ALIGNMENT IS NOT LEARNING: LOOP-GAIN CONTROL IN HETEROSYNAPTIC PLASTIC CIRCUITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Heterosynaptic plasticity (HSP) rules combine local activity-dependent potentiation with compensatory decay, providing a biologically and physically plausible substrate for learning. However, in a learning network the local plasticity rule is embedded in a closed loop: perturbations of plastic weights alter neural signals, which in turn alter the plasticity drive. This suggests that stability of networks built with these rules is not guaranteed. Linearizing an HSP motif around a balanced operating point yields a loop-gain condition: stability depends on the closed-loop sensitivity from weights back to their local learning signals. We apply this analysis to continuous-time Kolen–Pollack learning, a well-known HSP-style mechanism for aligning feedback weights with forward weights. Our analysis shows that while KP's symmetric updates exponentially contract the forward–feedback weight mismatch, they critically leave a weight scale mode governed by the generic HSP loop-gain spectrum. Thus feedback alignment is not sufficient for stability. Finally, we show that bounded activation functions limit the effective plastic loop gain and stabilize continuous-time KP learning in deep multilayer perceptrons, outperforming the alternative local mechanisms we test at every depth and enabling training at depths where they fail. Across mechanisms, stability follows loop-gain control: a circuit whose loop gain is left uncontrolled fails, by fast divergence or by unbounded weight drift, even when its feedback weights align and its update magnitudes are bounded. These results identify loop-gain control as a central requirement for stable heterosynaptic learning in biological and continuous-time substrates.

1 INTRODUCTION

Heterosynaptic plasticity (HSP) offers a general local mechanism for learning in biological and physical systems. HSP rules are two-factor local learning rules whose motif can be tiled into many circuit topologies, allowing plastic change to be driven by signals carried elsewhere in the circuit; in many cases gradient-based learning emerges from their interaction rather than being explicitly programmed into the local rule (Ziyin et al., 2025). This makes HSP-style rules an attractive substrate for credit assignment: local weight updates depend only on nearby neural or circuit variables, while the network as a whole implements learning dynamics that resemble gradient-based optimization.

Heterosynaptic plasticity rules are often analyzed through their local update equations, but in a learning network those equations are embedded in a closed loop (Figure 1). A change in a plastic weight changes neural activity, neural activity changes the local plasticity signals, and those altered signals feed back into the same plastic weight. Thus, for stability, the relevant object is the return path created by the surrounding circuit. A topology that implements useful gradient-like learning can also create feedback loops that amplify perturbations. Understanding stable HSP learning therefore requires a closed-loop stability analysis, not just a local balance condition.

In this work, we develop such an analysis. We first study the linear stability of a single HSP motif near a balanced operating point and show that balance between potentiation and decay is not by itself sufficient in the presence of feedback. Stability also depends on the sensitivity of the local learning signals to perturbations of the plastic weights. This gives an architecture-level loop-gain condition for

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

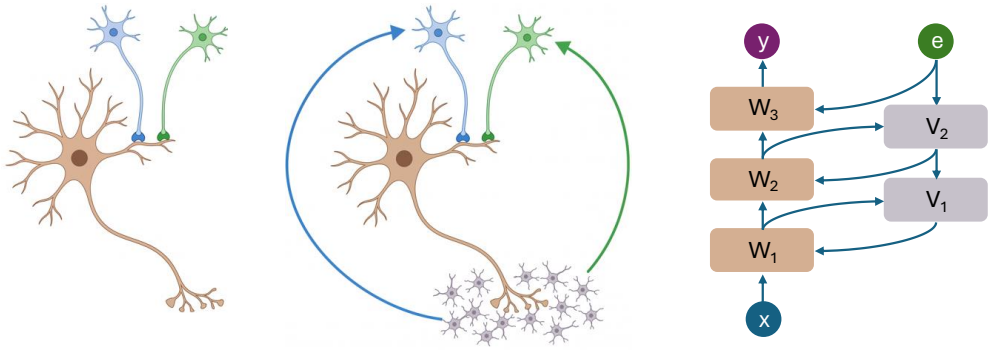


Figure 1: **From a heterosynaptic motif to a closed-loop learning circuit.** *Left:* a heterosynaptic plasticity (HSP) motif: two presynaptic inputs enter a postsynaptic neuron, whose synapse is modified purely by local activity. *Middle:* when embedded in a network, the HSP motif sits in a closed loop: the neuron's output propagates through the surrounding circuit and returns to drive the very synapses being updated. *Right:* Kolen–Pollack learning is a structured form of HSP circuit: a forward pathway of weights W_l carries activity from input x to output y, and a feedback pathway of weights V_l carry the error e backward. Activations and error signals that drive inference and weight updates, denoted by arrows, form feedback loops.

HSP stability: stable learning requires the loop gain from plastic weights back to their own learning signals to remain controlled.

We use continuous-time Kolen–Pollack (KP) learning as a concrete case study. KP is a biologically plausible learning rule in which forward and feedback weights are driven by paired Hebbian-plus-decay updates that align the feedback path with the transpose of the forward path (Kolen & Pollack, 1994; Akrout et al., 2019) (Figure 1, right). Its update rules make it a natural example of an HSP-style learning circuit. We analyze its continuous-time formulation (Bacvanski et al., 2025), in which activities, errors, and both weight families evolve on their own time constants without the discrete forward and backward phases of standard KP (§4).

Continuous-time KP exposes the distinction between alignment and stability. The KP updates locally align reciprocal forward and feedback weights, and this alignment mechanism remains exact in continuous time. Nevertheless, deep continuous-time KP networks tend to become unstable. Prior implementations of KP-like and asymmetric-feedback systems usually avoid this problem through engineered stabilizers such as batch normalization, regularization, gradient clipping, optimizer choices, and learning-rate schedules (Liao et al., 2016; Akrout et al., 2019; Moskovitz et al., 2018; Webster et al., 2020; Kunin et al., 2020). These stabilizing mechanisms are effective in conventional machine-learning implementations, but they rely on information that is nonlocal in space (aggregated across the units of a layer or the examples of a batch) or nonlocal in time (globally synchronized update phases, or statistics, optimizer state, and schedules computed over many past examples). Locality in space is the standard requirement for a biologically plausible learning rule (Crick, 1989; Whittington & Bogacz, 2019; Lillicrap et al., 2020); a continuous-time substrate processing a temporal stream of inputs imposes locality in time as well: because information flows at finite speed, an update cannot depend on quantities that have not yet reached the synapse (Salami et al., 2003; Debanne et al., 2011; Bengio et al., 2015; Bacvanski et al., 2025). Mechanisms that violate either form of locality do not translate directly to continuous-time biological or analog substrates; for a truly biologically plausible learning rule, a mitigation local in both senses is essential.

This perspective suggests a local route to stabilization. Instead of importing nonlocal normalization or optimizer mechanisms, one can reduce the effective plastic loop gain by bounding the neural signals that enter the local update. We study bounded activations as a substrate-compatible gain-control mechanism for continuous-time KP. Bounded activations cap forward activity and gate backward error propagation through saturation, reducing the magnitude and sensitivity of the local plasticity drive. We find that this empirically stabilizes continuous-time KP in deep multilayer perceptrons and outperforms the alternative local stabilization mechanisms we test, enabling training at depths where

they fail. Together, these results identify loop-gain control as a central design requirement for stable HSP-based learning in biological and continuous-time physical substrates.

