Alignment Is Not Stability: Closed-Loop Gain Control in Heterosynaptic Plastic Circuits

Marc Gong Bacvanski MARCBAC@MIT.EDU
MIT

Liu Ziyin
NTT Research; MIT

Tomaso Poggio
MIT

Abstract

Heterosynaptic plasticity (HSP) rules combine local activity-dependent potentiation with compensatory decay, a biologically and physically plausible substrate for learning. However, in a learning network the local rule is embedded in a closed loop: perturbations of plastic weights alter neural signals, which in turn alter the plasticity drive, so stability is not guaranteed. We show that linearizing an HSP motif around a balanced operating point yields a *return-gain* condition: stability depends on the closed-loop sensitivity from a weight back to its own learning signal. We apply this analysis to continuous-time Kolen–Pollack (KP) learning, a well-known HSP-style mechanism for aligning feedback weights with forward weights. While KP's symmetric updates exponentially contract the forward–feedback mismatch, they leave a weight-scale mode governed by the generic HSP return-gain spectrum; feedback alignment is therefore not sufficient for stability, and this causes deep continuous-time KP networks to diverge. We show that bounded activation functions limit the effective return gain and stabilize continuous-time KP in deep multilayer perceptrons, matching or exceeding layer-local normalization. These results identify closed-loop gain control as a central requirement for stable heterosynaptic learning in biological and continuous-time substrates.

1. Introduction

Heterosynaptic plasticity (HSP) is a general, local mechanism for learning in biological and physical systems: two-factor rules that pair activity-dependent potentiation with a compensatory decay, tiled into a circuit so that plastic change is driven by signals carried elsewhere in the network [13]. Such rules are attractive for credit assignment because each update depends only on nearby variables, yet the network as a whole can implement gradient-like learning. But in a learning network the rule is embedded in a *closed loop* (Figure 1): a change in a plastic weight changes neural activity, which changes the local plasticity signals, which feed back into the same weight. Stability therefore depends not on the local rule alone but on the return path created by the surrounding circuit.

Linearizing a single HSP motif within its circuit yields a *return-gain* condition: stability depends on the closed-loop sensitivity from a weight back to its own learning signal; when this return gain is large relative to the decay, perturbations are amplified rather than damped. Local balance between potentiation and decay fixes where the dynamics can rest, not whether it stays there. We apply this lens to continuous-time Kolen–Pollack (KP) learning [2, 3], a biologically plausible rule whose paired Hebbian-plus-decay updates align the feedback path with the transpose of the forward path (Figure 1, right) without weight transport [1]. KP's reciprocal symmetry contracts the forward–feedback mismatch exactly, so alignment succeeds, but this controls only one mode of the circuit.

© M.G. Bacvanski, L. Ziyin & T. Poggio.

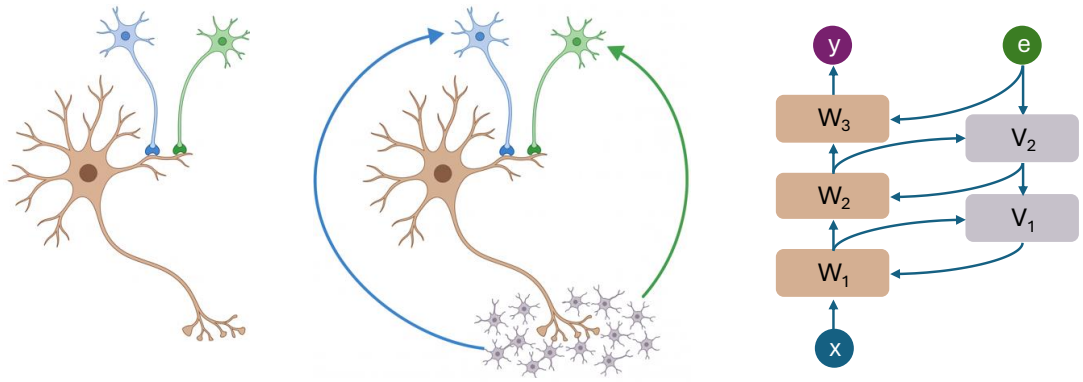


Figure 1: **From a heterosynaptic motif to a closed-loop learning circuit.** *Left:* an HSP motif: presynaptic inputs onto a postsynaptic neuron, each synapse modified purely by local activity. *Middle:* embedded in a network, the motif sits in a closed loop: the neuron's output propagates through the surrounding circuit and returns to drive the very synapses being updated. *Right:* Kolen–Pollack learning as a structured 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 carries the error e backward.

