COMMON-MODE COLLAPSE AND RECOVERY IN DIRECT FEEDBACK ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Direct feedback alignment (DFA) trains hidden layers through fixed random projections of output error. With tanh hidden units and independent sigmoid outputs, plain stochastic gradient descent can stall near the loss of a constant predictor of class frequencies. We trace this stall to the error's common mode, the component shared across inputs. An exact mean–covariance decomposition separates a rank-one update formed by the mean teaching signal and mean presynaptic activity. Its leading component drives tanh units toward saturation. At initialization, random feedback provides no systematic correction of the shared error on average; readout learning limits its duration. A reduced model initialized from the network, without fitted parameters, predicts the concentration of activation sensitivity across 48 settings. On MNIST, class decodability largely survives collapse, but readout learning remains slow at a fixed learning rate. Adam learns faster despite deeper collapse. Calibrating the baseline readout to the class prior suppresses collapse and speeds learning; weaker feedback trades less collapse for slower learning. Replacing errors by their signs sustains collapse; subtracting the signal's batch mean prevents sustained collapse and improves learning in the tested setting. Related effects occur in deeper and convolutional networks and on CIFAR-10, with severity and cost depending on the readout, optimizer and input statistics.

1 INTRODUCTION

How neural circuits assign errors to synapses is a central problem in theories of learning (Lillicrap et al., 2020). Backpropagation (BP) uses downstream weights to compute hidden error signals (Rumelhart et al., 1986). Direct feedback alignment (DFA) instead broadcasts output error through fixed random matrices (Nøkland, 2016). Like layerwise feedback alignment (FA), it avoids copying forward weights into the feedback pathway (Lillicrap et al., 2016). Its performance varies across tasks and architectures (Bartunov et al., 2018; Launay et al., 2020). In particular, saturating tanh derivatives can weaken DFA gradients (Launay et al., 2019), and plateaus or low-rank updates can arise (Refinetti et al., 2021; Boeshertz et al., 2026). We ask how a broadcast teaching signal can stall learning.

The original DFA experiments used independent sigmoid outputs with binary cross-entropy, and tanh hidden units to bound hidden updates that persist while the error is nonzero (Nøkland, 2016). With one-hot targets, initial predictions near one half leave a large mean error: a *common mode* shared across inputs. Such an error arises whenever the mean prediction differs from the mean target, as can occur with imbalanced softmax or multi-label readouts. Fixed feedback sends it to each hidden unit with a different random coefficient. Together with mean presynaptic activity and trainable biases, this signal pushes tanh units toward saturation. We call the resulting suppression of activity differences across inputs *representation collapse*. It occurs early and across classes, unlike the late within-class convergence studied as neural collapse (Papyan et al., 2020); class information can nevertheless remain decodable. As the output layer, or *readout*, learns the class frequencies (the *class prior*), the mean drive weakens, but input-dependent learning can remain slow for hundreds of updates.

We derive a reduced model from the mean–covariance decomposition (Section 3) and test its predictions across learning conditions (Section 4). Frozen-feature experiments distinguish decodability from learning speed (Section 5). We then assess the mechanism's scope and the costs of interventions across learning rules, architectures and readouts (Sections 6–7).

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.

2 SETTING AND THE PHENOMENON

We first define the learning signals and activity measurements needed to describe collapse and recovery (Figure 1a–c).

Network and learning rules. In a feedforward network, layer ℓ has d_ℓ units, weights W_ℓ, biases b_ℓ, total inputs $a_\ell = W_\ell h_{\ell-1} + b_\ell$ (*preactivations*) and activities $h_\ell = \phi(a_\ell)$. Here ϕ is the activation function, $h_0 = x$, and L is the number of hidden layers. The readout forms logits $z = W_{\text{out}} h_L + b_{\text{out}}$, the values before the output nonlinearity. For C outputs, let $\hat{y}(x) \in \mathbb{R}^C$ denote the prediction and $y(x) \in \mathbb{R}^C$ the target. We use independent sigmoids with binary cross-entropy (BCE) summed over outputs, softmax with cross-entropy, or a linear readout with half squared error. Writing the per-example loss as $\mathcal{L}(x)$, the logit error is $e(x) = \partial \mathcal{L}(x)/\partial z = \hat{y}(x) - y(x)$. With stochastic gradient descent (SGD), DFA projects this error through a fixed matrix $B_\ell \in \mathbb{R}^{d_\ell \times C}$ and multiplies it by the activation derivative, or *gate*. The resulting hidden teaching signal δ_ℓ determines each local update:

$$\begin{aligned} \delta_\ell(x) &= \phi'(a_\ell(x)) \odot B_\ell e(x), \\ \Delta W_\ell &= -\eta \langle \delta_\ell h_{\ell-1}^\top \rangle, \qquad \Delta b_\ell = -\eta \langle \delta_\ell \rangle. \end{aligned} \tag{1}$$

Here Δ denotes a change in one update, $\odot$ denotes elementwise multiplication, $\langle \cdot \rangle$ is a minibatch mean, and $\eta = \eta_{\text{hid}}$ is the hidden learning rate. Unless stated otherwise, entries of B_ℓ are independent $\mathcal{N}(0, g^2)$ draws, where g is the feedback gain. The output layer receives its true gradient at learning rate η_{out}. BP propagates error through the transposed forward weights; layerwise FA uses fixed random matrices between successive layers (Lillicrap et al., 2016).

The baseline is MNIST (LeCun et al., 1998a), with pixels in $[0, 1]$, a multilayer perceptron (MLP) with three tanh layers of width $d = 300$, Xavier-uniform weights (Glorot & Bengio, 2010), zero biases, $g = 1$, and SGD with $\eta = \eta_{\text{out}} = 10^{-3}$ and batch size 128. The readout, loss, hidden nonlinearity and input scaling follow the original DFA experiments (Nøkland, 2016) and direct random target projection (DRTP) (Frenkel et al., 2021), which used adaptive optimizers and other initializations; in our reconstruction of Nøkland's protocol the DFA stall lasts only tens of updates, and at the larger tested rate BP also saturates (Appendix E.3). Diagnostics use a fixed 2,048-example validation set, called the *probe*. Appendix A gives the splits, architectures and protocol.

Measurements. Here $\mathbb{E}_x$ averages over probe inputs; $\|\cdot\|$ is the Euclidean norm and $\|\cdot\|_F$ the Frobenius norm of a matrix. We write $\bar{e}$ for mean error and $\tilde{e}(x) = e(x) - \bar{e}$ for its input-dependent part. We describe a representation by its mean pairwise cosine $\cos_\ell$ across inputs and its mean-activity energy $H_\ell = \|\bar{h}_\ell\|^2$, with $\bar{h}_\ell = \mathbb{E}_x h_\ell$. For unit i, let $\mu_{\ell i} = \mathbb{E}_x a_{\ell i}$ and $\sigma_{\ell i}^2 = \text{Var}_x(a_{\ell i})$; its *gate energy* $u_{\ell i} = \mathbb{E}_x[\phi'(a_{\ell i})^2]$ is its mean squared sensitivity. For nonzero total gate energy, the gate participation ratio

$$p_\ell = \frac{(\sum_i u_{\ell i})^2}{d_\ell \sum_i u_{\ell i}^2} \tag{2}$$

equals one when gate energies are equal and approaches $1/d_\ell$ when one unit dominates; equally weak gates also give high participation, so it does not measure feedback strength. We track activity–target covariance $\Lambda_\ell = \text{Cov}(h_\ell, y)$, through its energy $\|\Lambda_\ell\|_F^2$. Covariances average centered outer products over the probe for diagnostics and over the minibatch for updates. To assess useful learning, we compare the DFA gradient estimate with the true BP weight gradient: their cosine $\cos \alpha_\ell$ measures direction, and projected descent Π_ℓ (Safaai et al., 2026), the cosine times the DFA-to-BP norm ratio, predicts a first-order loss decrease from the hidden weight update when positive (Appendix A.2).

The baseline trajectory. DFA loss rapidly approaches that of a predictor returning the class frequencies and plateaus for several hundred steps before decreasing again; accuracy remains near chance during the plateau (Figure 1d). For sigmoid outputs, this prior loss is $\mathcal{L}_{\text{prior}} = \sum_c \mathcal{H}(\pi_c)$, where $\pi = (\pi_c)$ is the vector of class frequencies and $\mathcal{H}(q) = -q \ln q - (1-q)\ln(1-q)$ is binary entropy. At this learning rate, matched BP passes through this loss without pausing but learns more slowly later. Along the DFA trajectory, raw DFA weight-gradient estimates are several times larger than BP gradients evaluated at the same network state (Figure 9d).

During the plateau, top-layer hidden vectors become nearly parallel, unlike those of the matched BP network (Figure 1e). After subtracting each unit's mean activity, the minimum variance across

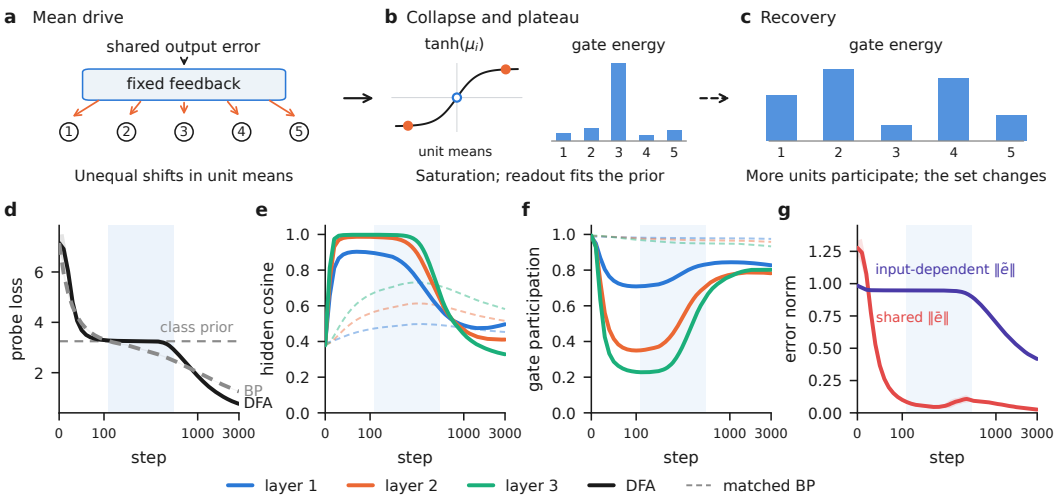

Figure 1: **Collapse and recovery.** (a–c) Schematic: shared error shifts unit means toward tanh saturation (b: open, initial; filled, shifted), concentrating gate energy. More units participate during recovery and their identities change. Points and bars are illustrative; numbers identify units. The solid arrow represents the approximate collapse mechanism, the dashed arrow observed recovery beyond the model. (d) Probe loss for DFA and matched BP (same initialization and minibatches); the horizontal line is the class-prior loss and shading marks the mean detected plateau interval, repeated in e–g. (e) Mean pairwise hidden cosine and (f) gate participation per layer; colours denote layers, solid curves DFA and dashed curves BP. (g) Shared-error norm $\|\bar{e}\|$ and root-mean-square input-dependent error, both measured on the fixed probe. Time is linear through step 150 and logarithmic thereafter. Curves average 15 DFA trajectories and five BP networks; bands in d and g show one standard deviation.

inputs averages 0.028 of its initial value, against 0.83 for BP (Appendix C.1). Gate participation falls in every hidden layer, most strongly in the deeper layers (panel f), and the top-layer DFA gradient estimate is slightly anti-aligned with the BP gradient while the readout supplies the loss improvement (Figure 4c). As training recovers, cosine declines and participation rebounds. The shared error norm falls from 1.278 to 0.10 within 100 updates, whereas the root-mean-square norm of input-dependent error stays between 0.90 and 0.95 throughout the plateau (panel g). We now derive how the shared component drives collapse.