2 Related Work

The weight-transport problem, namely that backpropagation requires the feedback pathway to copy the forward weights, has long been a central objection to backpropagation as a model of learning in the brain (Grossberg, 1987; Crick, 1989). Feedback alignment showed that fixed random feedback weights can suffice, with the forward weights aligning to them during training (Lillicrap et al., 2016); direct feedback alignment broadcasts the output error to every layer (Nøkland, 2016; Launay et al., 2019); sign-symmetric feedback scales to harder tasks (Liao et al., 2016; Xiao et al., 2019); and weight mirrors and the Kolen–Pollack rule learn the feedback weights themselves (Akrout et al., 2019; Kolen & Pollack, 1994). Theory in this line asks when alignment emerges (Refinetti et al., 2021) and benchmarks ask whether it scales (Bartunov et al., 2018); in all of it, stability is obtained through the engineered mechanisms discussed above. Closest to our work, Kunin et al. (2020) show that the weight-mirror rule is destabilized by growth of a common (symmetric) weight mode rather than by misalignment, and counteract it with regularization. We show that this is one instance of a general principle, the loop-gain condition derived below for any heterosynaptically plastic circuit, and stabilize continuous-time KP with a strictly local mechanism rather than with regularization.

That Hebbian growth is unstable without compensation is a classical theme in neuroscience: Oja's rule (Oja, 1982) and the BCM rule (Bienenstock et al., 1982) stabilize the single neuron, synaptic scaling regulates activity over hours to days (Turrigiano & Nelson, 2004; Turrigiano, 2008), and measured homeostasis is far slower than Hebbian plasticity (Zenke et al., 2017). These analyses concern the local rule or the single neuron; ours is at the level of the circuit, where the plasticity drive is itself a function of the weights through the surrounding network, stability is a property of the closed loop, and no motif-level balance condition can guarantee it.

3 Linear Stability of Heterosynaptic Plastic Circuits

The simplest heterosynaptic plasticity motif is the local update rule of a single weight

$$\dot{W} = \eta a b^\top - \gamma W \tag{1}$$

where $W \in \mathbb{R}^{m\times n}$, $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$ are two local signals, $\eta > 0$ is the potentiation rate, and $\gamma > 0$ is the decay rate. A balanced operating point, denoted throughout by starred quantities, satisfies

$$\eta a^* b^{*\top} \approx \gamma W^* \tag{2}$$

We can analyze the effect of a small perturbation around this point, such that $W = W^* + \delta W$, $a = a^* + \delta a$, and $b = b^* + \delta b$. Substituting into Equation 1 gives

$$\dot{W}^* + \delta\dot{W} = \eta(a^* + \delta a)(b^* + \delta b)^\top - \gamma(W^* + \delta W) \tag{3}$$

Expanding and dropping the second-order term $\eta\, \delta a\, \delta b^\top$ gives

$$\delta\dot{W} = -\gamma\delta W + \eta(\delta a\, b^{*\top} + a^*\, \delta b^\top) \tag{4}$$

If a and b are external signals, this motif is trivially stable since $\delta a = \delta b = 0$. This relation becomes interesting when the plasticity motif is embedded in a circuit that provides a return path for perturbations of W to affect a and b, as is the case in all learning networks. This means that $\delta W \to \delta a, \delta b \to \delta\dot{W}$. Concretely, we can express the first-order signal sensitivities as $\delta a = A[\delta W]$ and $\delta b = B[\delta W]$, where A and B are linear maps determined by the rest of the circuit. Substituting into Equation 4, we get

$$\delta\dot{W} = -\gamma\delta W + \eta\big(A[\delta W]\, b^{*\top} + a^*\, B[\delta W]^\top\big) \tag{5}$$

Defining the loop-gain operator $T[\delta W] := A[\delta W]\, b^{*\top} + a^*\, B[\delta W]^\top$, known in classical feedback theory as the return ratio (Åström & Murray, 2008), the perturbation dynamics can be written as

$$\delta\dot{W} = (-\gamma I + \eta T)[\delta W] \tag{6}$$

which is stable iff every eigenvalue λ_i of $-\gamma I + \eta T$ has a negative real part

$$\max_i \operatorname{Re}\lambda_i(-\gamma I + \eta T) < 0 \tag{7}$$

Architectural interpretation. The sensitivity maps A and B represent the linearization of the full closed-loop sensitivity of the circuit surrounding the plasticity rule. In a network with many HSP motifs, perturbing one plastic variable can alter all neural states and other plastic drives before returning to the motif being analyzed. Thus the same loop-gain condition applies motif-wise or to any chosen plastic subspace, with the rest of the architecture absorbed into A and B.

4 CONTINUOUS-TIME KOLEN–POLLACK LEARNING

Kolen–Pollack (KP) learning (Kolen & Pollack, 1994) drives reciprocal forward and feedback weights with paired Hebbian-plus-decay updates that contract the feedback path toward the transpose of the forward path, avoiding the weight transport of backpropagation (Ziyin et al., 2025; Akrout et al., 2019). Standard KP relies on discrete forward and backward phases, hard to reconcile with the continuous, asynchronous dynamics of biological circuits or analog substrates; continuous-time KP (Bacvanski et al., 2025) removes them by letting forward activities, backward errors, forward weights, and feedback weights evolve simultaneously on separate time constants.

In the continuous-time KP model, each state relaxes through first-order dynamics, modeling 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 and n_l the width of layer l;
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}:

$$\tau_{\text{inf}}\, \dot{y}_l \;=\; \phi_l(s_l) - y_l, \qquad s_l \;=\; W_l\, y_{l-1}, \tag{8}$$

where ϕ_l is a nonlinearity (we use leaky ReLU with slope $\alpha = 0.01$). Backward error signals relax on a time constant τ_{err} toward the gated feedback signal:

$$\tau_{\text{err}}\, \dot{z}_l \;=\; \phi_l'(s_l) \odot \big(V_l\, z_{l+1}\big) - z_l, \tag{9}$$

with $\odot$ the elementwise product, and $V_l\, z_{l+1}$ 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}):

$$\tau_{\text{pot}}\, \dot{W}_l \;=\; z_l\, y_{l-1}^\top \;-\; \tfrac{\tau_{\text{pot}}}{\tau_{\text{dec}}}\, W_l, \qquad \tau_{\text{pot}}\, \dot{V}_l \;=\; y_l\, z_{l+1}^\top \;-\; \tfrac{\tau_{\text{pot}}}{\tau_{\text{dec}}}\, V_l, \tag{10}$$

matching the discrete KP rule. The four time constants satisfy

$$\tau_{\text{inf}}, \tau_{\text{err}} \;\ll\; T_{\text{sample}} \;\ll\; \tau_{\text{pot}} \;\ll\; \tau_{\text{dec}}, \tag{11}$$

with $\tau_{\text{inf}} = \tau_{\text{err}} = 1\,\text{ms}$, $T_{\text{sample}} = 50\,\text{ms}$, $\tau_{\text{pot}} = 10\,\text{s}$, $\tau_{\text{dec}} = 10^3\,\text{s}$. These values are chosen to realize this separation rather than to fit a particular preparation: signals relax on the order of milliseconds, a stimulus is held for tens of milliseconds, Hebbian potentiation accumulates over seconds, and homeostatic regulation of synaptic strength is slower still by orders of magnitude (Turrigiano, 2008; Zenke et al., 2017). Each example is held constant for T_{sample} with a smooth ramp between examples, and the four state families evolve according to Eqs. (8), (9), and (10) throughout, driving both learning and inference without phase separation.

