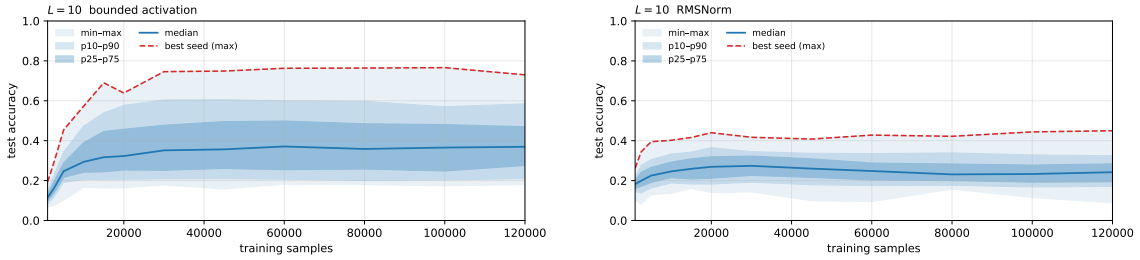
The hierarchy (4) permits a clean adiabatic reduction. On the plasticity timescale τ_{pot}, the forward and backward fast variables y_l and z_l have already equilibrated to their fixed-input fixed-weight slow manifolds, since $\tau_{\mathrm{inf}}, \tau_{\mathrm{err}}$ are five orders of magnitude smaller. Setting $\dot{y}_l = \dot{z}_l = 0$ in Eqs. (1)–(2) for fixed $\{W_l, V_l\}$ gives the slow-manifold values

$$y_l^* \;=\; \phi_l\big(W_l\, y_{l-1}^*\big), \qquad z_l^* \;=\; \phi_l'(s_l^*) \odot \big(V_l\, z_{l+1}^*\big), \qquad z_L^* \;=\; y^* - \mathrm{softmax}(s_L^*). \tag{8}$$

These are exactly the discrete-time forward and backward recursions with feedback matrix V_l replacing $W_{l+1}^\top$. If feedback alignment has been achieved, $V_l = W_{l+1}^\top$ for all l, and Eq. (8) reduces to the ordinary backpropagation recursion

$$z_l^* \;=\; \phi_l'(s_l^*) \odot W_{l+1}^\top\, z_{l+1}^*. \tag{9}$$

Substituting the slow-manifold values into the plasticity equations (3) and averaging over the sample-window time scale T_{sample} (still much shorter than τ_{pot}), the effective forward-weight update is the average of $z_l^* y_{l-1}^{*\top}$ over the data distribution, minus the slow weight-decay term. This is the standard backpropagation-with-weight-decay update, up to sign conventions and the plasticity-rate prefactor $\eta_l = 1/\tau_{\mathrm{pot}}$. KP additionally contracts the transpose mismatch $V_l^\top - W_{l+1}$ on the decay timescale, providing a local mechanism for the aligned limit to be reached. We emphasize that this reduction is conditional on the timescale separation (4) actually holding during training, which itself depends on the plasticity drive remaining bounded against the decay term.



(*a*) Per-neuron saturation. (*b*) RMSnorm.

Figure 3: Test-accuracy percentile envelopes across 100 seeds at $L = 10$ for the two scale-control mechanisms, vs. training samples. Bands are the min–max, p_{10}–p_{90}, and p_{25}–p_{75} envelopes; solid blue is the median; dashed red is the per-sample maximum. For per-neuron saturation (a) the maximum saturates near 0.76 by 60,000 samples and the median plateaus near 0.37; the distribution shape is set by $\sim$60,000 samples and not narrowed by another 60,000. RMSnorm (b) shows a substantially narrower distribution stuck at much lower accuracy.

	min	p_{10}	p_{25}	median	p_{75}	p_{90}	max
$L=5$ @ 20k samples	0.900	0.912	0.915	0.923	0.928	0.935	0.943
$L=5$ @ 120k samples	0.921	0.930	0.933	0.936	0.943	0.949	0.959
$L=10$ @ 20k samples	0.164	0.207	0.253	0.323	0.457	0.577	0.639
$L=10$ @ 120k samples	0.180	0.209	0.276	0.369	0.470	0.584	0.730

Table 1: Test-accuracy percentiles across the seed-exploration sweep with per-neuron saturation. $L = 5$ is essentially deterministic across seeds. At $L = 10$ the distribution is broad with an architectural ceiling near 0.73.