3 Theory: The Common-Mode Term

To explain collapse, we separate means from variation across inputs: $e(x) = \bar{e} + \tilde{e}(x)$ and $h_{\ell-1}(x) = \bar{h}_{\ell-1} + \tilde{h}_{\ell-1}(x)$, where tildes denote deviations from the batch mean. Write the derivative gate as $\gamma_\ell(x) = \phi'(a_\ell(x))$, with mean $\bar{\gamma}_\ell$. This exact separation forms the basis of an approximate dynamical model.

3.1 Exact Decomposition and the Common-Mode Closure

Proposition 1 (Mean and covariance parts of the DFA update)**.** *For any minibatch,*

$$\begin{aligned} \Delta W_\ell &= -\eta \operatorname{Cov}(\delta_\ell, h_{\ell-1}) - \eta \bar{\delta}_\ell \bar{h}_{\ell-1}^\top, \\ \bar{\delta}_\ell &= \bar{\gamma}_\ell \odot B_\ell \bar{e} + r_\ell, \qquad r_\ell = \big\langle (\gamma_\ell - \bar{\gamma}_\ell) \odot B_\ell \tilde{e} \big\rangle, \end{aligned} \tag{3}$$

and $\Delta b_\ell = -\eta \bar{\delta}_\ell$. The second weight-update term has rank at most one. At initialization, conditional on $\bar{e}$, Gaussian feedback independent of the forward network gives independent entries $(B_\ell \bar{e})_i \sim \mathcal{N}(0, g^2 \|\bar{e}\|^2)$.

This is the classical mean–covariance decomposition of an outer product (Sejnowski, 1977). The residual r_ℓ measures correlations between gate fluctuations and input-dependent error, which are

weak at initialization in the baseline. A separate approximation treats the mean-error direction as nearly fixed during the early, approximately class-symmetric transient. Neglecting r_ℓ gives the *common-mode closure*: the mean gated error is approximated by the product of mean gate and mean broadcast error. On the probe, $\|r_\ell\|$ is 0.01 of the leading term at initialization but exceeds it by step 300 (Appendix C.3): the closure describes the onset of collapse, not the plateau.

The rank-one update shifts preactivation means and can also change their spread. Holding presynaptic activity fixed, we neglect spread changes, local covariance and the gate–error residual. With the tanh approximation $\bar{\gamma}_{\ell i} \approx \operatorname{sech}^2 \mu_{\ell i}$, this gives

$$\frac{d\mu_{\ell i}}{dt} = -\eta \operatorname{sech}^2(\mu_{\ell i})(B_\ell \bar{e})_i (H_{\ell-1} + 1). \tag{4}$$

Here t counts updates, $\operatorname{sech} \mu = 1/\cosh \mu$, and $+1$ comes from the bias update. Saturation slows fixed-direction drift to logarithmic growth. In baseline layer 3, upstream motion dominates the initial displacement; the mean term becomes dominant during early saturation (Appendix C.3). Exactly centering $h_{\ell-1}$ and freezing the postsynaptic bias b_ℓ removes local mean drift when presynaptic activity is fixed.

3.2 Why the drive persists, and the collapse number

Collapse also depends on how long the mean error acts. We combine the readout update with the local top-layer drift, using the same closure and omitting upstream motion and covariance terms. Let $\bar{\Gamma}_L = \operatorname{diag}(\bar{\gamma}_L)$ collect the mean derivative gates and I be the $C \times C$ identity. The mean logits then obey

$$\frac{d\bar{z}}{dt} \approx -\Big[\eta_{\text{out}}(H_L + 1)I + \eta(H_{L-1} + 1)\, W_{\text{out}} \bar{\Gamma}_L^2 B_L\Big]\bar{e}. \tag{5}$$

For a positive multiple $B_L \propto W_{\text{out}}^\top$, the hidden-layer matrix is positive semidefinite: within this approximation, hidden drift moves the mean logits in a direction that lowers the mean-only loss. This relation holds for BP and initially for pre-aligned DFA. Random feedback has no systematic restoring effect at initialization when averaged over feedback draws; the readout supplies the systematic restoring term in this approximation. The shared error falls more quickly under pre-aligned feedback and BP with an amplified readout, consistent with this distinction (Appendix C.1).

Keeping only the readout term, balanced classes with $C > 2$, zero initial logits and constant H_L give an effective slope for the sigmoid $\sigma(z) = (1 + e^{-z})^{-1}$, $\sigma'_{\text{eff}} = (1/2 - 1/C)/\ln(C - 1)$ and initial error norm $\|\bar{e}_0\| = \sqrt{C}(1/2 - 1/C)$ (Appendix B.3). Together with the drift, this gives the initialization estimate

$$\hat{\kappa}_\ell = \frac{g\|\bar{e}_0\|(H_{\ell-1}(0) + 1)}{\sigma'_{\text{eff}}(H_L(0) + 1)(\eta_{\text{out}}/\eta)}. \tag{6}$$

This dimensionless *collapse number* is unchanged by scaling both learning rates; at fixed effective output slope, gain and initial mean error enter through their product. It ignores the evolution of activity energies and input-dependent learning. We therefore also use the reduced-model prediction below and the measured *delivered dose* along the initial mean-error direction,

$$\kappa_\ell(t) = g \sum_{s<t} \eta \langle \bar{e}_s, \hat{e}_0 \rangle (H_{\ell-1}(s) + 1), \qquad \hat{e}_0 = \bar{e}_0/\|\bar{e}_0\|, \tag{7}$$

where s indexes updates, $\hat{e}_0$ is the unit initial-error vector, and $\langle u, v \rangle = u^\top v$ denotes a vector inner product, distinct from the minibatch average $\langle \cdot \rangle$. For comparisons of initial collapse, the *transient dose* stops this sum when the probe mean-error norm falls to 15% of its initial value (Appendix B.3).

In deeper layers, saturation of presynaptic activity also narrows the within-unit preactivation distributions. For small spread, large width and negligible initial means, averaging over unit means gives $p_\ell \sim (15/8)\sqrt{2/\pi}/\kappa_\ell$ at large dose, bounded below by $1/d_\ell$ (Appendix B.4). Gaussian reconstructions from measured means and spreads reproduce early participation (Appendix D.2).

3.3 Coupled dynamics across layers

The initialization estimate reverses the measured ordering of dose across layers because activity energies evolve: as a layer saturates, H_ℓ grows toward its width, increasing the drive on the next layer and accelerating the readout's learning of the prior.

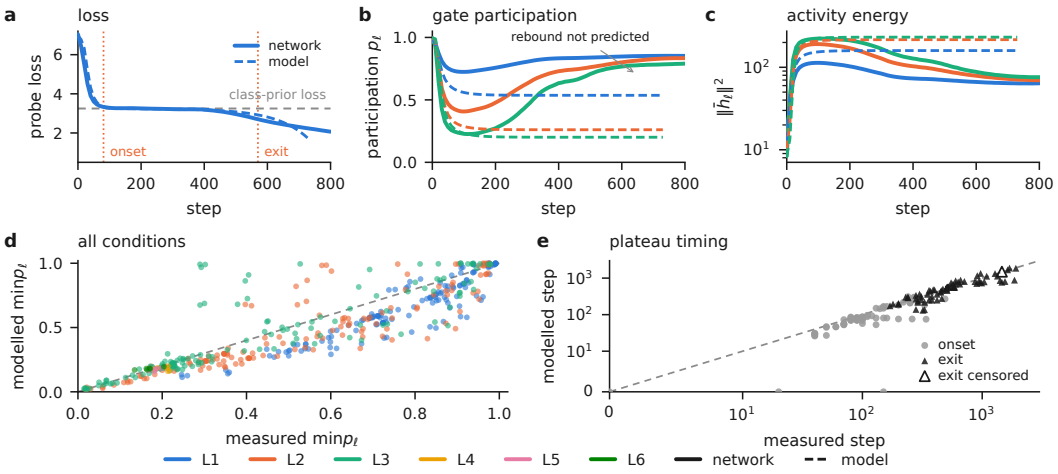

Figure 2: **The model predicts early concentration and approximate exit times, but not participation recovery.** (a–c) Model (dashed) and matched network (solid), both using initialization/feedback seeds $(0, 0)$: loss, gate participation and mean-activity energy. Dotted vertical lines in a mark predicted onset and exit, the exit from the covariance extension of Section 3.4. (d) Participation minima within the first 300 updates across 48 settings and three seed pairs; colours denote layers (L1–L6). Points with model values near one but low measured minima are two-class and squared-error settings, for which the model predicts almost no drive. (e) Paired onset and exit times; the open triangle gives a lower bound for an exit not reached within the model horizon. Diagonals in d,e indicate equality.

Gaussian quadrature computes mean activities and gates by averaging over each unit's evolving Gaussian preactivation distribution. The reduced model uses a shared within-unit variance σ_ℓ^2, fixed in layer 1 and propagated as $\sigma_\ell^2 = q_\ell \bar{u}_{\ell-1} \sigma_{\ell-1}^2$ for $\ell \geq 2$. Here $q_\ell = d_{\ell-1} \operatorname{Var}(W_{\ell,ij})$ is fixed at initialization and $\bar{u}_\ell = d_\ell^{-1} \sum_i u_{\ell i}$ is mean gate energy. Retaining only the readout term of Equation (5) closes the system, initialized from the network without fitted parameters (Appendix B.5). It captures the baseline plateau loss, onset and layer ordering of participation minima (Figure 2a–d). The covariance extension below adds exit predictions.

3.4 Recovery and the deeper-but-shorter effect

Ending the mean drive does not immediately restore learning: the network remains close to the prior predictor. Near that predictor, the input-dependent readout gradient is approximately $-\Lambda_L^\top$, so its first-order contribution to loss decrease is $\eta_{\text{out}} \|\Lambda_L\|_F^2$. This energy falls sharply while a scale-normalized statistic changes little (Table 19; Section 5). During recovery, label covariance grows again, beginning in earlier layers. A phenomenological covariance model initialized from $\operatorname{Cov}(x, y)$ predicts when this growth lowers loss (Figure 2e; Appendix B.6).