4.1 EMPIRICAL DEPTH INSTABILITY OF VANILLA CONTINUOUS-TIME KP

The continuous-time formulation of KP is a natural model for biological or analog credit assignment, but it becomes unstable in deeper networks. Prior KP-like and asymmetric-feedback implementations typically control such instabilities using engineered mechanisms such as batch normalization, regularization, gradient clipping, optimizer choices, and learning-rate schedules (Liao et al., 2016; Akrout et al., 2019; Moskovitz et al., 2018; Webster et al., 2020). These mechanisms rely on nonlocality in

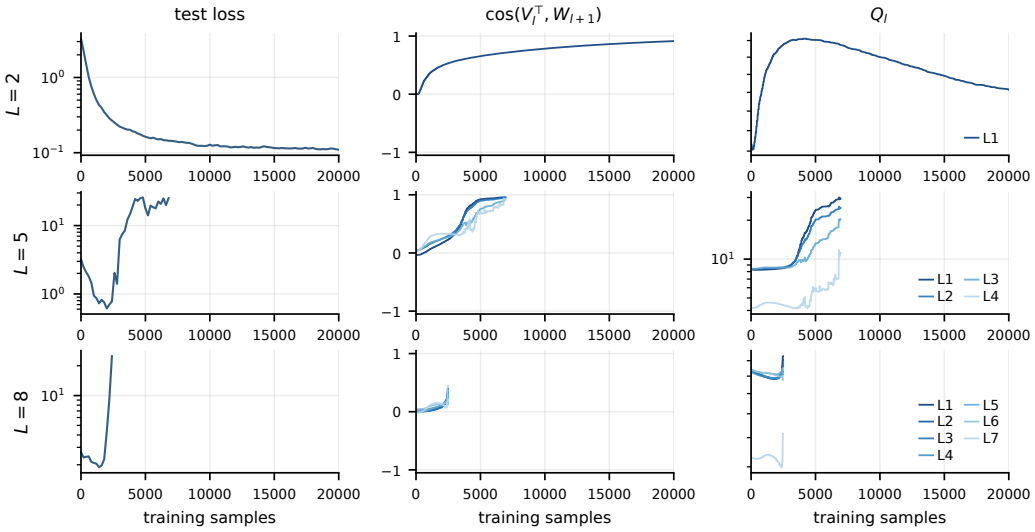


Figure 2: Continuous-time KP trained on UCI digits-8×8 exhibits instability dependent on network depth. Rows are network depths $L = 2, 5, 8$. Columns: test accuracy, test loss (log y), per-layer cosine similarity $\cos(V_l^\top, W_{l+1})$, and per-layer weight scale Q_l (log y); per-layer curves are shaded blue from darkest (input-side layer) to lightest (output-side layer).

time and/or space, and therefore do not transfer naturally to a continuous-time substrate processing a temporal sequence of inputs.

Figure 2 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 Q_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.

5 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 guarantee stability. Define the common coordinate Q and differential coordinate Δ as

$$Q_1 := W_1, \quad Q_{l+1} := \frac{1}{2}\big(V_l^\top + W_{l+1}\big), \quad \Delta_l := V_l^\top - W_{l+1} \tag{12}$$

for $l = 1, \ldots, L-1$. Subtracting and adding the plasticity equations for each reciprocal pair V_l and W_{l+1} gives

$$\dot{\Delta}_l \;=\; -\tfrac{1}{\tau_{\text{dec}}}\,\Delta_l, \qquad \tau_{\text{pot}}\,\dot{Q}_l \;=\; z_l\,y_{l-1}^\top \;-\; \tfrac{\tau_{\text{pot}}}{\tau_{\text{dec}}}\,Q_l. \tag{13}$$

where the second equation holds for $l = 1, \ldots, L$. The mismatch coordinates Δ contract exponentially on timescale τ_{dec}, independently of neural activity, error signals, input distribution, and network depth; this is KP's alignment guarantee. The common coordinate Q, however, has no such guarantee.

5.1 Loop-gain analysis of KP

We analyze the plasticity dynamics under timescale separation, assuming the forward and backward signals relax rapidly relative to the weights. Let

$$\overline{G}(Q, \Delta) = \big(\langle z_l y_{l-1}^\top \rangle\big)_{l=1}^{L} \tag{14}$$

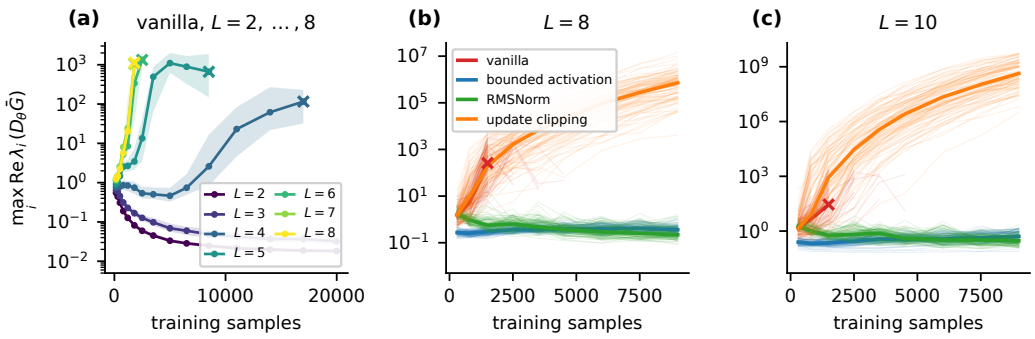


Figure 3: Loop-gain spectral abscissa $\max_i \mathrm{Re}\,\lambda_i(D_\theta \overline{G})$ along training on UCI digits-8×8 (Appendix B). (a) Vanilla continuous-time KP, $L = 2, \ldots, 8$, 100 seeds per depth: median and interquartile band, drawn while at least half the seeds remain finite; $\times$ marks where over half diverge. (b, c) $L = 8$ and $L = 10$, 100 seeds per condition (thin: per seed; thick: median over the still-finite seeds). Bounded activations (blue) and RMSNorm (green) hold the abscissa bounded; vanilla (red) runs away until the weights overflow ($\times$); update clipping (orange, c_u=1) shows the same uncontrolled growth of the gain while its weights stay finite. For clipping the abscissa is that of the pre-clip drive, since clipping rescales the update after the drive is formed; the cap on update speed, not the gain, is what keeps the weights finite.

denote the stacked KP outer-product drives, averaged over the fast signal dynamics and inputs while the plastic weights are held fixed. Since all mismatch perturbations are stable, we can consider the averaged dynamics of Q on the aligned manifold where $\Delta = 0$:

$$\tau_{\text{pot}} \dot{Q} = \overline{G}(Q, 0) - \frac{\tau_{\text{pot}}}{\tau_{\text{dec}}} Q \tag{15}$$

Let Q^* be an equilibrium on this manifold, and define $D_Q \overline{G}_* = D_Q \overline{G}(Q^*, 0)$, the Jacobian of the averaged drive with respect to the common weights evaluated at Q^*, to be the KP loop gain. It maps a perturbation of the common plastic weights to the resulting first-order change in the averaged plasticity drive, taking into account all the forward, backward, and cross-layer paths through the network. Writing $q := Q - Q^*$ for the perturbation, linearizing around Q^* gives

$$\tau_{\text{pot}} \dot{q} = \left(D_Q \overline{G}_* - \frac{\tau_{\text{pot}}}{\tau_{\text{dec}}} I \right) q \tag{16}$$

Because the mismatch dynamics are independent of Q, their eigenvalues are all $-1/\tau_{\text{dec}}$, while the remaining eigenvalues are $D_Q \overline{G}_* / \tau_{\text{pot}} - I/\tau_{\text{dec}}$, whose real parts are not constrained by the KP alignment mechanism. The common mode is therefore stable if and only if the *spectral abscissa* of the loop gain (the largest real part among its eigenvalues, a standard stability index for continuous-time linear dynamics; Trefethen & Embree, 2005) lies below the decay-to-potentiation ratio:

$$\max_i \mathrm{Re}\,\lambda_i(D_Q \overline{G}_*) \;<\; \frac{\tau_{\text{pot}}}{\tau_{\text{dec}}} \tag{17}$$

KP fixes the alignment eigenvalues but leaves the scale eigenvalues to the loop-gain spectrum of the surrounding circuit, including all within- and cross-layer return paths. This bound is most stringent under separation of timescales: the wider the separation $\tau_{\text{pot}} \ll \tau_{\text{dec}}$, the smaller the admissible ratio $\tau_{\text{pot}}/\tau_{\text{dec}}$, and the tighter the constraint on the loop gain.

5.2 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 (Pascanu et al., 2013) applies: for a depth-L leaky-ReLU chain with layerwise gain Q, $\|z_{l+1}\, y_l^\top\| \sim Q^{L-1}$. Continuous-time KP additionally creates a potentially 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. This loop does not exist in standard chain-rule backpropagation.