The weight-*scale* mode is governed by the generic return-gain spectrum, which alignment alone does not constrain. **Alignment is not stability**: the scale mode runs away with depth, and continuous-time KP diverges in deep networks.

This suggests a local route to stabilization: reduce the return gain by bounding the neural signals that enter the plasticity drive. We show that a strictly per-neuron bounded activation does exactly this. Bounded activations hold the *measured* return-gain abscissa down across depth and seeds, and match or exceed layer-local normalization (RMSNorm) on MNIST—all without the batch statistics, weight decay, gradient clipping, or learning-rate schedules that prior KP-like and asymmetric-feedback implementations rely on [1, 4, 6, 7, 11] and that do not translate to a continuous-time substrate.

2. Linear stability of heterosynaptically plastic circuits

The simplest HSP motif updates a weight $W \in \mathbb{R}^{m\times n}$ from two local signals $a \in \mathbb{R}^m$, $b \in \mathbb{R}^n$,

$$\dot{W} = \eta\, ab^\top - \gamma W, \tag{1}$$

with potentiation rate $\eta > 0$ and decay rate $\gamma > 0$; a balanced operating point satisfies $\eta\, a^* b^{*\top} \approx \gamma W^*$. Perturbing $W = W^* + \delta W$, $a = a^* + \delta a$, $b = b^* + \delta b$ and dropping the second-order term,

$$\delta\dot{W} = -\gamma\, \delta W + \eta\big(\delta a\, b^{*\top} + a^*\, \delta b^\top\big). \tag{2}$$

In isolation $\delta a = \delta b = 0$ and the motif is trivially stable. In a learning network, however, perturbing W changes the signals through the surrounding circuit: $\delta a = A[\delta W]$ and $\delta b = B[\delta W]$ for linear

maps A, B determined by the rest of the architecture. Defining the return-gain operator $T[\delta W] := A[\delta W]\, b^{*\top} + a^*\, B[\delta W]^\top$,

$$\delta \dot{W} = (-\gamma I + \eta\, T)[\delta W], \qquad \text{stable} \iff \max_i \operatorname{Re} \lambda_i(-\gamma I + \eta T) < 0. \tag{3}$$

The same condition applies to any plastic subspace of a multi-motif circuit, with the rest of the architecture absorbed into A and B: *stability is governed by the closed-loop return gain*, the sensitivity of a weight's drive to its own perturbation, which local balance alone does not constrain.

3. Continuous-time Kolen–Pollack learning

Continuous-time KP [2] removes the forward/backward phase separation of the discrete rule by letting every state relax through first-order dynamics, modeling membrane RC integration or analog capacitive nodes. For a depth-L network with forward activities y_l, backward errors z_l, forward weights W_l, and feedback weights $V_l \approx W_{l+1}^\top$, the signals relax on time constants $\tau_{\text{inf}}, \tau_{\text{err}}$,

$$\tau_{\text{inf}}\, \dot{y}_l = \phi_l(s_l) - y_l, \quad s_l = W_l\, y_{l-1}, \qquad \tau_{\text{err}}\, \dot{z}_l = \phi_l'(s_l) \odot \big(V_l\, z_{l+1}\big) - z_l, \tag{4}$$

($V_l z_{l+1}$ replaced by the loss derivative at the output), and each weight follows Hebbian potentiation minus linear decay,

$$\tau_{\text{pot}}\, \dot{W}_l = z_l\, y_{l-1}^\top - \tfrac{\tau_{\text{pot}}}{\tau_{\text{dec}}} W_l, \qquad \tau_{\text{pot}}\, \dot{V}_l = y_l\, z_{l+1}^\top - \tfrac{\tau_{\text{pot}}}{\tau_{\text{dec}}} V_l. \tag{5}$$