Coupled changes in features and readout make the plateau duration T_{plateau}, from onset to exit (Appendix A), depend on both rates. Under plain SGD, gain and hidden rate enter only through $a = \eta_{\text{hid}} g$, and a fit to the detected plateaus gives

$$T_{\text{plateau}} \propto a^{-0.40} \, \eta_{\text{out}}^{-0.60} \tag{8}$$

(Table 17). The exponents sum to approximately -1, as the measured $1/\eta$ scaling requires when both rates change together. Over this tested range, increasing gain deepens the initial collapse (Section 4) while shortening the plateau. Recovery also changes the gates: participation rebounds, the participating units turn over, and the within-unit spread grows; the reduced model does not reproduce these features (Appendix D.3).

4 Predictions and their tests

The account predicts that the strength and duration of the shared drive affect collapse differently. We test this by varying feedback gain, learning rates, initial predictions and input means (Appendix C).

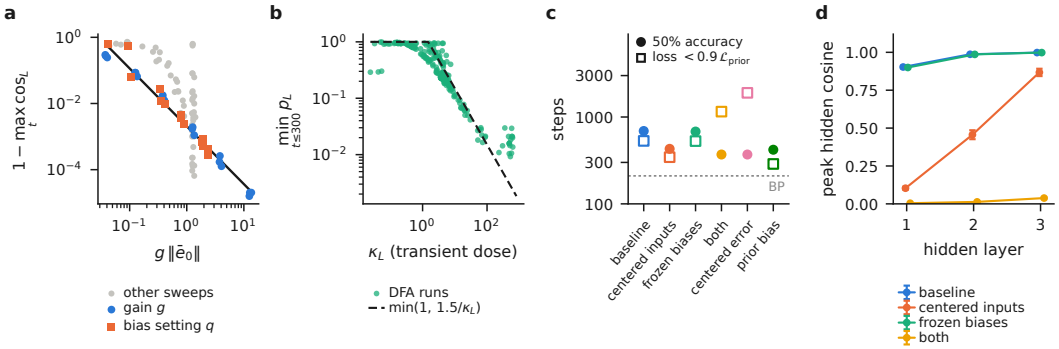

Figure 3: **Scaling, learning cost and mean-channel controls.** (a) $1 - \max_t \cos_L$ against $g\|\bar{e}_0\|$: smaller values indicate higher cosine. Gain and initial-prediction sweeps are coloured, other sweeps grey; the line fits the coloured points, counting their shared baseline once. (b) Deepest-layer participation minimum within 300 updates against measured transient dose; dashed line: large-dose prediction capped at one. (c) Steps to 50% probe accuracy (filled; median) and to a sustained loss below $0.9\mathcal{L}_{\text{prior}}$ (open; mean) for the controls in d, centered output error and prior-bias initialization; dotted line: BP at 50% accuracy. (d) Peak cosine in each layer over the first 800 updates: input centering and frozen hidden biases jointly prevent collapse. Error bars in d: one standard deviation; trajectories in Figure 5a–d.

Reduced-model predictions. Across 48 distinct settings, the condition-averaged correlation between predicted and measured participation minima is $r = 0.91$ (mean absolute error 0.10). The model outperforms the initialization estimate $\min(1, 1.5/\hat{\kappa}_\ell)$ obtained from Equation (6) and the large-dose asymptote (Table 13). Its errors reveal the approximation's limits: it overestimates shallow-layer collapse in the baseline (Figure 2b) and misses concentration in two-class and squared-error settings (panel d). Table 14 retains detection failures and censored exits. The recovery extension predicts broad variation in exit times (panel e), including on untested width–depth combinations; removing local covariance growth predicts no exits in those new settings (Appendix C.5).

Learning rate, gain and initial predictions. Changing both learning rates together over a tenfold range leaves peak hidden cosine and participation minima within seed noise, while exit time scales as $1/\eta$ (Appendix C.2): the rates change the trajectory's timescale with little effect on collapse severity. Varying gain and the nominal initial sigmoid probability q, set by output bias $\ln[q/(1-q)]$, changes the drive; for constant initial prediction, $\|\bar{e}_0\| = \sqrt{C}|q - 1/C|$. Both sweeps follow one power-law trend in $g\|\bar{e}_0\|$: stronger drive produces more nearly parallel hidden activities (Figure 3a). This scaling is empirical because q also changes the readout timescale (Appendix B.3), and other sweeps at similar $g\|\bar{e}_0\|$ scatter widely. At large delivered dose, participation decreases on the scale predicted by Gaussian quadrature (panel b).

Readout and input means. Balanced softmax has zero mean error at uniform prediction, and with standardized inputs its collapse is weak (peak cosine 0.271, or 0.55 with raw pixels; Table 28); drawing from class 0 with probability 0.7 and from the original distribution otherwise restores it (0.938, against 0.386 for BP; Table 4). Per-pixel centering sets $\bar{x}$ to zero on the training split. It prevents layer-1 collapse but only reduces deeper collapse. Freezing hidden biases alone has little effect; combining both prevents collapse throughout the network (Figure 3d). Global standardization raises the input mean energy and deepens layer-1 gate concentration despite lowering peak cosine (Appendix C.1).

Learning cost. Collapse delays early discrimination (Figure 3c; Table 5). Centering inputs with frozen biases, or centering output error, prevents collapse and reaches 50% accuracy after 370 rather than 690 updates. However, unsaturated layers keep H_L small (6.0 versus 182 at step 300 under error centering), which slows the readout's fit to the prior, so the time to 80% accuracy improves little. Prior-bias initialization suppresses the drive while speeding both early discrimination and loss reduction, with unchanged final accuracy.

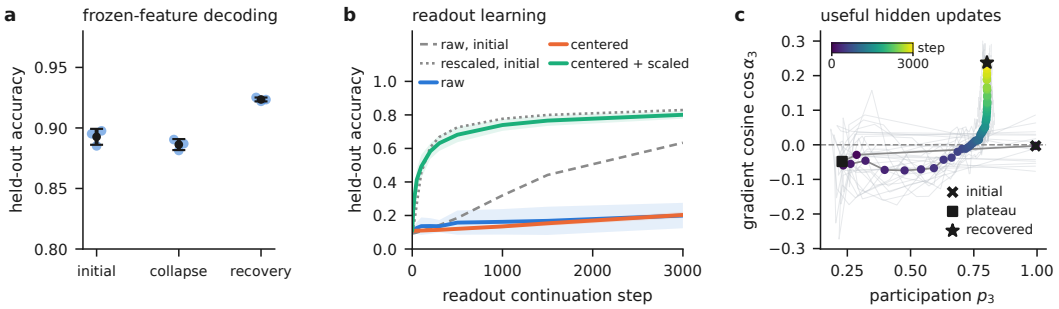

Figure 4: **Decodability, readout learning and useful descent.** (a) Least-squares ridge decoding of one-hot targets on separate evaluation examples at three frozen layer-3 checkpoints. (b) Sigmoid/BCE readout continuation from the collapse checkpoint at nominal SGD rate 10^{-3} with raw, centered and centered/rescaled features; transformed parameters preserve the initial predictions. Grey curves continue the same raw (dashed) and centered, rescaled (dotted) readouts from the initial checkpoint. Curves and error bars in a,b show means and one standard deviation over three seed pairs. (c) Baseline gradient cosine against participation across 15 trajectories: grey curves are individual runs, coloured points follow the mean in time order; markers denote initialization, step 100 and step 3000.

For comparisons across conditions, we define *learning time* as the first probe loss below $0.9\mathcal{L}_{\text{prior}}$ that remains below it for 50 updates. Weaker feedback prevents collapse but increases this learning time: at $g = 0.03$, close to the feedback scale of Nøkland (2016) at this width, minimum participation is 0.98 but learning time is 1897 rather than 537 updates (Appendix C.2). Thus reduced collapse need not mean faster learning: saturation increases mean-activity energy and can accelerate the fit to the prior while weakening input sensitivity. The delay persists at larger learning rates: at $\eta = 0.1$, the largest rate tested, no run diverges, and plain DFA needs 130 updates to reach 80% accuracy, against 28 with a calibrated readout and 20 for BP (Appendix C.2).

Unit drift and feedback orientation. A nearly class-symmetric error predicts drift along $B_\ell \mathbf{1}$, with $\mathbf{1}$ the all-ones vector; measured unit mean shifts are anticorrelated with it in all three layers (Table 6), as Equation (4) predicts. Pre-aligned feedback at the Frobenius norm of the random draw delivers a dose of only 0.12 rather than 4.6 and, like BP with an amplified readout, learns quickly with little collapse (Figure 5f,g), consistent with Equation (5). Reorienting feedback also changes $B_\ell \bar{e}$, so this control does not isolate the restoring contribution.

5 Decodable features can support slow learning

The covariance measurements suggest that collapse weakens learning signals while preserving class structure. We freeze the baseline network at initialization, at its early participation minimum, and after recovery. A least-squares ridge readout fitted to one-hot targets retains 88.6% held-out accuracy at collapse, close to 89.3% at initialization (Figure 4a).

Learning from the collapsed features is nevertheless slow. With identical minibatches and a fixed nominal SGD rate, a sigmoid readout continued on them reaches 20.0% held-out accuracy after 3,000 updates, against 63.4% from the initial features (panel b). Collapsed features are 5.9-fold smaller in root-mean-square (RMS) variation, which lowers the readout's effective learning rate about 35-fold. Centering them changes little, whereas dividing centered features by their RMS and transforming the readout parameters to preserve its initial predictions raises accuracy to 80.1%, close to 82.9% for rescaled initial features. This comparison identifies an amplitude bottleneck at a fixed optimizer scale despite retained class decodability (Appendix D.1). Adam (Kingma & Ba, 2015) normalizes updates coordinatewise and learns faster despite deeper gate concentration. Across three matched seed pairs, its learning time is 117 rather than 537 updates, with minimum participation 0.046 rather than 0.214 (Appendix E.2).

Recovery also involves a change in the direction of hidden-layer updates. Participation recovers while the top-layer gradient cosine is still near zero, and alignment then rises at nearly fixed participation

(panel c). Participation therefore does not determine update direction; describing recovery requires tracking task-dependent covariance and alignment alongside unit sensitivity.

6 Generality and limits

We next ask whether the transient depends on the baseline architecture and task, and what changes when the shared signal persists.

Rules, architectures and objectives. The same signature appears under layerwise FA (peak top-layer cosine 0.991), in a six-layer MLP whose layers 4–6 reach cosine 1.000, and in a small convolutional network (CNN), whose deepest block reaches minimum participation 0.021 against 0.84 for one matched BP run (Figure 12; Appendix E.5). Changing the objective changes the transient: an augmented twelve-output multi-label target also produces a prior-loss plateau, whereas large-target squared error produces only a brief interruption. With rectified linear units (ReLU), mean drift instead leaves many units inactive or always active, more often than in the matched BP control, and across settings the early ratio of mean to input-dependent error correlates with peak cosine (Appendix E.5).