Appendix B. Seed distribution at $L = 10$.

The high seed variance for per-neuron saturation at $L = 10$ in Figure 2 reflects a wide underlying distribution over initial conditions. To characterize it we ran an additional sweep of 100 seeds at each of $L = 5, 10$ trained for 120,000 samples. Figure 3 shows the resulting test-accuracy distribution at $L = 10$: at 20,000 samples the seeds are spread over $[0.16, 0.64]$, the spread continues to widen until about 60,000 samples, after which the distribution shape (and in particular the maximum) stops changing materially.

The percentile summary at end of training is given in Table 1. We report it instead of a mean $\pm$ std because the underlying distribution is not Gaussian: at $L = 10$ it is broad and roughly flat over $[0.2, 0.7]$, with a clear architectural ceiling near 0.73 that no seed exceeds. The right reading of the $L = 10$ plateau is akin to a basin-finding probability rather than a mean accuracy: the network reliably learns when initialized in a small fraction of the seed space and fails on the rest, and the fraction is fixed early in training. By contrast at $L = 5$ the distribution is essentially deterministic (0.92–0.94 across all 77 seeds at 20,000 samples).

Appendix C. Common failure modes of KP at depth

Failure mode 1: alignment without training. The weight-scale dynamics of §3 already show that vanilla continuous-time KP can drive $\Delta_l \to 0$ while A_l blows up: the mismatch contracts and the weight scale runs away simultaneously. The loss panel of Figure 2 shows a milder version of the same phenomenon for RMSNorm: $\cos(V_l^\top, W_{l+1}) \to 1$ within a few thousand samples and stays there, but the test loss does not drop below ≈ 5. Applying RMSNorm to z_L at the output layer rescales the gradient signal away from its loss-derived magnitude $y^* - \mathrm{softmax}(s_L)$, replacing it by a unit-RMS direction with the wrong absolute scale. The feedback path is correctly aligned with the forward transpose, but the propagated error no longer carries useful magnitude information, so the gradient-like update has the right direction but the wrong size. Per-neuron saturation does not modify z_L at all and does not suffer this pathology. The general lesson is that the alignment cosine $\cos(V_l^\top, W_{l+1})$ is necessary but not sufficient as a diagnostic of training progress; one must also check that the magnitude of the propagated error is doing useful work.

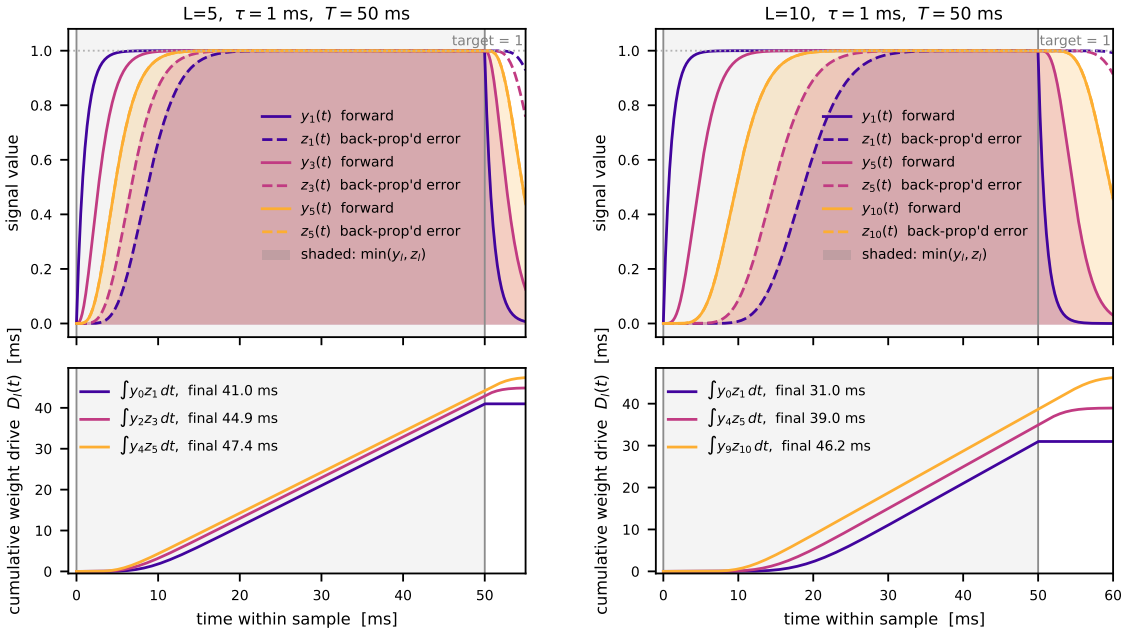
Failure mode 2: cascade-lag signal starvation. Every layer in the model is a first-order low-pass filter (LPF) on the inference timescale τ_{inf}, and the depth-L network is therefore a chain of L such LPFs in series for the forward signal and another L in series for the back-propagated error. The cumulative weight drive at layer l over a sample window of length T_{sample} is the convolution