We use a leaky-ReLU ϕ_l and the timescale separation $\tau_{\text{inf}}, \tau_{\text{err}} \ll T_{\text{sample}} \ll \tau_{\text{pot}} \ll \tau_{\text{dec}}$ ($1\,\text{ms}, 1\,\text{ms}, 50\,\text{ms}, 10\,\text{s}, 10^3\,\text{s}$); all states evolve together, without phase separation. This is a natural model for biological or analog credit assignment, but it becomes unstable in deeper networks, which is what we now analyze.

4. Stability of continuous-time KP

KP aligns the feedback path but does not by itself control weight scale. For each reciprocal pair define the common and mismatch coordinates $Q_{l+1} = \frac{1}{2}(V_l^\top + W_{l+1})$ (with $Q_1 = W_1$) and $\Delta_l = V_l^\top - W_{l+1}$. Adding and subtracting the paired updates of Eq. (5),

$$\dot{\Delta}_l = -\tfrac{1}{\tau_{\text{dec}}} \Delta_l, \qquad \tau_{\text{pot}}\, \dot{Q}_l = z_l\, y_{l-1}^\top - \tfrac{\tau_{\text{pot}}}{\tau_{\text{dec}}} Q_l. \tag{6}$$

KP's alignment guarantee is that the mismatch Δ contracts exponentially, independent of activity, data, and depth. However, the common (scale) coordinate Q has no such guarantee.

Return-gain condition. Under timescale separation the fast signals equilibrate, and the slow drive is $\overline{G}(Q) = \big(\langle z_l\, y_{l-1}^\top \rangle\big)_l$, averaged over inputs at fixed weights. On the aligned manifold $\Delta = 0$, $\tau_{\text{pot}} \dot{Q} = \overline{G}(Q) - (\tau_{\text{pot}}/\tau_{\text{dec}}) Q$; linearizing at an equilibrium Q^* with return gain $D_Q \overline{G}_*$, the common mode is stable iff the return-gain spectral abscissa stays below the decay-to-potentiation ratio:

$$\max_i \operatorname{Re} \lambda_i\big(D_Q \overline{G}_*\big) \;<\; \frac{\tau_{\text{pot}}}{\tau_{\text{dec}}} \tag{7}$$

KP fixes the alignment eigenvalues; the scale eigenvalues are left to the surrounding circuit's return gain. The bound tightens as the timescales separate: the wider the gap $\tau_{\text{pot}} \ll \tau_{\text{dec}}$, the smaller the admissible ratio.

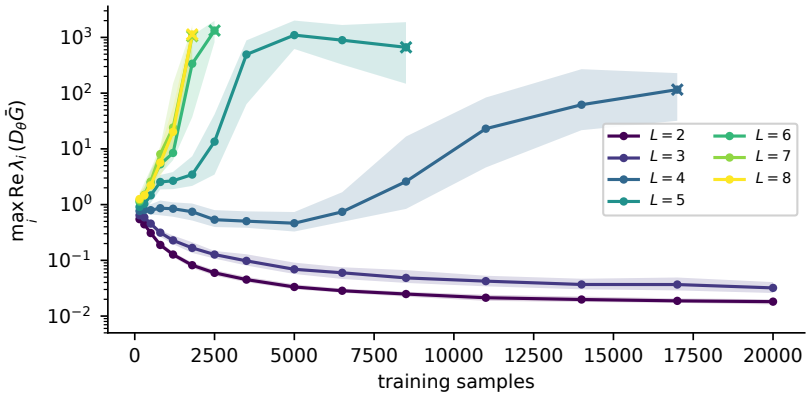


Figure 2: Return-gain spectral abscissa $\max_i \mathrm{Re}\,\lambda_i(D_\theta \overline{G})$ along training for continuous-time KP at depths $L = 2, \ldots, 8$ (UCI digits-8×8, 100 seeds per depth; median solid, p_{25}–p_{75} shaded, $\times$ where over half the seeds diverge). Shallow networks stay bounded; from $L \gtrsim 4$ the abscissa runs away.