CIFAR-10 and the size of the input mean. On CIFAR-10 (Krizhevsky, 2009), a tanh CNN with a sigmoid readout reaches minimum participation 0.009 under DFA against 0.756 under BP (Table 25); because BP also has high cosine on uncentered images (Figure 13a), participation is the more informative diagnostic here. With standardized inputs, balanced softmax largely avoids collapse, while imbalanced sampling restores it (panels b,c). The larger CIFAR-10 input mean also drives collapse in MLP layer 1 (panel d). Calibrating initial logits toward the class prior reduces mean error without preventing collapse, although prior-bias initialization raises CNN accuracy at step 3000 from 0.276 to 0.349 (BP: 0.343). Combining error and input centering reduces collapse in both networks. CIFAR-100 shows high transient cosine even under softmax with small initial error (Appendix E.7). These controls show why output calibration alone can be insufficient when input means are large.

A common mode that does not decay. For sigmoid outputs and binary targets, sign-error feedback broadcasts $\mathrm{sign}(e) = \mathbf{1} - 2y$. Before gating, this is a fixed target projection, as in DRTP (Frenkel et al., 2021), plus a constant. Its population mean $\mathbf{1} - 2\pi$ does not shrink as the readout learns. With this signal, the MNIST baseline stays collapsed through 3,000 updates (final cosine 0.96, participation 0.05, accuracy 0.75), prior-bias initialization leaves its hidden updates unchanged, and subtracting its batch mean prevents sustained collapse and improves learning (cosine 0.03, accuracy 0.90; Appendix E.4).

7 Interventions and their costs

The mechanism suggests three interventions: subtract the mean error, center presynaptic activity, or shorten the error's duration; Figure 10a–c compares them in three networks.

Centering the broadcast error. Subtracting $\langle e \rangle$ before projection removes the mean-error contributions to the rank-one update and gate covariance (Equation (11)). The gate–error residual can still contribute to the mean teaching signal. Error centering prevents strong convergence of hidden activities in the tanh MNIST baseline (Figure 5e); centering the teaching signal after the derivative gate is less effective. A faster readout offsets the slower prior fit under error centering (Section 4; Appendix A.4). With ReLU, Gaussian error linear units (GELU) (Hendrycks & Gimpel, 2016), or linear units, however, error centering can drive loss far above the constant-predictor baseline as mean activity grows (Appendix E.1). Input centering suppresses this extreme loss growth, although the linear network's BCE remains close to the constant-predictor loss (Table 20).

Calibration and activity centering. Initializing output biases near $\mathrm{logit}(\pi_c) = \ln[\pi_c/(1-\pi_c)]$, the prior initialization used for class-imbalanced detection (Lin et al., 2017), suppresses MNIST collapse and shortens learning; a faster readout shortens the drive. Neither suffices in the CIFAR-10 CNN. On the activity side, per-pixel input centering (LeCun et al., 1998b) protects the first layer,

and batch normalization (Ioffe & Szegedy, 2015) of hidden preactivations reduces collapse; since normalization also rescales and shifts, this does not isolate mean subtraction.

Changing update geometry. Muon-style orthogonalization of hidden updates (Jordan et al., 2024), which Boeshertz et al. (2026) applied to FA, reduces collapse. An adaptation of the error-side conditioner of Safaai et al. (2026), which suppresses dominant teaching-signal directions through an inverse second moment, retains high cosine at every tested ridge regularization strength and, at the default strength, lowers step-3000 accuracy to 0.58 (Figure 10d; Appendix A.4).

8 Related work

Our account concerns the cause and duration of early saturation, a long-recognized source of plateaus in sigmoid networks (Lee et al., 1993; Vitela & Reifman, 1997). Under BP, sigmoid hidden units can saturate while the readout learns its biases, then desaturate slowly (Glorot & Bengio, 2010). Saturating derivatives also weaken tanh DFA gradients (Launay et al., 2019) and can halt learning under DRTP (Frenkel et al., 2021). We identify a mean-driven contribution under random feedback, predict the concentration of sensitivity and approximate exit times, and test interventions on it.

This early transient complements accounts of alignment. Refinetti et al. (2021) establish align-then-memorise dynamics; our stall precedes substantial alignment (Appendix E.8). Boeshertz et al. (2026) find low-rank error and update dynamics in layerwise FA and improve learning through orthogonalization and activity normalization; whether a common mode contributes there is open. Lee & Lee (2026) restore weak weight and gradient alignment when fine-tuning with DFA by constructing feedback from pretrained weights and re-initializing forward weights to match it. Our aligned control instead starts from an untrained network. Safaai et al. (2026) condition the activity and error factors of the DFA update through their second moments; we study their first moments before alignment.

Centering has several antecedents. Nonzero-mean inputs bias gradient updates (LeCun et al., 1998b), and Schraudolph (1998) centers activities, errors and activation slopes. Concurrently, Yamada et al. (2026) subtract the batch mean of one-vs-all errors in a dual-stream network with structured routing, noting its strong class-wise offset early in training; we show how such an offset saturates units under random feedback. The mean–covariance separation is classical (Sejnowski, 1977; Sejnowski & Tesauro, 1989; Dayan & Abbott, 2001), and centered modulatory signals matter in reward-dependent plasticity (Frémaux et al., 2010; Gerstner et al., 2018). The distinction between sensitivity and useful learning connects to dormant units and plasticity loss (Sokar et al., 2023; Lyle et al., 2025).

9 Discussion

The early DFA stall reflects competition between shared drive and input-dependent learning. With random feedback, mean output error can push units toward saturation before the readout learns the class prior. Our local approximation explains why this drift initially lacks the systematic restoring contribution supplied by transposed feedback. Class information remains decodable, but reduced feature variation weakens readout gradients. Learning resumes as covariance grows, a changing set of units regains sensitivity, and updates align with the loss gradient.

The controls suggest ways to limit this transient. Prior-calibrated initialization speeds learning in the MNIST baseline at no extra training cost, but is insufficient on CIFAR-10. Input centering protects early layers and prevents the observed instabilities under error centering with non-saturating units. For the tested sign-error rule, calibration cannot reduce the mean broadcast signal; centering it prevents sustained collapse and improves learning.

Our evidence comes mainly from small networks trained with plain SGD at fixed learning rates. Adaptive optimizers can shorten the transient (Appendices E.2 and E.3). The reduced model captures early collapse but not participation rebound; its recovery extension is phenomenological. These findings support a mechanism for early dynamics, rather than a general account of DFA performance. Whether biological circuits regulate a similar interaction between shared teaching signals and input-dependent plasticity remains open.

Reproducibility statement

The supplementary code contains the training configurations, reduced model, tests and analysis scripts. Appendix A specifies the data splits and measurements; Appendix B states the model assumptions and derivations. Figures and numerical tables are generated from the experiment records.

AI use statement

Generative AI assisted coding, experiment execution, figure preparation, literature search and manuscript editing. The authors reviewed the outputs, checked numerical results and references, and take responsibility for the manuscript and accompanying artifacts.

References

Shun-ichi Amari, Tomoko Ozeki, Ryo Karakida, Yuki Yoshida, and Masato Okada. Dynamics of learning in MLP: Natural gradient and singularity revisited. *Neural Computation*, 30(1):1–33, 2018. doi: 10.1162/neco_a_01029.

Sergey Bartunov, Adam Santoro, Blake A. Richards, Luke Marris, Geoffrey E. Hinton, and Timothy Lillicrap. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Gauthier Boeshertz, Razvan Pascanu, and Claudia Clopath. Overcoming rank collapse in feedback alignment. *arXiv preprint arXiv:2606.11123*, 2026.

Blake Bordelon and Cengiz Pehlevan. The influence of learning rule on representation dynamics in wide neural networks. In *International Conference on Learning Representations*, 2023. arXiv:2210.02157.

Peter Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, Cambridge, MA, 2001.

Nicolas Frémaux, Henning Sprekeler, and Wulfram Gerstner. Functional requirements for reward-modulated spike-timing-dependent plasticity. *The Journal of Neuroscience*, 30(40):13326–13337, 2010. doi: 10.1523/JNEUROSCI.6249-09.2010.

Charlotte Frenkel, Martin Lefebvre, and David Bol. Learning without feedback: Fixed random learning signals allow for feedforward training of deep neural networks. *Frontiers in Neuroscience*, 15:629892, 2021. doi: 10.3389/fnins.2021.629892.

Kenji Fukumizu and Shun-ichi Amari. Local minima and plateaus in hierarchical structures of multilayer perceptrons. *Neural Networks*, 13(3):317–327, 2000. doi: 10.1016/S0893-6080(00)00009-5.

Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: Experimental support of NeoHebbian three-factor learning rules. *Frontiers in Neural Circuits*, 12:53, 2018. doi: 10.3389/fncir.2018.00053.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256. PMLR, 2010.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456. PMLR, 2015.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022. arXiv:2110.09348.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. Blog post, 2024. https://kellerjordan.github.io/posts/muon/.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. arXiv:1412.6980.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Julien Launay, Iacopo Poli, and Florent Krzakala. Principled training of neural networks with direct feedback alignment. *arXiv preprint arXiv:1906.04554*, 2019.

Julien Launay, Iacopo Poli, François Boniface, and Florent Krzakala. Direct feedback alignment scales to modern deep learning tasks and architectures. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9346–9360. Curran Associates, Inc., 2020.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998a. doi: 10.1109/5.726791.

Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient BackProp. In Genevieve B. Orr and Klaus-Robert Müller (eds.), *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, pp. 9–50. Springer, 1998b. doi: 10.1007/3-540-49430-8_2.

Youngjik Lee, Sang-Hoon Oh, and Myung Won Kim. An analysis of premature saturation in back propagation learning. *Neural Networks*, 6(5):719–728, 1993. doi: 10.1016/S0893-6080(05)80116-9.

Yunseok Lee and Seulki Lee. Enabling fine-tuning of Direct Feedback Alignment via feedback-weight matching. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/e64d114a4f858be58edfdd5cb94c0f0b-Abstract-Conference.html.

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.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017. doi: 10.1109/ICCV.2017.324.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, Hado van Hasselt, Razvan Pascanu, James Martens, and Will Dabney. Disentangling the causes of plasticity loss in neural networks. In *Proceedings of the 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 750–783, 2025. URL https://proceedings.mlr.press/v274/lyle25a.html.

Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. arXiv:1609.01596.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020. doi: 10.1073/pnas.2015509117.

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*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8925–8935. PMLR, 2021.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *Proceedings of the 15th European Signal Processing Conference (EUSIPCO)*, pp. 606–610, Poznan, Poland, 2007.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986. doi: 10.1038/323533a0.

Houman Safaai, Varun Reddy, and Bernardo L. Sabatini. Conditioned Direct Feedback Alignment via activity and error geometry. *arXiv preprint arXiv:2607.18574*, 2026.

Albert Jiménez Sanfiz and Mohamed Akrout. Benchmarking the accuracy and robustness of feedback alignment algorithms. *arXiv preprint arXiv:2108.13446*, 2021.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014. arXiv:1312.6120.

Nicol N. Schraudolph. Centering neural network gradient factors. In Genevieve B. Orr and Klaus-Robert Müller (eds.), *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, pp. 207–226. Springer, 1998. doi: 10.1007/3-540-49430-8_11.

Terrence J. Sejnowski. Storing covariance with nonlinearly interacting neurons. *Journal of Mathematical Biology*, 4(4):303–321, 1977. doi: 10.1007/BF00275079.