This loop-gain runaway can be measured directly. Figure 3a tracks the spectral abscissa $\max_i \operatorname{Re} \lambda_i(D_\theta \overline{G})$ of the Jacobian of the averaged drive with respect to all plastic weights $\theta = (W, V)$ along training across depths $L = 2, \ldots, 8$ on UCI digits-8×8, evaluated by automatic differentiation of the averaged drive followed by a shift-invert eigensolve. By the block-triangular argument of Appendix B, this $\max_i \operatorname{Re} \lambda_i(D_\theta \overline{G})$ equals the common-mode abscissa $\max_i \operatorname{Re} \lambda_i(D_Q \overline{G})$ that Condition (17) constrains. For shallow networks the abscissa stays small, within an order of magnitude of the stability threshold $\tau_{\mathrm{pot}}/\tau_{\mathrm{dec}} = 10^{-2}$, and the network trains; from $L \gtrsim 4$ it grows several orders of magnitude and the run diverges, with onset earlier at larger depth. The depths whose measured spectrum stays bounded are exactly the depths that train in Figure 2, and the spectrum's runaway precedes the weight explosion, consistent with our claim that the instability is a common-mode loop-gain runaway rather than a failure of alignment.

5.3 Bounding the loop-gain spectrum

Condition (17) asks that the loop-gain spectral abscissa stay below $\tau_{\mathrm{pot}}/\tau_{\mathrm{dec}}$. The loop gain grows with the magnitudes of the forward activities y_l and backward errors z_l that enter the plasticity drive $\langle z_l\, y_{l-1}^\top \rangle$, so the loop gain can be controlled locally by bounding the per-layer signals $\|y_l\|$ and $\|z_l\|$ at run time. We compare bounded activations with layer normalization (RMSNorm) and update clipping, two other local stabilization mechanisms.

Bounded activations cap each hidden unit individually, replacing the activation function by its clipped variant

$$\phi_l^{\mathrm{sat}}(s) \;=\; \mathrm{clip}\big(\phi_l(s),\, -c,\, +c\big), \tag{18}$$

with cap c shared across the network ($c = 1$ in our experiments), used in place of ϕ_l throughout Eqs. (8)–(9). With $|y_{l,i}| \le c$ at every hidden neuron, $\|y_l\|_2 \le c\sqrt{n_l}$; and the backward derivative is gated by saturation, so when $|y_{l,i}|$ sits at the cap the activation derivative is zero and the corresponding entries of $V_l\, z_{l+1}$ in Eq. (9) are killed, damping $\|z_l\|$ as well.

RMSNorm rescales each layer activity (and error) vector to a fixed root-mean-square magnitude set by a gain α (Zhang & Sennrich, 2019), directly fixing $\|y_l\|$ and $\|z_l\|$. Unlike BatchNorm it uses no batch statistics or running averages, depending only on the instantaneous layer vector. It is, however, a *layer-level* operation: forming the root-mean-square and rescaling couples every unit in the layer, with no natural local realization in a biological or analog substrate. We use it as a strong baseline for what good per-layer magnitude control achieves.

Bounded activations cap each unit while RMSNorm pins each layer's RMS to a fixed target; both bound the per-layer signal magnitudes that multiply into the drive, shrinking the entries of $D_\theta \overline{G}$ and reducing its spectral abscissa by one to two orders of magnitude so that it stays bounded rather than running away; update clipping leaves those entries untouched.

Update clipping leaves the signals untouched and instead caps how fast the weights can move, in analogy with gradient clipping: each layer's total update $d_l = z_l\, y_{l-1}^\top/\tau_{\mathrm{pot}} - W_l/\tau_{\mathrm{dec}}$ (and likewise for each V_l) is rescaled by $\min\big(1,\, c_u/\|d_l\|_F\big)$. This does not act on the loop gain: the drive Jacobian $D_Q \overline{G}$ is a function of the weights alone, and the cap binds only where $\|d_l\|_F > c_u$. At any equilibrium the update vanishes, so on a neighborhood of every equilibrium clipping is the identity and the clipped and unclipped flows coincide. Clipping therefore cannot change the stability of any equilibrium, and Condition (17) applies to it unchanged; what it changes is the speed at which an unstable configuration is left. Clipping should thus be able to slow the loop-gain runaway but not remove it.

5.4 Bounded activations outperform RMSNorm and update clipping

We now compare the mechanisms empirically (RMSNorm gain $\alpha = 0.5$; update-clipping caps $c_u \in \{0.3, 1\}$, chosen from the update magnitudes observed in stable training, with c_u=1 above nearly all of them and c_u=0.3 inside their upper range, so that clipping engages only on abnormally large updates). The stabilizing mechanism is visible directly in the loop gain. Figures 3b and 3c track the spectral abscissa $\max_i \operatorname{Re} \lambda_i(D_\theta \overline{G})$ along training for vanilla, bounded-activation, and RMSNorm continuous-time KP at $L = 8$ and $L = 10$ on digits-8×8, holding initialization and inputs

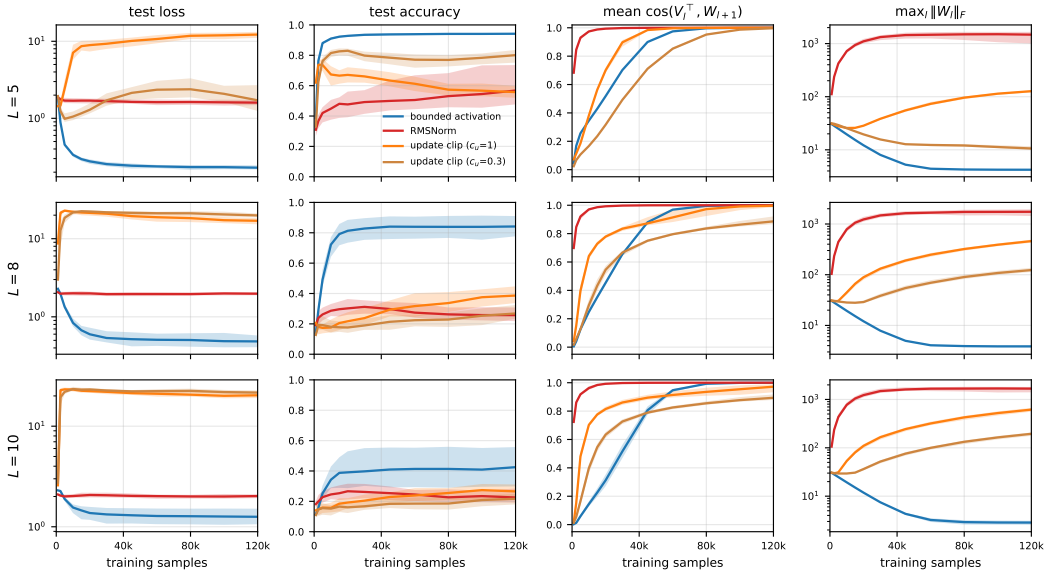


Figure 4: Continuous-time KP on MNIST at $L = 5, 8, 10$ (rows): test loss (log), test accuracy, mean alignment $\cos(V_l^\top, W_{l+1})$, and largest per-layer weight norm (log), for bounded activations (blue), RMSNorm (red), and update clipping at c_u=1 (orange) and c_u=0.3 (tan); 100 seeds per condition, median with interquartile band. Every condition aligns to $\cos \approx 1$; accuracy, loss, and weight scale are what separate them. Per-seed traces and full envelopes: Figures 6 and 5.

fixed, across 100 seeds per condition. Vanilla runs away by orders of magnitude on every seed until the weights overflow; both bounded activations and RMSNorm instead hold the abscissa at order 10^{-1}–10^{0} for the entire run (final medians 0.2–0.5 across depths, transient peaks near 2), orders of magnitude below the escape. Update clipping (c_u=1) shows the complementary dissociation: it holds the weights finite for the entire run, yet its abscissa continues along vanilla's runaway trajectory, past the point where vanilla's weights overflow, reaching 10^{5}–10^{9}. The clipped circuit survives by moving slowly through the same runaway landscape, not by reshaping it. The two signal-bounding mechanisms thus control the loop-gain spectrum directly, the measured counterpart of the magnitude argument of §5.3, identifying it as the mechanism by which they restore stability. This validates the central claim of §5: stability is decided by the loop-gain spectrum. Where the spectrum is held bounded, the weights and the loss stay bounded and training can proceed; where it escapes, the circuit fails, by weight overflow when the updates are unconstrained and by cap-rate weight drift with rising loss when they are clipped.