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 [8] 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 return-gain runaway can be measured directly. Figure 2 tracks 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 $\theta = (W, V)$ 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 $\max_i \mathrm{Re}\,\lambda_i(D_\theta \overline{G})$ equals the common-mode abscissa $\max_i \mathrm{Re}\,\lambda_i(D_Q \overline{G})$ that Condition (7) 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.

Kunin et al. [4] show that the weight-mirror rule's instability is caused by a common (symmetric) weight mode, not forward–backward misalignment, and that it worsens with depth. We recover this for Kolen–Pollack as a closed-loop return-gain condition in continuous time, and we stabilize it by bounding the drive rather than by adding regularization.

Bounding the return gain. The abscissa grows with the magnitudes of the activities y_l and errors z_l that enter the drive, so it can be controlled locally by bounding those per-layer signals. We compare two mechanisms. *RMSNorm* [12] pins each $\|y_l\|, \|z_l\|$ to a fixed root-mean-square; it is an idealized reference, but a layer-level operation with no local realization in a biological or analog substrate. A strictly per-neuron *bounded activation* instead caps each unit,

$$\phi_l^{\mathrm{sat}}(s) = \mathrm{clip}\big(\phi_l(s),\, -c,\, +c\big) \quad (c = 1), \tag{8}$$

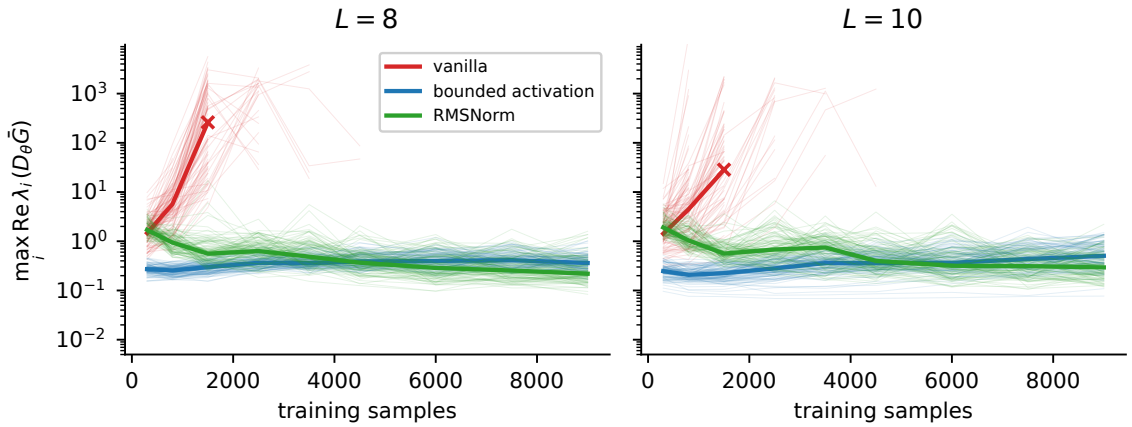


Figure 3: Return-gain abscissa $\max_i \operatorname{Re} \lambda_i(D_\theta \overline{G})$ along training, $L = 8$ (left) and $L = 10$ (right), 100 seeds per condition (UCI digits-8×8). Vanilla KP (red) runs away until the weights overflow ($\times$); both bounded activations ($c = 1$; blue) and RMSNorm ($\alpha = 0.5$; green) hold the abscissa bounded.

giving $\|y_l\|_2 \leq c\sqrt{n_l}$ and gating the backward derivative at saturation (damping $\|z_l\|$), with no cross-unit aggregation. Both shrink the entries of $D_\theta \overline{G}$ and hold its abscissa down rather than letting it run away.