Terrence J. Sejnowski and Gerald Tesauro. The Hebb rule for synaptic plasticity: Algorithms and implementations. In John H. Byrne and William O. Berry (eds.), *Neural Models of Plasticity*, pp. 94–103. Academic Press, 1989. doi: 10.1016/B978-0-12-148955-7.50010-2.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32145–32168. PMLR, 2023.

Javier E. Vitela and Jaques Reifman. Premature saturation in backpropagation networks: Mechanism and necessary conditions. *Neural Networks*, 10(4):721–735, 1997. doi: 10.1016/S0893-6080(96)00117-7.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yutaro Yamada, Luca Grillotti, Rujikorn Charakorn, Sebastian Risi, David Ha, and Robert Tjarko Lange. Diffusing blame: Task-dependent credit assignment in biologically plausible dual-stream networks. In *ALIFE 2026: Proceedings of the 2026 Artificial Life Conference*, 2026. doi: 10.1162/ISAL.a.1004.

SUPPLEMENTARY MATERIAL

The supplement follows the main paper's argument. Appendix A defines the experimental protocol and measurements; Appendix B gives the mathematical model. Appendix C tests the mechanism, approximations and predictions. Appendix D examines decodability and gate recovery, and Appendix E establishes the scope and costs of the interventions. Numerical comparisons appear beside the relevant analyses. Complete condition summaries, including conditions not tabulated here, are supplied with the code.

A EXPERIMENTAL PROTOCOL AND REPRODUCIBILITY

This section specifies the data and network choices, defines the diagnostics used to separate activity changes from useful learning, and gives the intervention implementations and computational costs.

A.1 DATA, NETWORKS AND RANDOMNESS

We use MNIST (LeCun et al., 1998a), Fashion-MNIST (Xiao et al., 2017), CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) through `torchvision`. After any class restriction, the last $10{,}000$ examples of the official training set form the validation split; the remainder is used for training. The official test sets are not used. The diagnostic probe contains $2{,}048$ validation examples sampled with replacement according to the training sampler. Its indices remain fixed within a run, including after a label-prior shift; the probe then retains the pre-shift sampling distribution. Consequently, reported accuracy is a validation diagnostic, not a test-set benchmark.

Default preprocessing divides pixels by 255. Global standardization subtracts one scalar training mean and divides by one scalar standard deviation. Per-pixel centering subtracts the training mean image; per-pixel standardization also divides by the pixelwise standard deviation plus 10^{-3}. Only per-pixel centering makes the training mean vector zero, up to numerical precision; validation and minibatch means can remain nonzero. In imbalanced experiments, mixture weight p_0 selects a class-0 example, and the remaining weight selects from the original data distribution. Thus $p_0 = 0.7$ is a mixture parameter, not the final class-0 frequency. The headline softmax conditions use global standardization; the batch-size softmax sweep retains raw pixels. Table 28 compares both with matched horizons and seeds. Restricting classes before the fixed validation split also changes the training-set size. The class-count sweep therefore checks the initialization formula without isolating class count as a cause of later learning speed. Imbalanced results are accompanied by the majority-class reference and balanced accuracy in Table 26.

The base MLP has three hidden layers of width 300; depth and width sweeps change these values as specified in the configuration files. The small CNN has two 3×3 convolutions with 16 and 32 channels, padding one and strides one and two, followed by a 128-unit hidden fully connected layer and the output layer. Each hidden block uses the selected activation. DFA broadcasts to every flattened block activation; the local convolutional gradient then aggregates spatial contributions through weight sharing. The fully connected derivations describe the MLP, while CNN results test the diagnostic signature empirically.

For squared error, targets are scaled one-hot class vectors, optionally centered across classes. The multi-label readout retains ten class indicators and adds two binary targets, each active for three randomly selected classes. Accuracy always uses the original class outputs. Initialization and minibatch order share an initialization seed; feedback uses an independent seed. Main DFA conditions use five initializations and three feedback draws each (15 trajectories); BP uses the five corresponding initializations. Sweeps generally use three initialization–feedback pairs, with the exact count reported in each table.

Readout functions and losses. For sigmoid outputs, $\hat{y}_c = \sigma(z_c)$ with $\sigma(z) = (1 + e^{-z})^{-1}$; their probabilities are independent and need not sum to one. Softmax instead gives $\hat{y}_c = \exp(z_c)/\sum_j \exp(z_j)$. Their per-example losses are $-\sum_c [y_c \log \hat{y}_c + (1 - y_c)\log(1 - \hat{y}_c)]$ and $-\sum_c y_c \log \hat{y}_c$, respectively. For linear outputs, $\hat{y} = z$ and the loss is $\|z - y\|^2/2$. Each gives the error $e = \hat{y} - y$ used in Equation (1); training averages the loss over a minibatch.

A.2 DIAGNOSTICS AND NOTATION

Unless a configuration overrides the interval, we log the probe loss, accuracy, mean error and layer statistics every ten updates, and the more expensive diagnostics every fifty. Per-unit means, spreads and gate energies are saved at fixed snapshots. Tables report means $\pm$ one standard deviation across runs unless stated otherwise. Norms of input-dependent error mean the root mean square, $(\mathbb{E}_x\|\tilde{e}(x)\|^2)^{1/2}$.

Averages and symbols. Bars and $\langle\cdot\rangle$ in an update refer to its training minibatch. Diagnostic expectations $\mathbb{E}_x$ instead use the fixed probe; expectations over random feedback or sampled batches are stated separately. Layer indices run from 1 to L, unit indices from 1 to d_ℓ, and subscript 0 denotes initialization when attached to an error or time-dependent statistic. The notation below distinguishes measured unit moments from the approximation used in the reduced model.

Table 1: Notation used in the analysis. A bar denotes an average over inputs, except for $\bar{u}_\ell$ and $\overline{u^2}$, which average gate energies across units.

Symbol	Meaning
$C,\ L,\ d_\ell$	Number of outputs, hidden layers and units in layer ℓ; i indexes units.
$x,\ h_\ell,\ a_\ell$	Input, layer activity vector and preactivation vector; $h_0 = x$.
$z,\ \pi,\ \mathcal{L}_{\text{prior}}$	Readout logits, class-frequency vector and loss of the constant prior predictor.
$\hat{y},\ y,\ e$	Prediction, target and output error $e = \hat{y} - y$.
$\bar{e},\ \tilde{e},\ \hat{e}_0$	Mean error, deviation $e - \bar{e}$ and unit vector along the initial mean error.
$W_\ell,\ b_\ell,\ B_\ell$	Forward weights, biases and fixed feedback matrix.
$\gamma_\ell,\ \delta_\ell$	Activation derivative and gated teaching signal $\gamma_\ell \odot B_\ell e$.
$\bar{\Gamma}_\ell,\ r_\ell$	Diagonal matrix of mean gates and the gate–error covariance residual in Equation (3).
$\mu_{\ell i},\ \sigma_{\ell i}$	Mean and standard deviation of one unit's preactivation across inputs.
σ_ℓ^2	Shared within-unit variance assumed by the reduced model; distinct from the measured $\sigma_{\ell i}^2$.
$\sigma(z),\ \sigma'_{\text{eff}}$	Sigmoid output function and its effective slope in the fixed-activity dose estimate; distinct from preactivation spread.
H_ℓ	Squared norm of the activity mean, $\|\bar{h}_\ell\|^2$; $H_0 = \|\bar{x}\|^2$.
$u_{\ell i},\ \bar{u}_\ell,\ p_\ell$	Unit gate energy, its mean across units and the gate participation ratio.
$\hat{\kappa}_\ell,\ \kappa_\ell(t)$	Initialization estimate of drive and measured accumulated drive along the initial error direction.
$\Lambda_\ell,\ R_\ell$	Activity–label covariance and its regularized, scale-normalized decoding statistic.
$G_\ell^{\text{DFA}},\ G_\ell^{\text{BP}}$	DFA gradient estimate ($\Delta W_\ell = -\eta G_\ell^{\text{DFA}}$) and true weight gradient on the same probe.
$\Pi_\ell,\ \cos\alpha_\ell$	Normalized signed gradient projection and gradient-direction cosine.
$g,\ \eta_{\text{hid}},\ \eta_{\text{out}}$	Feedback standard deviation and hidden/output learning rates.
$a,\ T_{\text{plateau}}$	Effective hidden rate $a = \eta_{\text{hid}} g$ and time from detected plateau onset to exit; scalar a is distinct from a_ℓ.

Hidden cosine. Mean pairwise cosine averages $h(x)^\top h(x')/(\|h(x)\|\|h(x')\|)$ over distinct pairs among 512 probe examples. A value near one indicates similar directions; by itself it does not establish equal vector magnitudes or an input-independent representation.

Teaching-signal strength and useful descent. Let $G_\ell^{\text{DFA}} = \langle \delta_\ell h_{\ell-1}^\top \rangle$ and $G_\ell^{\text{BP}} = \nabla_{W_\ell}\mathcal{L}$ on the same probe sub-batch, where $\mathcal{L}$ denotes its average loss. The signed projected descent, the projected BP-step ratio of Safaai et al. (2026), is

$$\Pi_\ell = \frac{\langle G_\ell^{\text{DFA}}, G_\ell^{\text{BP}} \rangle_F}{\|G_\ell^{\text{BP}}\|_F^2} = \frac{\|G_\ell^{\text{DFA}}\|_F}{\|G_\ell^{\text{BP}}\|_F} \cos\alpha_\ell, \tag{9}$$

where $\langle U, V \rangle_F = \text{tr}(U^\top V)$ is the Frobenius inner product, tr sums diagonal entries, and $\cos\alpha_\ell$ is the cosine between the two vectorized matrices. A positive value predicts local loss decrease from that hidden weight update; a negative value predicts increase, to first order in the step size. The statistic

Table 2: The diagnostics answer different questions. A single value does not establish the properties in the final column.

Diagnostic	Measures	Does not by itself establish
Hidden cosine	Similarity of activity directions	Equal magnitudes or loss of decodability
Gate participation p_ℓ	Distribution of squared derivatives across units	Total gate strength or descent
Mean gate energy $\bar{u}_\ell$	Average squared derivative	Feedback magnitude or alignment
Error norm $\|\bar{e}\|$	Shared output teaching signal	Delivered parameter displacement
Gradient cosine $\cos\alpha_\ell$	Direction of the gradient estimate relative to BP	Size of loss improvement
Projected descent Π_ℓ	Alignment weighted by relative magnitude	Optimizer-preconditioned descent
Weight alignment	Feedback versus downstream weight product	Gradient alignment through gates
Label covariance $\|\Lambda_\ell\|_F^2$	Amplitude of activity–label association	Scale-normalized decodability

includes relative update magnitude and need not lie in $[-1, 1]$. These diagnostics use the raw DFA rule; intervention-specific gradients are not reconstructed by this diagnostic. Weight alignment in the teacher–student comparison is the cosine between vectorized B_ℓ and $M_\ell = (W_{\text{out}} W_L \cdots W_{\ell+1})^\top$, without derivative gates (Appendix E.8).