Figure 4 compares all four conditions on continuous-time KP trained on MNIST at depths $L = 5$, 8, and 10, Xavier initialization for both forward and feedback weights, two MNIST epochs (120,000 samples), and 100 seeds per condition; Table 1 (Appendix C) summarizes the end of training. Bounded activations reach the highest accuracy at every depth (medians $0.94/0.84/0.43$ at $L = 5/8/10$), with RMSNorm at $0.57/0.26/0.23$ and update clipping between the two at c_u=1 ($0.56/0.39/0.27$). Clipping is also cap-sensitive, with no uniformly good value: the tight cap that wins at L=5 (c_u=0.3 reaches 0.80, ahead of RMSNorm) is the one that freezes learning at depth ($0.27/0.22$ at $L = 8/10$), and the loose cap does the reverse. Two columns of Table 1 carry the argument. First, alignment does not separate the conditions: every mechanism drives $\cos(V_l^\top, W_{l+1})$ to 0.89–1.00, in agreement with the contraction guarantee of §5, while accuracy spans 0.22–0.94. Second, the weight scale and the loss expose what accuracy hides: bounded activations *shrink* the weights (largest per-layer norm $\sim$3–4) and reduce the loss; RMSNorm normalizes the signals but lets the weights grow to $\sim 1.7 \times 10^3$ with loss pinned near chance; clipping merely slows the weight growth (10^2–6×10^2) while its loss climbs to 12–22, far above chance: enormous, confidently wrong logits from weights drifting along the runaway. Controlling the update magnitude is not controlling the gain.

6 DISCUSSION

The central finding of this work is that alignment does not imply stability. A heterosynaptic plasticity rule may carry an explicit compensatory decay term, but once the plastic weight is embedded in a learning circuit with feedback loops, perturbing it changes the very neural signals that drive its own update. Stability depends on the closed-loop *loop gain*, the sensitivity of the plasticity drive to weight perturbations. Local balance fixes where the dynamics can rest, not whether it stays there.

Continuous-time Kolen–Pollack learning makes this concrete. Its reciprocal updates produce an exact local alignment guarantee independent of data, activity, and depth, but they leave the common (scale) mode governed by the loop-gain spectrum, which alignment does not constrain. When we measure that spectrum directly, we find its abscissa stays bounded for shallow networks but runs away with depth (Figure 3a), and this runaway is the cause of the depth-dependent divergence in Figure 2. A circuit can thus solve the weight-transport problem and still fail to control the gain of its own plastic dynamics.

This perspective reframes the engineered stabilizers used in prior KP-like and asymmetric-feedback implementations (Liao et al., 2016; Akrout et al., 2019; Moskovitz et al., 2018; Webster et al., 2020). Batch normalization (Ioffe & Szegedy, 2015), weight decay, gradient clipping (Pascanu et al., 2013), adaptive optimizers (Kingma & Ba, 2015), and learning-rate schedules are not merely implementation conveniences; each regulates the activity, error, or weight magnitudes that set the loop gain, at the price of a nonlocality in time or space that a continuous-time substrate cannot afford.

A substrate-plausible way to control the loop gain is to bound it with strictly local machinery. Taking RMSNorm as an idealized reference for per-layer magnitude control, we find that bounded activation functions control the loop-gain abscissa just as well, and in fact train better: at $L = 5$ they reach 0.94 test accuracy on MNIST, and at $L = 10$ they outperform RMSNorm, even though both attain near-perfect alignment. Controlling the loop gain, rather than aligning the feedback path, is what restores stable learning. Update clipping sharpens this conclusion: it is no less local than RMSNorm, keeps the weights finite over the training horizon, and aligns the feedback path, yet it leaves the loop gain uncontrolled and learning stalls (Figures 3b and 3c). What must be bounded is the gain of the loop, not the speed of the weights.

Two caveats bound these claims. First, linear stability analysis is not a full account of the nonlinear dynamics; in practice the abscissa is supercritical at every depth we tested, so the operative distinction is bounded versus runaway: in the stabilized runs the abscissa settles at order 10^{-1}–10^{0}, one to two orders of magnitude above the decay ratio $\tau_{\mathrm{pot}}/\tau_{\mathrm{dec}}$, whereas in the unstable runs it escapes by many further orders. Second, gain control is necessary but not sufficient for high accuracies: signal starvation from cascade lag through the depth-L chain of low-pass filters (Appendix D) caps trainable accuracy at $L = 10$ and drives the broad seed distribution there. This is a separate, abscissa-independent failure mode, regardless of how well the loop gain is controlled, and seems to be an important open question.

More broadly, stable local learning appears to require two separable ingredients: credit-assignment alignment, so that useful error information reaches each plastic synapse, and loop-gain control, so that plastic perturbations are not amplified by the signals they induce. KP supplies the first for free through reciprocal symmetry; the second must be added, and we show that bounded activations are one local, substrate-native way to add it. We expect the same loop-gain lens, and the same need for explicit gain control, to apply to other HSP circuits, including those realized in continuous-time analog substrates.

AI USE STATEMENT

In this work we used generative AI tools (large language models operating as coding and writing assistants) for drafting and editing text, for writing, debugging, and running experiment and analysis code, for producing figures, and for checking bibliographic records against publisher and index records. The research questions, the theoretical analysis, the experimental design, and the interpretation of the results are the authors' own. We have reviewed all AI-assisted work: every claim and number in the paper was checked against the underlying data by the authors, and all AI-generated code was reviewed and tested by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Reproducibility statement

The continuous-time KP model, its four state families, and all time constants are specified in §4; the stabilization mechanisms and their hyperparameters ($c = 1$, $\alpha = 0.5$, $c_u \in \{0.3, 1\}$) in §5.3; datasets, initializations, sample budgets, and seed counts in §5.4 and Appendix C. The matrix-free computation of the loop-gain spectral abscissa, including the automatic-differentiation drive, the numerical-abscissa bound, the shift-invert eigensolve, and its validation against dense eigendecompositions, is described in Appendix B. Every experiment is seeded, and a single integer seed determines both the initialization and the data order. Source code for the simulator, the abscissa computation, and all figure generators will be released with the camera-ready version.

References

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.

Karl Johan Åström and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

Marc Gong Bacvanski, Liu Ziyin, and Tomaso Poggio. Does feedback alignment work at biological timescales?, 2025. URL https://arxiv.org/abs/2510.18808.

Sergey Bartunov, Adam Santoro, Blake A. Richards, Luke Marris, Geoffrey E. Hinton, and Timothy P. Lillicrap. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. In *Advances in Neural Information Processing Systems*, 2018. URL https://arxiv.org/abs/1807.04587.

Yoshua Bengio, Dong-Hyun Lee, Jorg Bornschein, Thomas Mesnard, and Zhouhan Lin. Towards biologically plausible deep learning, 2015. URL https://arxiv.org/abs/1502.04156.

Elie L. Bienenstock, Leon N. Cooper, and Paul W. Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1):32–48, 1982. doi: 10.1523/JNEUROSCI.02-01-00032.1982.

Francis Crick. The recent excitement about neural networks. *Nature*, 337(6203):129–132, 1989. doi: 10.1038/337129a0.

Dominique Debanne, Emilie Campanac, Andrzej Bialowas, Edmond Carlier, and Gisèle Alcaraz. Axon physiology. *Physiological Reviews*, 91(2):555–602, 2011. doi: 10.1152/physrev.00048.2009.