Both control the abscissa; bounded activations train better. Figure 3 tracks the abscissa across 100 seeds at $L = 8, 10$: vanilla KP runs away until the weights overflow, while bounded activations and RMSNorm each hold it bounded within one to two orders of magnitude of the threshold, the measured counterpart of the magnitude argument above. On MNIST (Figure 4) bounded activations reach higher accuracy than RMSNorm at every depth (≈ 0.94 vs ≈ 0.60 at $L = 5$; ahead at $L = 8$ and $L = 10$, where both degrade with large seed-to-seed spread), even though both drive $\cos(V_l^\top, W_{l+1}) \to 1$ and keep $\|W\|_F \approx \|V\|_F$ throughout, in agreement with the contraction guarantee of §4.

5. Discussion

In heterosynaptic learning circuits, alignment is not stability. Once a plastic weight sits in a feedback loop, perturbing it changes the very signals that drive its own update, so stable learning depends on the closed-loop return gain, not on the local rule alone. Continuous-time Kolen–Pollack makes this concrete: its reciprocal symmetry solves feedback alignment exactly, yet leaves the weight-scale mode unstable. This lens reframes the BatchNorm/weight-decay/clipping stack of prior KP-like systems as load-bearing gain control rather than mere convenience.

Two caveats bound these claims. First, the analysis is linear; in practice the abscissa is supercritical at every depth we tested, so the operative distinction is bounded versus runaway. Second, for high accuracy, gain control is necessary but not sufficient: a separate, depth-induced signal-starvation effect—cascade lag through the depth-L chain of low-pass filters—caps trainable accuracy at $L = 10$ and drives the broad seed spread there, independent of the scale-control mechanism (see

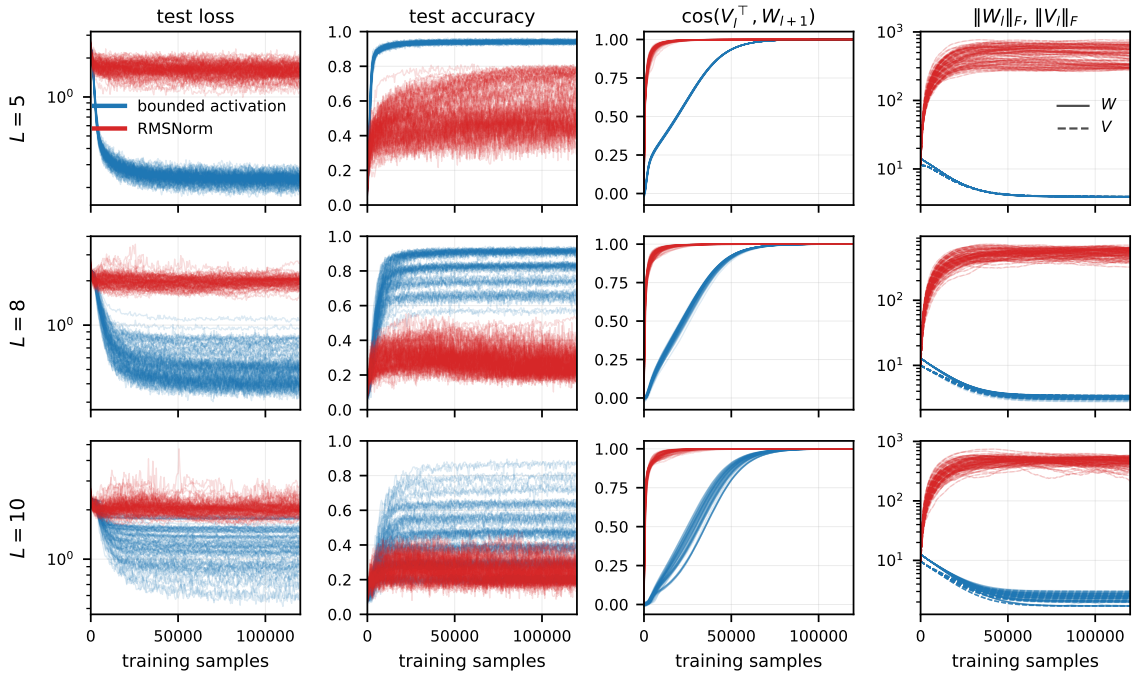


Figure 4: Continuous-time KP on MNIST at $L = 5$ (top), 8 (middle), 10 (bottom): bounded activations (blue) vs RMSNorm (red), 100 seeds per condition (one line per seed). Columns: test loss (log), test accuracy, mean alignment $\cos(V_l^\top, W_{l+1})$, and $\|W_l\|_F$ (solid)/$\|V_l\|_F$ (dashed, log). Bounded activations reach higher accuracy at every depth (≈ 0.94 at $L = 5$) even though both attain alignment ≈ 1 and matched $\|W\|_F \approx \|V\|_F$.