Label covariance and linear decoding. For classification diagnostics, y denotes the original one-hot class target, including in the multi-label and scaled-target controls. Let $\Sigma_{h,\ell} = \mathrm{Cov}(h_\ell, h_\ell)$ and $\Lambda_\ell = \mathrm{Cov}(h_\ell, y)$. The energy $\|\Lambda_\ell\|_F^2$ measures the amplitude of activity–label association. Its reported fold decrease is the ratio of the initial value to the minimum after step 30, computed within each run and then averaged. To remove much of the dependence on activity scale, we use the regularized statistic

$$R_\ell = \frac{\mathrm{tr}[\Lambda_\ell^\top (\Sigma_{h,\ell} + \epsilon_\ell I)^{-1} \Lambda_\ell]}{C^{-1} \sum_c \mathrm{Var}(y_c)}, \qquad \epsilon_\ell = 10^{-4}\,\mathrm{tr}(\Sigma_{h,\ell})/d_\ell. \tag{10}$$

Here I is the identity, ϵ_ℓ is a ridge regularizer, and C counts the original classes, including when the readout has extra binary outputs. The statistic uses at most 1,024 probe examples, with target-variance normalization from the full probe. It quantifies linearly accessible covariance after correcting for feature scale; it is not held-out decoder accuracy or an information-theoretic quantity. Because the denominator is the average rather than total target variance, the ideal population statistic can approach C, not one. The matched-checkpoint comparison in Table 19 evaluates both statistics when covariance energy is smallest among checkpoints with a recorded R_ℓ; independently sampled minima are not compared.

Centered dimensionality. We measure the entropy effective rank of the centered covariance $\Sigma_{h,\ell}$ (Roy & Vetterli, 2007). If its eigenvalues are λ_j and $w_j = \lambda_j / \sum_k \lambda_k$, the effective rank is $\exp(-\sum_j w_j \log w_j)$, with $0 \log 0 = 0$. It uses 1,024 probe examples and describes how variance is distributed across activity directions, independently of its total amplitude. Gate participation instead describes how squared activation derivatives are distributed across units.

Gate sets. For ReLU, an inactive unit has positive preactivation on less than 5% of probe inputs; an always-active unit has positive preactivation on more than 95%. The latter need not have constant output. For each snapshot at step t, the gate-energy set S_t contains the top fifth of units ranked by $u_{\ell i}$. Turnover is measured by Jaccard overlap $|S_t \cap S_{120}|/|S_t \cup S_{120}|$. For independent sets containing a fraction f of units, the large-width chance reference is $f/(2-f)$. The Gaussian reconstruction uses each saved $(\mu_{\ell i}, \sigma_{\ell i})$ to compute $u_{\ell i} = \mathbb{E}_{Z \sim \mathcal{N}(\mu_{\ell i}, \sigma_{\ell i}^2)} \operatorname{sech}^4 Z$, then substitutes these energies into Equation (2).

Virtual-update diagnostics. At fixed training snapshots, we copy the network and apply a full-probe SGD step. Before/after means and every term in the drift budget use the same validation probe;

the original network and training optimizer are unchanged. This measures a full-probe local update, not the actual stochastic training step.

For gate-set comparisons, we instead change one hidden layer at a time. We mask rows of its local weight gradient and the corresponding bias entries using all units, the current top fifth by gate energy, the top fifth fixed at step 120, a fixed random fifth, or the fixed set's complement. Each virtual displacement is normalized to joint weight-plus-bias norm 10^{-3}. We compare its direction with the true probe gradient and record its finite loss change. Current, fixed and random fifths match unit count; the complement is larger. All comparisons match displacement norm.

A.3 Plateau detection and reporting

The five-point-smoothed probe loss must remain within 3% of the constant-predictor loss for at least 50 steps, with probe accuracy no greater than $\max_c \pi_c + 0.05$. Exit is the first later logged step below $0.9\mathcal{L}_{\mathrm{prior}}$. Plateau duration is exit minus onset. Separately, ***learning time*** is the first crossing below $0.9\mathcal{L}_{\mathrm{prior}}$ that remains below it for 50 updates, whether or not a plateau was detected; this permits the same comparison for all optimizers and controls. Participation minima use the first 300 steps because late saturation can also lower participation. Peak cosine uses the full recorded trajectory unless a figure specifies a shorter window. Curves show means across trajectories; shaded bands, where present, show one standard deviation, not confidence intervals. Trajectory bands use the population standard deviation across runs; tables and error bars use the sample standard deviation. Model timing uses the analogous loss criterion without an accuracy condition. Unless specified otherwise, r denotes Pearson correlation; the unit-drift comparison uses Spearman rank correlation. Model correlations pool condition–seed–layer observations, which are not independent replicates.

A.4 Intervention implementations and cost

Error centering subtracts the minibatch mean before feedback projection. Teaching-signal centering subtracts the mean after applying derivative gates. In both cases, the output layer still receives its ordinary gradient. Prior-bias initialization sets sigmoid biases to nominal prior logits (Lin et al., 2017); the balanced MNIST controls use $\mathrm{logit}(1/C)$. The distinct mean-logit calibration control uses empirical class frequencies and a training subsample to shift initial mean logits toward the corresponding prior logits. Because the sigmoid is nonlinear, this latter operation only approximately calibrates the mean prediction.

Batch normalization is applied to each hidden linear preactivation before the activation function, with trainable scale and offset. The local DFA gradient differentiates through the activation and minibatch normalization; inputs from earlier blocks are detached. Probe evaluation and snapshots also use their own batch statistics, rather than stored running statistics. Reported preactivation moments and derivative gates are measured after normalization and its affine transform; the gate statistic does not include the normalization Jacobian. The trainable offset means that normalized preactivations need not retain zero mean.

The local-error conditioner forms an exponential moving average S_ℓ of $\langle \delta_\ell \delta_\ell^\top \rangle$, with decay 0.99. It replaces each teaching signal by $(S_\ell + \lambda_\ell I)^{-1}\delta_\ell$ and rescales the transformed batch to the original Frobenius norm. Here I is the identity and $\lambda_\ell = \lambda_{\mathrm{rel}}\,\mathrm{tr}(S_\ell)/d_\ell$, where λ_{rel} is the dimensionless ridge strength varied in Figure 10d. This is inverse-second-moment preconditioning, not inverse-square-root whitening. It acts in the d_ℓ-dimensional hidden teaching space, as in the error-side operator of Safaai et al. (2026). Our control adapts that operator: the cited implementation uses current-minibatch moments and absolute additive ridges, whereas ours uses an exponential average, trace-relative damping and restoration of the teaching-signal norm. The comparison should not be read as an identical implementation of the published method.

The orthogonalization control applies five Newton–Schulz iterations to each hidden weight-gradient matrix and restores its original Frobenius norm. It is Muon-style gradient orthogonalization (Jordan et al., 2024), not the full momentum-based optimizer. Table 3 gives both asymptotic overhead and measured runtimes for the implemented controls.

The timing ratios use three paired runs and include probes and snapshots as well as training. Their configurations record CUDA, but not the specific GPU model. Small differences near unity should be interpreted as comparable measured cost, rather than established speed improvements.

Table 3: Additional per-step cost for minibatch size n, C outputs and square hidden maps of width d, and descriptive mean wall clock per run relative to plain DFA on the base network. Rectangular orthogonalization costs depend on both matrix dimensions. Costs exclude the ordinary DFA update.

Intervention	Extra operations	Persistent state	Wall clock
Center output error	$O(nC)$	none	0.95
Center teaching signal	$O(nd)$ per layer	none	1.08
Prior bias / faster readout	none during training	none	0.92 / 0.98
Batch normalization	$O(nd)$ per layer	$O(d)$ per layer	1.10
Orthogonalize updates	$O(d^3)$ per layer	none beyond SGD	1.21
Local-error conditioner	$O(nd^2 + d^3)$ per layer	$O(d^2)$ per layer	1.74

A.5 Code and computation

The supplementary archive contains the training configurations, analysis and figure scripts, tests, and reference outputs for the additional controls. Complete condition summaries and a mapping from each printed comparison to its source are included in `results/paper_summary_sources/`; they retain sweeps omitted from the printed supplement. The README gives regeneration commands and identifies which full training logs must be regenerated. Numerical macros and tables are generated from these records.

The frozen-feature data include the fitting/evaluation protocol, and the fresh-architecture data include predictions saved before training. Virtual-update measurements record configurations and probe/feedback hashes. The execution manifest records an A100 MIG slice for the additional GPU controls and CPU integrations for the reduced model; per-run configurations also record device and wall time. The additional learning-rate and imbalanced-calibration runs (E21 and E22) used CPUs. The intervention costs in Table 3 are implementation- and hardware-dependent.

B Derivations and reduced-model assumptions

We derive the exact update decomposition, then state the approximations that yield mean drift, accumulated drive and gate participation. The final subsection extends the model to approximate plateau exit. Empirical checks of the omitted terms and predictions are collected in Appendix C.

B.1 Exact mean–covariance decomposition

For any batch, $\langle \delta h^\top \rangle = \langle (\delta - \bar{\delta})(h - \bar{h})^\top \rangle + \bar{\delta}\bar{h}^\top$. Substituting $\delta = \gamma \odot B(\bar{e} + \tilde{e})$ and $\langle \tilde{e} \rangle = 0$ gives Equation (3); the bias update follows from Equation (1). At initialization, zero-mean Gaussian feedback independent of the forward computation gives the residual zero expectation over feedback draws. This does not bound its norm in one network or during training.

The decomposition by output-error mean is different from the decomposition by teaching-signal mean. Suppressing the layer index, put $v = B\bar{e}$, and let $\mathrm{diag}(v)$ denote the diagonal matrix with entries v_i. At a fixed network state, the weight update contributed by shared output error is exactly

$$\Delta W_{\bar{e}} = -\eta \,\mathrm{diag}(v) \langle \gamma h^\top \rangle = -\eta(\bar{\gamma} \odot v)\bar{h}^\top - \eta \,\mathrm{diag}(v)\,\mathrm{Cov}(\gamma, h). \tag{11}$$

The first term has rank at most one; the second need not. Pre-gate error centering removes both, while centering the post-gate teaching signal removes its mean outer product and bias update. These operations therefore test different interventions. Appendix C.3 measures the two shared-error components alongside the readout budget.

B.2 Gated drift

To turn the mean update into a prediction of saturation, consider one layer with trainable biases and suppress its index. Write $\mu_i = w_i^\top \bar{h} + b_i$ and $v = B\bar{e}$, where $w_i^\top$ is row i of W. Holding presynaptic activity fixed, the exact local update is

$$\Delta \mu_i = -\eta(\bar{\gamma}_i v_i + r_i)(\|\bar{h}\|^2 + 1) - \eta \,\mathrm{Cov}(\delta_i, h)\bar{h}. \tag{12}$$

The full continuous-time trajectory additionally contains $w_i^\top \dot{\bar{h}}$, where a dot denotes a derivative with respect to update count. Dropping this upstream contribution, the covariance contribution and r_i, and setting $\bar{\gamma}_i \approx \operatorname{sech}^2 \mu_i$, gives Equation (4). Evolving the energies in the reduced model does not restore the omitted terms.