Stephen Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11(1):23–63, 1987. doi: 10.1111/j.1551-6708.1987.tb00862.x.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, volume 37, pp. 448–456, 2015. URL https://proceedings.mlr.press/v37/ioffe15.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

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

Daniel Kunin, Aran Nayebi, Javier Sagastuy-Brena, Surya Ganguli, Jonathan Bloom, and Daniel Yamins. Two routes to scalable credit assignment without weight symmetry. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 5511–5521, 2020. URL https://proceedings.mlr.press/v119/kunin20a.html.

Julien Launay, Iacopo Poli, and Florent Krzakala. Principled training of neural networks with direct feedback alignment, 2019. URL https://arxiv.org/abs/1906.04554.

Richard B. Lehoucq, Danny C. Sorensen, and Chao Yang. *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM, 1998.

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, pp. 1837–1844, 2016. doi: 10.1609/aaai.v30i1.10279.

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: 13276, 2016. doi: 10.1038/ncomms13276.

Timothy P. Lillicrap, Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020. doi: 10.1038/s41583-020-0277-3.

Theodore H. Moskovitz, Ashok Litwin-Kumar, and L. F. Abbott. Feedback alignment in deep convolutional networks, 2018. URL https://arxiv.org/abs/1812.06488.

Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. In *Advances in Neural Information Processing Systems*, 2016.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982. doi: 10.1007/BF00275687.

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*, pp. 1310–1318, 2013. URL https://proceedings.mlr.press/v28/pascanu13.html.

Maria Refinetti, Stéphane D'Ascoli, Ruben Ohana, and Sebastian Goldt. Align, then memorise: The dynamics of learning with feedback alignment. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8925–8935, 2021. URL https://proceedings.mlr.press/v139/refinetti21a.html.

Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986. doi: 10.1137/0907058.

Mahmoud Salami, Chiaki Itami, Tadaharu Tsumoto, and Fumitaka Kimura. Change of conduction velocity by regional myelination yields constant latency irrespective of distance between thalamus and cortex. *Proceedings of the National Academy of Sciences*, 100(10):6174–6179, 2003. doi: 10.1073/pnas.0937380100.

Lloyd N. Trefethen and Mark Embree. *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*. Princeton University Press, 2005.

Gina G. Turrigiano. The self-tuning neuron: Synaptic scaling of excitatory synapses. *Cell*, 135(3): 422–435, 2008. doi: 10.1016/j.cell.2008.10.008.

Gina G. Turrigiano and Sacha B. Nelson. Homeostatic plasticity in the developing nervous system. *Nature Reviews Neuroscience*, 5(2):97–107, 2004. doi: 10.1038/nrn1327.

Matthew Bailey Webster, Jonghyun Choi, and Changwook Ahn. Learning the connections in direct feedback alignment, 2020. URL https://openreview.net/forum?id=zgGmAx9ZcY. Submission to ICLR 2021; not accepted.

James C. R. Whittington and Rafal Bogacz. Theories of error back-propagation in the brain. *Trends in Cognitive Sciences*, 23(3):235–250, 2019. doi: 10.1016/j.tics.2018.12.005.

Will Xiao, Honglin Chen, Qianli Liao, and Tomaso Poggio. Biologically-plausible learning algorithms can scale to large datasets. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1811.03567.

Friedemann Zenke, Wulfram Gerstner, and Surya Ganguli. The temporal paradox of Hebbian learning and homeostatic plasticity. *Current Opinion in Neurobiology*, 43:166–176, 2017. doi: 10.1016/j.conb.2017.03.015.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, pp. 12360–12371, 2019. URL https://papers.neurips.cc/paper/9403-root-mean-square-layer-normalization.

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

A ADIABATIC REDUCTION IN THE ALIGNED LIMIT

The hierarchy (11) 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. (8)–(9) for fixed $\{W_l, V_l\}$ gives the slow-manifold values

$$y_l^* \;=\; \phi_l\big(W_l\, y_{l-1}^*\big), \qquad z_l^* \;=\; \phi_l'(s_l^*) \odot \big(V_l\, z_{l+1}^*\big), \qquad z_L^* \;=\; y^* - \mathrm{softmax}(s_L^*). \tag{19}$$

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. (19) reduces to the ordinary backpropagation recursion

$$z_l^* \;=\; \phi_l'(s_l^*) \odot W_{l+1}^\top\, z_{l+1}^*. \tag{20}$$

Substituting the slow-manifold values into the plasticity equations (10) 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 (11) actually holding during training, which itself depends on the plasticity drive remaining bounded against the decay term.

B COMPUTING THE LOOP-GAIN SPECTRAL ABSCISSA

Figures 3a and 3b plot the spectral abscissa $\max_i \mathrm{Re}\,\lambda_i(D_\theta \overline{G})$ of the Jacobian of the averaged drive with respect to *all* plastic weights $\theta = (W_l, V_l)$, evaluated at the network's actual weights along training. As we now show, it equals the common-mode loop-gain abscissa $\max_i \mathrm{Re}\,\lambda_i(D_Q \overline{G})$ of §5 that Condition (17) constrains:

$$\max_i \mathrm{Re}\,\lambda_i(M), \qquad M \;=\; D_\theta \overline{G}. \tag{21}$$

The two coincide for a structural reason, not because the network happens to be aligned. The reciprocal updates of V_l and W_{l+1} share the same outer-product drive, so the mismatch drive cancels identically and $\dot{\Delta}_l = -\Delta_l/\tau_{\mathrm{dec}}$ (13) holds at *every* weight configuration, aligned or not. Hence $\partial\dot{\Delta}/\partial Q = 0$ everywhere, and in the (Q, Δ) coordinates M is block upper-triangular,

$$M \;=\; \begin{pmatrix} D_Q \overline{G} & * \\ 0 & 0 \end{pmatrix},$$

where the upper-right block $* = \partial\overline{G}/\partial\Delta$ does not affect the spectrum and the lower block is zero, since the mismatch drive vanishes identically. The drive Jacobian therefore contributes mismatch eigenvalues at 0 (which become the contracting $-1/\tau_{\mathrm{dec}}$ modes of Eq. (13) once the decay term is restored), so $\max_i \mathrm{Re}\,\lambda_i(M) = \max\big(\max_i \mathrm{Re}\,\lambda_i(D_Q\overline{G}),\, 0\big)$; the two abscissae coincide whenever the common mode is supercritical, as it is at every depth we measure. We therefore report $\max_i \mathrm{Re}\,\lambda_i(D_\theta \overline{G})$ evaluated at the actual, generally unaligned weights, equal in the aligned limit $\Delta \to 0$ to the manifold gain $\max_i \mathrm{Re}\,\lambda_i(D_Q \overline{G}_*)$ that KP drives the system toward. Working with the full operator M is what lets us evaluate the abscissa at the actual (generally unaligned) weights by automatic differentiation. M is dense with $P \sim 10^4$–10^5 rows, so we never build it explicitly: we access it only through matrix–vector products and recover the abscissa with iterative methods. Three ingredients are involved.

The drive and its action (automatic differentiation). On the slow manifold (19) the averaged drive $\overline{G}(\theta) = \big(\langle z_l\, y_{l-1}^\top\rangle\big)_l$ is computed by one forward pass through W, one backward pass through V, and the data-averaged outer products $z_l\, y_{l-1}^\top$ (with bounded activations the forward pass clips each unit and the backward pass uses the saturation-gated derivative of ϕ_l^{sat} (18), which is the exact gradient of the clipped network; for RMSNorm the forward and backward passes additionally rescale each layer to the fixed homeostatic RMS, and the automatic differentiation accounts for that rescale). Automatic differentiation of this computation supplies the two products we need, without ever forming M:

$$Mv \;=\; \tfrac{d}{d\epsilon}\,\overline{G}(\theta+\epsilon v)\big|_{\epsilon=0} \text{ (forward mode, a Jacobian–vector product),} \tag{22}$$

$$M^\top v \text{ (reverse mode, a vector–Jacobian product),} \tag{23}$$

each one forward-plus-backward sweep through $\overline{G}$.