Appendix D). This is a separate, abscissa-independent failure mode, regardless of how well the return 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 closed-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 return-gain lens, and the same need for explicit gain control, to apply to other HSP circuits and to learning in physical substrates more broadly.

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] 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*, pages 5511–5521, 2020. URL https://proceedings.mlr.press/v119/kunin20a.html.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

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

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

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

[13] 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 timescale hierarchy 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 $\tau_{\text{inf}}, \tau_{\text{err}}$ are five orders of magnitude smaller. Setting $\dot{y}_l = \dot{z}_l = 0$ in Eq. (4) for fixed $\{W_l, V_l\}$ gives the slow-manifold values

$$y_l^* \;=\; \phi_l\big(W_l\, y_{l-1}^*\big), \qquad z_l^* \;=\; \phi_l'(s_l^*) \odot \big(V_l\, z_{l+1}^*\big), \qquad z_L^* \;=\; y^* - \text{softmax}(s_L^*). \tag{9}$$

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. (9) reduces to the ordinary backpropagation recursion

$$z_l^* \;=\; \phi_l'(s_l^*) \odot W_{l+1}^\top z_{l+1}^*. \tag{10}$$

Substituting the slow-manifold values into the plasticity equations (5) 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 $\eta_l = 1/\tau_{\text{pot}}$. 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 actually holding during training, which itself depends on the plasticity drive remaining bounded against the decay term.

Appendix B. Computing the return-gain spectral abscissa

Figures 2 and 3 plot 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 $\theta = (W_l, V_l)$, evaluated at the network's actual weights along training. As we now show, it equals the common-mode return-gain abscissa $\max_i \operatorname{Re}\lambda_i(D_Q \overline{G})$ of §4 that Condition (7) constrains:

$$\max_i \operatorname{Re}\lambda_i(M), \qquad M \;=\; D_\theta \overline{G}. \tag{11}$$

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 $\dot{\Delta}_l = -\Delta_l/\tau_{\text{dec}}$ (6) 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 $-1/\tau_{\text{dec}}$ modes of Eq. (6) once the decay term is restored), so $\max_i \operatorname{Re}\lambda_i(M) = \max\big(\max_i \operatorname{Re}\lambda_i(D_Q\overline{G}),\, 0\big)$; the two abscissae coincide whenever the common mode is supercritical, as it is at every depth we measure. We therefore report $\max_i \operatorname{Re}\lambda_i(D_\theta\overline{G})$ evaluated at the actual, generally unaligned weights, equal in the aligned limit $\Delta \to 0$ to the manifold gain $\max_i \operatorname{Re}\lambda_i(D_Q\overline{G}_*)$ 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 (9) 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} (8), 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:

$$Mv \;=\; \tfrac{d}{d\epsilon}\,\overline{G}(\theta+\epsilon v)\big|_{\epsilon=0} \text{ (forward mode, a Jacobian–vector product),} \tag{12}$$

$$M^\top v \text{ (reverse mode, a vector–Jacobian product),} \tag{13}$$

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 [10], 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 $Mu = \lambda u$,

$$\mathrm{Re}\,\lambda \;=\; \mathrm{Re}\,(u^*Mu) \;=\; u^*Su \;\le\; \lambda_{\max}(S) \;=:\; \nu, \tag{14}$$

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 [5]); 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 [9], a Krylov *linear* solver that again uses only the product Mv. 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 $\sim 10^{-8}$.