For a finite simultaneous network update, the complete identity includes $W\Delta\bar{h} + \Delta W \Delta\bar{h}$ in addition to the local terms above. We measure all five vector contributions with virtual full-probe SGD updates in Appendix C.3. This tests the local closure separately from the empirical accuracy of the reduced model.

For a fixed error direction, define $s(t) = \eta \int_0^t \langle \bar{e}(t'), \hat{e}_0 \rangle (H_{\ell-1}(t') + 1) dt'$ and $\hat{v} = B_\ell \hat{e}_0$. Separating variables gives

$$F(\mu_i(t)) = F(\mu_i(0)) - \hat{v}_i s(t), \qquad F(\mu) = \frac{\mu}{2} + \frac{\sinh(2\mu)}{4}. \tag{13}$$

For large $|\hat{v}_i| s$, $|\mu_i| \approx \frac{1}{2} \ln(8|\hat{v}_i| s)$. Since $F'(\mu) = \cosh^2 \mu$, it integrates the inverse derivative gate. The accumulated drive can therefore grow much faster than the saturated mean. If hidden biases are frozen, the physical drift factor is $H_{\ell-1}$ rather than $H_{\ell-1} + 1$.

B.3 Readout dynamics and the dose

With $\sigma(z) = (1 + e^{-z})^{-1}$, each mean logit in the symmetric mean-only sigmoid model equals z, $\bar{e} = (\sigma(z) - 1/C)\mathbf{1}$, and $\dot{z} = -\eta_{\text{out}}(H_L + 1)(\sigma(z) - 1/C)$. Starting at $z = 0$ with $C > 2$, the fixed point is $z^* = \operatorname{logit}(1/C)$, where $\operatorname{logit}(q) = \ln[q/(1-q)]$. For constant H_L,

$$\int_0^\infty \|\bar{e}\| dt = \frac{\sqrt{C}}{\eta_{\text{out}}(H_L + 1)} \int_{z^*}^0 dz = \frac{\sqrt{C} \ln(C-1)}{\eta_{\text{out}}(H_L + 1)}. \tag{14}$$

For $C = 2$ there is no initial drive; the effective-slope formula has limiting value $1/4$, while the dose is zero. Near the fixed point, the tail decay rate is $\eta_{\text{out}}(H_L + 1)\pi(1 - \pi)$ for $\pi = 1/C$. This tail rate differs from the effective slope that summarizes the whole integral.

At an exactly constant initial prediction q, the sigmoid mean-error norm is $\sqrt{C}|q - 1/C|$ for balanced one-hot targets. Uniform softmax instead gives $\|C^{-1}\mathbf{1} - \pi\|$. Random-readout predictions fluctuate across inputs, so these ideal formulas need not equal the measured initial norm. For general readouts the reduced model integrates the readout equation numerically and uses the actual initial mean logits and target mean.

For arbitrary constant $q \in (0, 1)$, the same fixed-activity sigmoid calculation gives total dose proportional to $\sqrt{C}|\operatorname{logit}(q) - \operatorname{logit}(1/C)|$, rather than to $\sqrt{C}|q - 1/C|$ alone. The product scaling observed when varying q is therefore empirical, not an exact consequence of keeping σ'_{eff} fixed.

The online diagnostic uses Equation (7), projecting onto the mean-error direction of the first training minibatch. Its code normalizes with a small numerical stabilizer; an exactly zero initial vector therefore gives zero projected dose. This does not rule out later stochastic or residual-driven activity. The logged diagnostic retains the $+1$ convention even in frozen-bias controls; the reduced model removes the bias channel for those controls. For transient comparisons, the sum is stopped at the first logged point where the validation-probe mean-error norm is below 15% of its initial probe value.

Finite-batch error fluctuations remain even at zero population mean; Appendix E.7 derives and tests their sampling law separately from collapse.

B.4 Participation quadrature

We now relate the mean shifts to concentration of gate energy across units. With negligible within-unit spread, $u_i = \operatorname{sech}^4 \mu_i$. For initially zero means and independent coefficients $\hat{v}_i \sim \mathcal{N}(0, g^2)$, Equation (13) reduces the large-width gate moments to one-dimensional integrals in $\kappa = gs$. Here $\bar{u}$ and $\overline{u^2}$ average over units, and φ is the standard normal density. As $\kappa \to \infty$, substitution

$w = |\hat{v}_i| s = F(|\mu|)$ gives

$$\bar{u} \sim \frac{2\varphi(0)}{\kappa} \int_0^\infty \operatorname{sech}^2 \mu \, d\mu = \frac{\sqrt{2/\pi}}{\kappa}, \tag{15}$$

$$\overline{u^2} \sim \frac{2\varphi(0)}{\kappa} \int_0^\infty \operatorname{sech}^6 \mu \, d\mu = \frac{8}{15} \frac{\sqrt{2/\pi}}{\kappa}. \tag{16}$$

Their ratio gives $p = \bar{u}^2/\overline{u^2} \sim (15/8)\sqrt{2/\pi}/\kappa$. This is an asymptote of the stated closure, not an exact finite-width identity for trained networks. The large-width limit precedes the large-dose limit. Since finite-width participation is at least $1/d_\ell$, the asymptote cannot be extrapolated to arbitrarily large dose at fixed width.

B.5 Complete common-mode reduced model

The implemented model uses a common preactivation variance σ_ℓ^2 within each layer and a separate mean $\mu_{\ell i}$ for each unit. The model replaces the narrow-spread gate in Equation (4) by a Gaussian average. For $Z \sim \mathcal{N}(0,1)$, it computes mean activity $m_{\ell i}$, mean gate $\bar{\gamma}_{\ell i}$ and gate energy $u_{\ell i}$:

$$m_{\ell i} = \mathbb{E}_Z \tanh(\mu_{\ell i} + \sigma_\ell Z), \quad \bar{\gamma}_{\ell i} = \mathbb{E}_Z \operatorname{sech}^2(\mu_{\ell i} + \sigma_\ell Z), \quad u_{\ell i} = \mathbb{E}_Z \operatorname{sech}^4(\mu_{\ell i} + \sigma_\ell Z). \tag{17}$$

We set $H_\ell = \sum_i m_{\ell i}^2$ and use these gate energies in Equation (2). The first-layer variance is initialized from the network and held fixed. Subsequent variances follow $\sigma_\ell^2 = q_\ell \bar{u}_{\ell-1} \sigma_{\ell-1}^2$, with $q_\ell = d_{\ell-1} \operatorname{Var}(W_{\ell,ij})$ fixed at initialization. This is a random-weight propagation approximation; it drops feature correlations and later changes in forward-weight statistics.

Let s_{out} be the sigmoid, softmax or identity output map, and $\bar{y}$ the training target mean. With indicator $\iota_b = 1$ for trainable hidden biases and $\iota_b = 0$ for frozen biases, the integrated mean equations are

$$\dot{\mu}_{\ell i} = -\eta_{\text{hid}} \bar{\gamma}_{\ell i} (B_\ell \bar{e})_i (H_{\ell-1} + \iota_b), \qquad \dot{\bar{z}} = -\eta_{\text{out}} (H_L + 1) \bar{e}, \qquad \bar{e} = s_{\text{out}}(\bar{z}) - \bar{y}. \tag{18}$$

Applying the output map to the mean logit is another closure: generally $\mathbb{E} s_{\text{out}}(z) \neq s_{\text{out}}(\mathbb{E} z)$. All initial means, first-layer spread, feedback matrices and weight variances are read from the initialized network. No trajectory parameters are fitted. The implementation uses 64-node Gauss–Hermite quadrature and forward Euler integration, with one substep per training update by default. Appendix C.4 checks sensitivity to both numerical resolutions.

The model's predicted hidden cosine is $H_\ell/(H_\ell + d_\ell \bar{u}_\ell \sigma_\ell^2)$. It approximates the mean pairwise cosine by a ratio of expected inner products and squared norms, requiring concentration of activity norms. It is not an exact cosine identity. In particular, high cosine alone cannot distinguish small centered variation from a large common offset.

Appendix C.4 evaluates these predictions across settings and reports detection failures.

B.6 Approximate covariance dynamics and exit prediction

To estimate when loss starts falling below the prior, we augment the collapse model with scalar covariance amplitudes $A_\ell \approx \|\Lambda_\ell\|_F$. Define $A_0 = \|\operatorname{Cov}(x,y)\|_F$, $\rho_\ell = d_\ell \operatorname{Var}(W_{\ell,ij})$, and an accumulated local contribution J_ℓ initialized to zero. The implementation uses

$$A_\ell = \sqrt{\rho_\ell \bar{u}_\ell}\, A_{\ell-1} + J_\ell, \qquad \dot{J}_\ell = \eta_{\text{hid}} g \sqrt{d_\ell / C}\, \frac{\bar{u}_\ell}{\sqrt{p_\ell}} A_{\ell-1}^2. \tag{19}$$

In Equation (19), the first term transmits upstream covariance instantaneously; the second accumulates a local learning contribution. Adding their norms is a phenomenological closure, not an exact matrix-covariance identity. The accumulated contribution is not attenuated again when gates later change.

An accumulated loss-reduction variable E starts at zero and evolves as $\dot{E} = \eta_{\text{out}} A_L^2$. The model loss is the mean-only readout loss minus E, with a constant offset chosen so that the mean-only fixed-point loss equals the data's prior-predictor loss. Near a constant predictor, $\operatorname{Cov}(e, h_L) \approx -\Lambda_L^\top$, motivating this descent estimate. Away from that regime, prediction-dependent covariance and higher-order terms are omitted. The integration is used to estimate the first exit below $0.9\mathcal{L}_{\text{prior}}$, not to model the eventual training loss. Because the variance cascade lacks recovery dynamics, the model can predict an exit while its gates remain concentrated. It therefore cannot account for the observed participation rebound.