Iterative eigensolvers. Given only the map $v \mapsto Mv$, Krylov eigensolvers (*Lanczos* when the operator is symmetric, *Arnoldi* in general) estimate a few eigenvalues of M from the subspace $\mathrm{span}\{v, Mv, M^2v, \ldots, M^kv\}$ by projecting M onto it and diagonalizing the small $k \times k$ projection ($k \ll P$). They converge quickly to eigenvalues that are *dominant* (largest in magnitude) and well separated, and poorly to interior ones. This is the difficulty: the quantity we want, the rightmost eigenvalue's real part, is *not* the largest in magnitude for our strongly non-normal M (Trefethen & Embree, 2005), and running Arnoldi on M directly returns spurious values. We obtain the abscissa in two steps that only ever ask the solvers for a dominant eigenvalue.

Step 1: a rigorous upper bound from the symmetric part. The symmetric part $S = \frac{1}{2}(M + M^\top)$ has real eigenvalues, and for any unit eigenvector $Mu = \lambda u$,

$$\mathrm{Re}\,\lambda \;=\; \mathrm{Re}\,(u^*Mu) \;=\; u^*Su \;\le\; \lambda_{\max}(S) \;=:\; \nu, \tag{24}$$

so the *numerical abscissa* ν upper-bounds the abscissa $\max_i \mathrm{Re}\,\lambda_i(M)$. It is the *largest* eigenvalue of a *symmetric* operator, a dominant target, so Lanczos computes it reliably from $Sv = \frac{1}{2}(Mv + M^\top v)$.

Step 2: shift-invert for the rightmost eigenvalue. To turn the rightmost eigenvalue into a dominant target, we run Arnoldi not on M but on the *shift-invert* operator $(M - \sigma I)^{-1}$, whose eigenvalues are $1/(\lambda_i - \sigma)$, using a real shift $\sigma = \nu + \delta$ placed just to the right of the whole spectrum ($\delta > 0$ small). Eigenvalues λ_i near σ have the largest $|\lambda_i - \sigma|^{-1}$ and become the dominant eigenvalues Arnoldi recovers easily (we use the implicitly restarted Arnoldi of ARPACK (Lehoucq et al., 1998)); we request the several closest to σ and report the largest real part among them. For a strongly non-normal operator with a complex spectrum, closest-to-σ is not guaranteed to be the globally rightmost eigenvalue, so we treat the result as an estimate, bracketed from above by the rigorous bound ν of Step 1 and cross-checked against a dense eigendecomposition wherever M is small enough to form (below). Each Arnoldi step needs $(M - \sigma I)^{-1}b$, i.e. a solution of the linear system $(M - \sigma I)x = b$, which we get with GMRES (Saad & Schultz, 1986), a Krylov *linear* solver that again uses only the product Mv (and whose convergence flag we check and warn on; regardless of the inner-solve quality, the numerical abscissa ν of Step 1 remains a rigorous upper bound on the true abscissa, so a failed solve is flagged rather than silently trusted). Since σ exceeds the bound ν it lies outside the field of values of M, so $(M - \sigma I)$ is well conditioned and GMRES converges quickly.

Validation. On small networks where M can be formed explicitly, the matrix-free abscissa matches a dense eigendecomposition to $\sim 10^{-9}$, and the autodiff drive matches the analytic gradient of the (clipped) cross-entropy to $\sim 10^{-8}$.

C SEED DISTRIBUTION ACROSS DEPTH

The seed-to-seed variability in Figure 4 grows with depth. Figure 6 shows the per-seed traces for the two signal-bounding mechanisms, and Figure 5 the test-accuracy percentile envelopes for all four conditions. With bounded activations the distribution is essentially deterministic at $L = 5$ (all seeds in $[0.93, 0.96]$), broadens at $L = 8$ (median 0.84, p_{25}–p_{75} of $[0.78, 0.91]$), and becomes wide and roughly flat at $L = 10$ (median 0.43, spanning $[0.18, 0.88]$), with the distribution shape set by $\sim$60,000 samples. RMSNorm is lower and narrower at every depth.

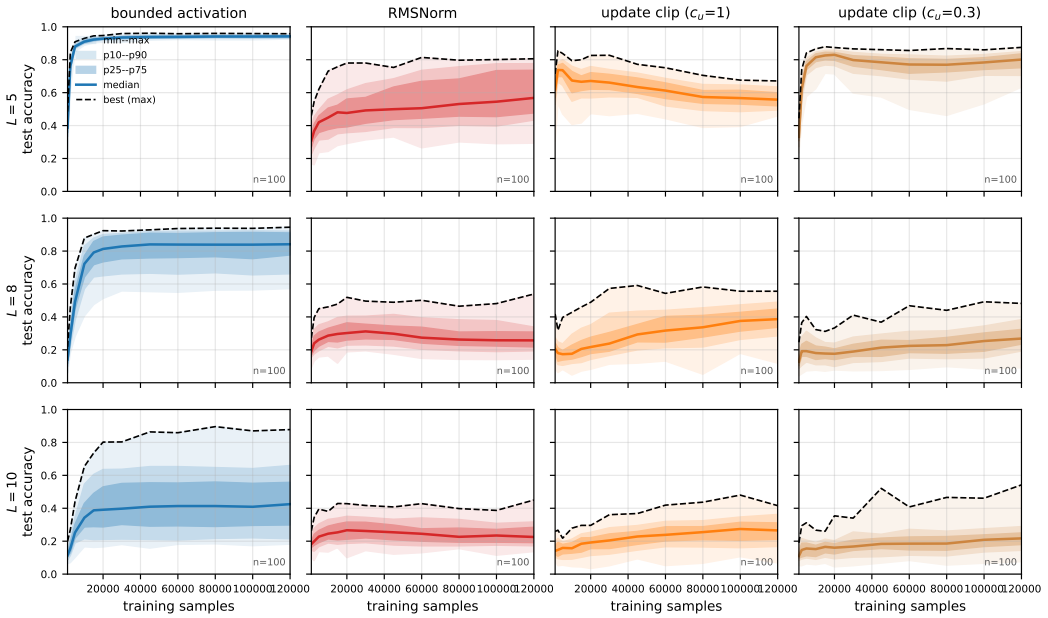


Figure 5: Test-accuracy percentile envelopes across seeds vs. training samples for all four conditions (columns: bounded activation, RMSNorm, update clipping at c_u=1 and c_u=0.3) at $L = 5, 8, 10$ (rows). Bands are the min–max, p_{10}–p_{90}, and p_{25}–p_{75} envelopes; the solid line is the median and the dashed black line the per-sample maximum (best seed). Bounded activations are essentially deterministic at $L = 5$ and broaden with depth, becoming wide and roughly flat at $L = 10$ (median 0.43, best 0.88), while RMSNorm stays lower and narrower throughout.

Table 1 gives the end-of-training summary across all four conditions; the distributions are not Gaussian, so the envelopes report percentiles rather than mean ± std.

L	mechanism	test acc	test loss	$\cos(V_l^\top, W_{l+1})$	$\max_l \|W_l\|_F$
5	bounded activation	**0.942** [0.936, 0.946]	**0.23**	1.00	**4.2**
	RMSNorm	0.568 [0.477, 0.736]	1.60	1.00	1.5×10^3
	update clip, c_u=1	0.558 [0.520, 0.599]	12.2	1.00	1.3×10^2
	update clip, c_u=0.3	0.801 [0.752, 0.835]	1.71	1.00	11
8	bounded activation	**0.841** [0.777, 0.910]	**0.49**	1.00	**3.9**
	RMSNorm	0.257 [0.218, 0.308]	1.98	1.00	1.7×10^3
	update clip, c_u=1	0.387 [0.337, 0.447]	17.0	1.00	4.6×10^2
	update clip, c_u=0.3	0.269 [0.235, 0.324]	19.9	0.89	1.2×10^2
10	bounded activation	**0.425** [0.299, 0.558]	**1.26**	1.00	**2.9**
	RMSNorm	0.225 [0.192, 0.284]	2.02	1.00	1.7×10^3
	update clip, c_u=1	0.265 [0.211, 0.311]	20.3	0.97	6.2×10^2
	update clip, c_u=0.3	0.217 [0.180, 0.250]	21.6	0.89	2.0×10^2

Table 1: End of training (120,000 samples): across-seed median of final test accuracy (with interquartile range), test loss, mean alignment, and the largest per-layer weight norm; 100 seeds per condition. Chance loss is $\ln 10 \approx 2.30$. Alignment is uniformly high in every condition; accuracy, loss, and weight scale are what distinguish the mechanisms.