Appendix C. Seed distribution across depth

The seed-to-seed variability in Figure 4 grows with depth. Figure 5 shows the test-accuracy percentile envelopes across the 100 seeds at each depth for both mechanisms. With bounded activations the

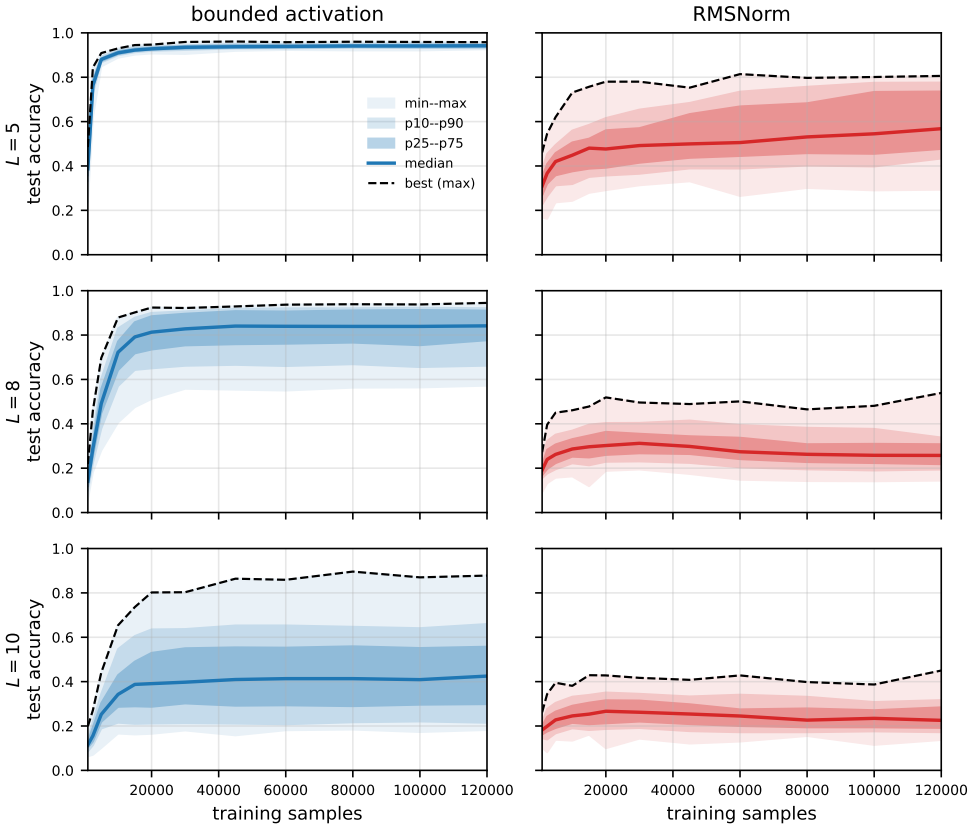


Figure 5: Test-accuracy percentile envelopes across 100 seeds vs. training samples, for bounded activations (left) and RMSNorm (right) at $L = 5, 8, 10$ (rows). Bands are the min–max, p_{10}–p_{90}, and p_{25}–p_{75} envelopes; the solid line is the median and the dashed black line the per-sample maximum (best seed). Bounded activations are essentially deterministic at $L = 5$ and broaden with depth: wide and roughly flat at $L = 10$ (median 0.43, best 0.88), while RMSNorm stays lower and narrower throughout.

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.

The end-of-training percentile summary for bounded activations is given in Table 1. We report percentiles rather than a mean $\pm$ std because the distribution is not Gaussian: at $L = 10$ it is broad and roughly flat over $[0.18, 0.88]$ with no sharp ceiling. The $L = 10$ result is closer to a basin-finding probability than a mean accuracy: the network learns well from a fraction of initializations and poorly from the rest, with the fraction fixed early in training; whereas at $L = 5$ it is essentially deterministic ($[0.93, 0.96]$ across all 100 seeds). Inspecting the per-class accuracies makes the basin structure concrete: at $L = 8$ and $L = 10$ each seed learns an integer number of digit classes nearly perfectly (recall $\gtrsim 0.8$) and leaves the rest near zero, so its test accuracy falls close to a multiple of

	min	p_{10}	p_{25}	median	p_{75}	p_{90}	max
$L=5$	0.926	0.931	0.936	0.942	0.946	0.950	0.958
$L=8$	0.572	0.662	0.777	0.841	0.910	0.921	0.945
$L=10$	0.182	0.215	0.299	0.425	0.558	0.660	0.878