$$D_l \;=\; \int_0^{T_{\mathrm{sample}}} y_{l-1}(t)\, z_l(t)\, dt \tag{10}$$

of the forward signal that has propagated through $l-1$ LPFs with the back-propagated error that has propagated through $L-l$ LPFs. The two cascades are temporally offset by approximately $(L-1)\,\tau_{\mathrm{inf}}$ in the forward direction and the same on the backward side, so as $L\,\tau_{\mathrm{inf}}$ approaches $T_{\mathrm{sample}}/2$ the input-side overlap window collapses and D_l at small l goes to zero. This is a structural property of the cascade and depends on L and $\tau_{\mathrm{inf}}/T_{\mathrm{sample}}$ but not on the choice of scale-control mechanism.

Figure 4 renders the cascade-lag effect for $L = 5$ and $L = 10$. At $L = 5$ the input-side and output-side cumulative drives are comparable in magnitude ($D_1 \approx 41\,\mathrm{ms}$, $D_5 \approx 47\,\mathrm{ms}$ over the $50\,\mathrm{ms}$ sample window), and the network is in a regime where both forward and backward signals reach every layer well within the sample. At $L = 10$ the input-side drive has dropped to $D_1 \approx 31\,\mathrm{ms}$ while the output-side stays at $D_{10} \approx 46\,\mathrm{ms}$. The asymmetry will continue to widen with L until the input-side drive collapses to zero, at which point no plasticity update accumulates at the input layer regardless of the alignment cosine or the satisfaction of Eq. (6).

Putting the failure modes together. The $L = 10$ plateau in Figures 2 and 3 reflects both effects. RMSNorm suffers failure 1 across all depths and additionally hits failure 2 at $L = 10$, hence the very high loss in the bottom row. Per-neuron saturation avoids failure 1 (its loss does drop) but is bottlenecked by failure 2 at $L = 10$, which is why its accuracy plateaus with an architectural ceiling near 0.73 and a median around 0.37 even though the within-layer scale balance and the cosine alignment are both satisfied. The wide seed distribution in Figure 3 is consistent with this picture: when the cumulative input-side drive D_1 is small, the input-layer plasticity update is dominated by whatever signal does survive the cascade, and which seeds escape into a useful basin depends on the random V initialisation. Neither failure is a property of the scale-control mechanism itself; both are structural to the continuous-time KP cascade at depth.



(*a*) $L = 5$. (*b*) $L = 10$.

Figure 4: Cascade-lag overlap of forward $y_l(t)$ (solid) and back-propagated error $z_l(t)$ (dashed) for representative layers, plus the cumulative weight drive $D_l(t) = \int_0^t y_{l-1}(s)\, z_l(s)\, ds$ (bottom subpanels). Top-hat input on $[0, T_{\text{sample}}]$ with $T_{\text{sample}} = 50\,\text{ms}$ and $\tau_{\text{inf}} = 1\,\text{ms}$. At $L = 5$ the input-side cumulative drive ($\ell = 1$) reaches $\sim 41\,\text{ms}$ by end of sample, close to the $50\,\text{ms}$ ideal; at $L = 10$ it has dropped to $\sim 31\,\text{ms}$. The shrinking input-side overlap window caps trainable accuracy at depth independently of the scale-control mechanism.