D Common failure modes of continuous-time KP in deep networks

Failure mode 1: alignment without training. The weight-scale dynamics of §5 already show that vanilla continuous-time KP can drive $\Delta_l \to 0$ while Q_l blows up: the mismatch contracts and the

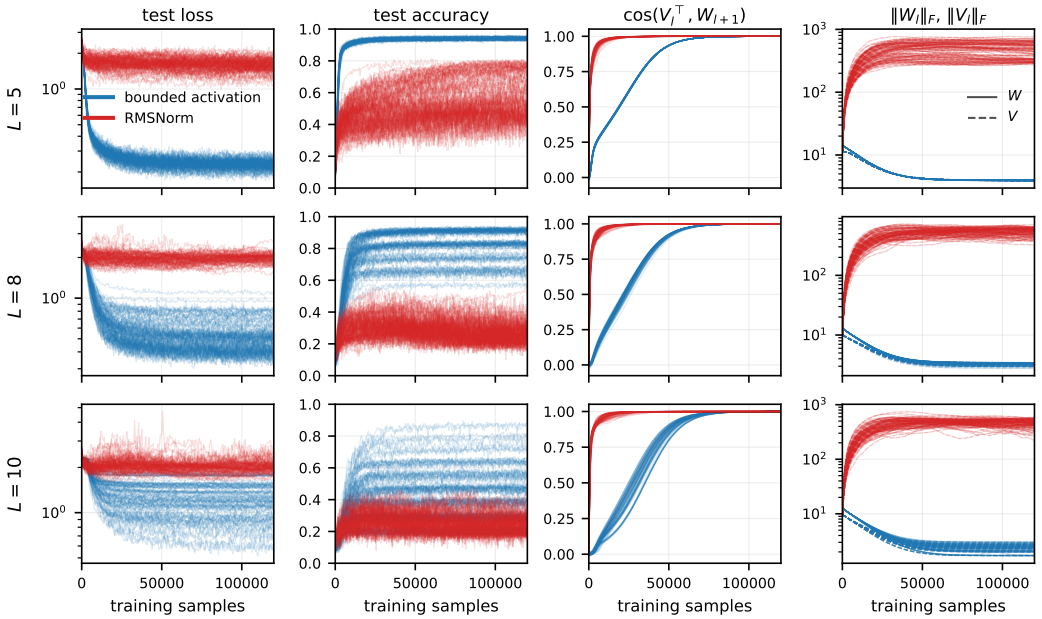


Figure 6: Per-seed traces for two signal-bounding mechanisms on MNIST at $L = 5$ (top), 8 (middle), and 10 (bottom): bounded activations (ϕ_l^{sat} at $c = 1.0$, blue) and RMSNorm ($\alpha = 0.5$, red); 100 seeds per condition, one line per seed (no median overlay). Columns: test cross-entropy loss (log scale), test accuracy, mean cosine alignment $\cos(V_l^\top, W_{l+1})$, and mean Frobenius norms $\|W_l\|_F$ (solid) and $\|V_l\|_F$ (dashed, log scale). The per-seed accuracy traces at $L = 8, 10$ collect into discrete bands, visible here but averaged away in the envelopes.

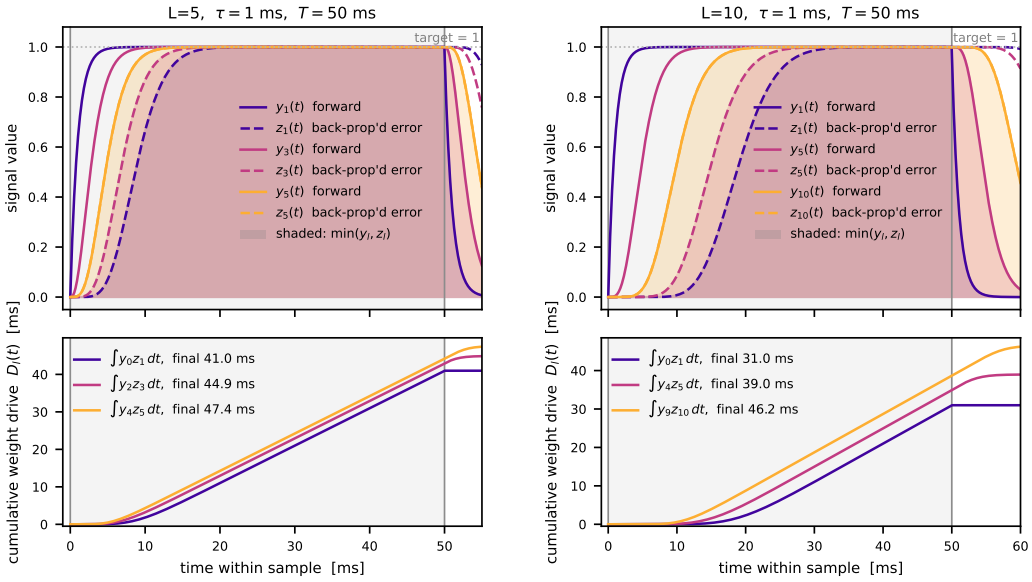
weight scale runs away simultaneously. The loss panel of Figure 4 shows a milder version of the same phenomenon for RMSNorm: $\cos(V_l^\top, W_{l+1}) \to 1$ within a few thousand samples and stays there, but the test loss only falls to $\approx$1.6–2.0 (depending on depth), far above the bounded-activation loss (e.g. $\approx$0.23 at $L = 5$) and, at $L = 8$ and $L = 10$, near the chance level $\ln 10 \approx 2.3$. Applying RMSNorm to z_L at the output layer rescales the gradient signal away from its loss-derived magnitude $y^* - \text{softmax}(s_L)$, 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

$$D_l \;=\; \int_0^{T_{\text{sample}}} y_{l-1}(t)\, z_l(t)\, dt \tag{25}$$

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 $(L-1)\,\tau_{\text{inf}}$ in the forward direction and the same on the backward side, so as $L\,\tau_{\text{inf}}$ approaches $T_{\text{sample}}/2$ 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 $\tau_{\text{inf}}/T_{\text{sample}}$ but not on the choice of scale-control mechanism.

Figure 7 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 ($D_1 \approx 41$ ms, $D_5 \approx 47$ ms over the 50 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 $D_1 \approx 31$ ms



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

Figure 7: Cascade-lag overlap of forward $y_l(t)$ (solid) and back-propagated signal $z_l(t)$ (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). To isolate the cascade timing from the error magnitude, the backward cascade is seeded with $z_L = y_L$ (a shape/overlap proxy rather than a literal output error). 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 ($l = 1$) 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.

while the output-side stays at $D_{10} \approx 46\,\text{ms}$. 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 whether the loop-gain condition (17) is satisfied.

Putting the failure modes together. The $L = 10$ plateau in Figures 4 and 5 reflects both effects. RMSNorm suffers failure 1 across all depths and additionally hits failure 2 at $L = 10$, so its accuracy stays well below bounded activations once $L > 5$ (median 0.26 vs 0.84 at $L = 8$ and 0.24 vs 0.43 at $L = 10$), with its test loss plateauing near $\ln 10$. Per-neuron saturation avoids failure 1 (it trains well at $L = 5$ and $L = 8$) but is bottlenecked by failure 2 at $L = 10$, which is why its accuracy spreads broadly (best seed near 0.88, median around 0.43) even though the loop gain is controlled and cosine alignment holds. The wide seed distribution in Figure 5 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.