Table 1: End-of-training (120,000 samples) test-accuracy percentiles across 100 seeds with bounded activations. $L = 5$ is essentially deterministic; the distribution broadens with depth, becoming wide and roughly flat at $L = 10$ (best seed 0.88).

the per-class fraction (≈ 0.1); these are the discrete bands visible in the per-seed accuracy traces of Figure 4. The broad $L = 10$ distribution is therefore the seed-to-seed spread in *how many* classes a basin captures, and the cascade-lag signal starvation of Appendix D is what caps that count at depth, independent of the scale-control mechanism.

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

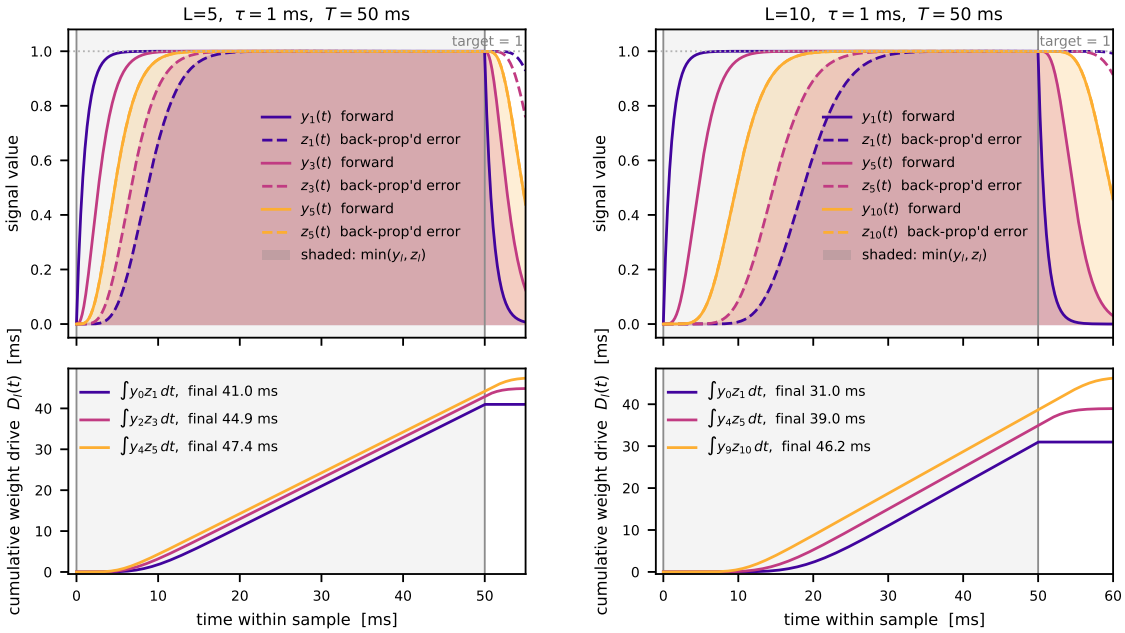
Failure mode 1: alignment without training. The weight-scale dynamics of §4 already show that vanilla continuous-time KP can drive $\Delta_l \to 0$ while Q_l blows up: the mismatch contracts and the weight scale runs away simultaneously. The loss panel of Figure 4 shows a milder version of the same phenomenon 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 ≈ 1.6–2.0 (depending on depth), far above the bounded-activation loss (e.g. ≈ 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 $y^* - \mathrm{softmax}(s_L)$, 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

$$D_l \;=\; \int_0^{T_{\text{sample}}} y_{l-1}(t)\, z_l(t)\, dt \tag{15}$$

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 6 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 ($D_1 \approx 41\,\text{ms}$, $D_5 \approx 47\,\text{ms}$ over the



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

Figure 6: 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.

$50\,\text{ms}$ 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 $D_1 \approx 31\,\text{ms}$ 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 return-gain condition (7) 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 return 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.