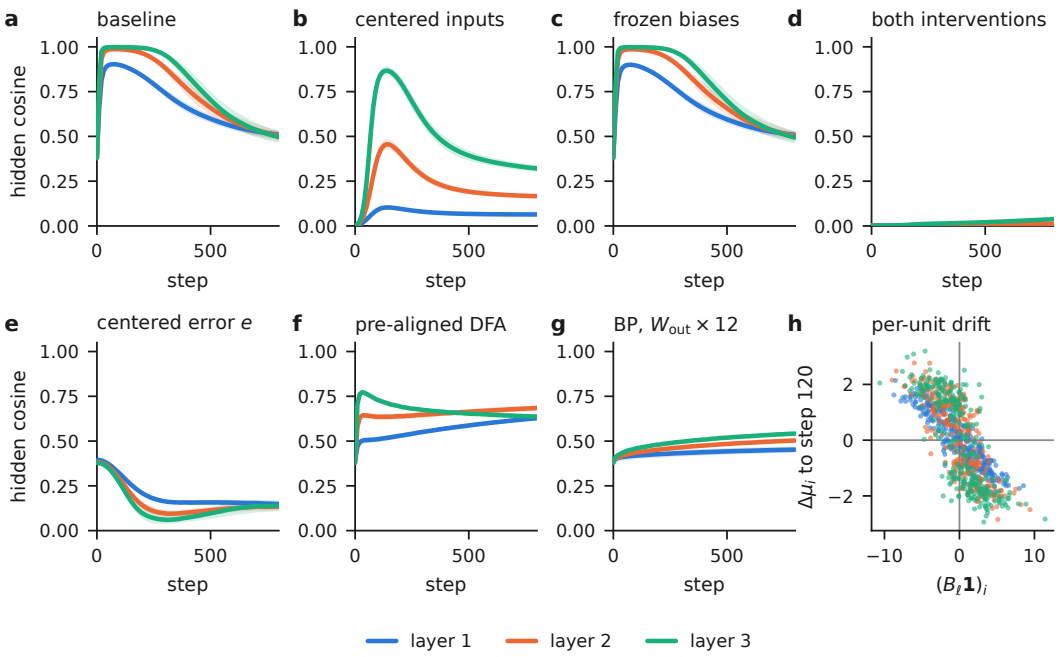

Figure 5: **Causal dissection.** (a–d) Input centering × frozen hidden biases: the combined intervention prevents collapse at every layer without forcing all hidden means to zero. (e) Output error centered before projection. (f) DFA with feedback pre-aligned to the transposed downstream map at the Frobenius norm of the random draw. (g) BP with its forward readout weights scaled by 12, which also raises $\|\bar{e}_0\|$ to 1.50: neither alignment control collapses. (h) Unit mean changes over the first 120 steps versus $(B_\ell \mathbf{1})_i$, for one baseline trajectory (seeds 0,0). Colours denote layers. Curves in a–g average runs; bands show one standard deviation. Counts are in Table 4.

C TESTS OF THE MECHANISM AND REDUCED MODEL

We first test the factors that produce the shared drive, then measure the terms omitted by the model. Prediction accuracy is assessed separately from these local checks, with all model detection failures retained.

C.1 MEAN-DRIVE AND ALIGNMENT CONTROLS

The centering controls test the two routes through which a mean teaching signal shifts a unit's preactivation. Removing the input mean protects the first layer, but deeper collapse persists; freezing hidden biases alone is also insufficient. Together they prevent collapse throughout the network (Figure 5a–d). Global standardization instead raises the MNIST input mean energy and deepens layer-1 gate concentration (minimum p_1 of 0.59 versus 0.71), although peak cosine is lower (0.80 versus 0.90). Centering output error acts on the other factor and suppresses baseline collapse (panel e). Table 4 summarizes these controls; values are means and standard deviations across trajectories, and unavailable endpoints remain marked.

Feedback orientation provides a separate test. Pre-aligned DFA learns quickly with little collapse (panel f), although matching the feedback matrix's Frobenius norm does not preserve $B_\ell \bar{e}$. BP remains uncollapsed when its output weights are amplified (panel g), so its smaller initial feedback scale is insufficient to explain the difference. Ten updates into training, the ratio of shared to input-dependent error norms is 1.28 under random feedback, 0.23 with pre-aligned feedback, 0.07 for BP with a twelvefold readout, and 1.06 for unscaled BP. These measurements are consistent with the local restoring contribution in Equation (5), although neither alignment control isolates it. Unit mean shifts are anticorrelated with $(B_\ell \mathbf{1})_i$, as the negative sign in Equation (4) predicts (panel h). Table 6 checks this relation across all baseline trajectories. These comparisons support mean-driven

Table 4: MNIST baseline and mean-drive controls. Peak cosine uses the recorded trajectory; participation uses the first 300 updates. Accuracy is measured at step 3000. The frozen-readout control ends at step 1500, so its endpoint is unavailable (–). Softmax rows use globally standardized inputs, while other rows use raw pixels unless input centering is specified. Values are means ± standard deviations.

Condition	n	Peak $\cos_3$	Min. p_3	Accuracy
Plain DFA	15	0.998 ± 0.000	0.225 ± 0.029	0.894 ± 0.006
Backpropagation	5	0.731 ± 0.012	0.951 ± 0.004	0.861 ± 0.006
Layerwise FA	15	0.991 ± 0.002	0.342 ± 0.039	0.882 ± 0.008
Pre-aligned DFA	15	0.772 ± 0.015	0.882 ± 0.007	0.912 ± 0.004
BP, output weights $\times 12$	5	0.607 ± 0.015	0.986 ± 0.002	0.927 ± 0.004
Center output error	15	0.379 ± 0.017	0.932 ± 0.011	0.877 ± 0.007
Center teaching signal	15	0.521 ± 0.062	0.743 ± 0.132	0.861 ± 0.007
Center inputs	15	0.867 ± 0.024	0.704 ± 0.030	0.897 ± 0.006
Freeze hidden biases	15	0.998 ± 0.001	0.229 ± 0.029	0.894 ± 0.006
Center inputs + freeze biases	15	0.081 ± 0.011	0.922 ± 0.015	0.893 ± 0.007
Faster readout	15	0.923 ± 0.010	0.816 ± 0.020	0.911 ± 0.006
Frozen readout	15	1.000 ± 0.000	0.018 ± 0.006	–
Prior-bias initialization	15	0.474 ± 0.145	0.928 ± 0.065	0.897 ± 0.006
Balanced softmax	15	0.271 ± 0.032	0.929 ± 0.009	0.922 ± 0.004
Imbalanced softmax	15	0.938 ± 0.028	0.477 ± 0.085	0.960 ± 0.005
Imbalanced softmax, BP	5	0.386 ± 0.021	0.972 ± 0.002	0.947 ± 0.005

Table 5: Learning cost of collapse in the MNIST baseline cohort. Learning time is the first probe loss below 90% of the constant-predictor loss that stays below it for 50 updates (the criterion of Table 21); mean ± standard deviation. Accuracy columns give the median first step reaching 50% and 80% probe accuracy; the last column is accuracy at step 3000. Peak cosine uses the full trajectory, as in Table 4. Centering prevents collapse and speeds discrimination but lowers mean-activity energy, which slows the readout's fit to the class prior; the sigmoid loss then stays high although accuracy rises.

Condition	n	Peak $\cos_3$	Learning time	50% acc.	80% acc.	Accuracy
Plain DFA	15	0.998 ± 0.000	531 ± 34	690	1220	0.894 ± 0.006
Frozen hidden biases	15	0.998 ± 0.001	528 ± 35	680	1220	0.894 ± 0.006
Centered inputs	15	0.867 ± 0.024	345 ± 35	430	910	0.897 ± 0.006
Centered inputs + frozen biases	15	0.081 ± 0.011	1153 ± 30	370	1050	0.893 ± 0.007
Centered output error	15	0.379 ± 0.017	1898 ± 153	370	1250	0.877 ± 0.007
Prior-bias initialization	15	0.474 ± 0.145	288 ± 35	420	860	0.897 ± 0.006
Faster readout ($10\times$)	15	0.923 ± 0.010	152 ± 11	180	460	0.911 ± 0.006
Pre-aligned DFA	15	0.772 ± 0.015	22 ± 4	20	100	0.912 ± 0.004
Backpropagation	5	0.731 ± 0.012	234 ± 17	210	1060	0.861 ± 0.006

saturation while showing that its consequences depend on the direction and speed of input-dependent learning.

Centered geometry. High mean pairwise cosine could reflect a shared offset rather than suppressed variation. The centered activity variance $\mathbb{E}_x\|h_3 - \bar{h}_3\|^2$ separates the two. Averaging per-run normalized minima within 500 updates gives 0.028 in the baseline, 0.83 for BP and 0.97 with prior-bias initialization (Table 7). Both error centering and input centering with frozen biases preserve the initial variance over this window. These are averages of per-run minima, not lower bounds for every trajectory. The effective rank of centered activity behaves differently (Appendix A.2). It falls from 46 to about five under random-feedback DFA whether or not collapse occurs (4.3 in the baseline, 4.6 with error centering and 5.3 with prior bias), but only to 37 under BP. Compression thus persists without the strong loss of activity amplitude. A low effective rank alone does not identify the common-mode transient.

C.2 Scaling with drive strength and learning rate

The learning-rate sweep tests whether the drive's strength and duration have separable effects. When both SGD rates vary together, peak hidden cosine (0.998–0.999) and participation minima (0.21–0.23)

Table 6: Unit mean shifts from initialization to step 120 against the shared-error feedback coefficient. Values are means and standard deviations of per-run Spearman correlations across the 15 baseline trajectories (five initializations crossed with three feedback draws); they are descriptive spreads, not independent-run confidence intervals.

Layer	Trajectories	Spearman correlation
1	15	-0.92 ± 0.03
2	15	-0.86 ± 0.03
3	15	-0.83 ± 0.04

Table 7: Centered geometry of top-layer activity in the MNIST baseline cohort. The centered variance is $\mathbb{E}_x\|h_3 - \bar{h}_3\|^2$, the amplitude of activity differences across inputs, and its minimum within the first 500 updates is given relative to initialization. The centered effective rank is the exponential of the entropy of the normalized centered covariance spectrum; its minimum uses the same 500-update window. Means $\pm$ standard deviations across runs.

Condition	n	Min. variance / initial	Initial rank	Min. rank
Plain DFA	15	0.028 ± 0.007	45.6 ± 0.7	4.3 ± 0.3
Backpropagation	5	0.826 ± 0.008	45.6 ± 0.7	37.4 ± 0.6
Pre-aligned DFA	15	0.808 ± 0.013	45.6 ± 0.7	12.2 ± 0.2
Prior-bias initialization	15	0.967 ± 0.126	45.6 ± 0.7	5.3 ± 0.2
Centered inputs + frozen biases	15	1.000 ± 0.000	43.7 ± 0.6	4.8 ± 0.8
Centered output error	15	1.000 ± 0.000	45.6 ± 0.7	4.6 ± 0.6

vary within seed noise, while the rescaled exit time ηt_{exit} lies in 0.52–0.55. These measurements support a change in timescale with little change in collapse severity.

Larger plain-SGD rates, with equal hidden and readout rates, test whether the stall's cost disappears when learning is fast (Table 8). No run diverged up to $\eta = 0.1$. Collapse deepens at the largest rates, and the relative saving from prior-bias initialization grows: the time to 80% accuracy falls by 30% at $\eta = 10^{-3}$ and by 78% at $\eta = 0.1$, where plain DFA needs 130 updates, calibrated DFA 28 and BP 20. The absolute delay is smaller than at the baseline rate, but a substantial advantage of prior-bias initialization remains. The initial biases use the nominal balanced prior, $q = 1/C$, rather than estimating each empirical class frequency.

Varying gain over 0.03–10 and initial sigmoid probability gives a common empirical trend against $g\|\bar{e}_0\|$. Learning time falls as gain rises even as collapse deepens: from 1897 updates at $g = 0.03$, with minimum participation 0.98, to 260 at $g = 10$, with 0.021. Across 33 distinct runs spanning 361-fold, the fitted relation is

$$1 - \max_t \cos_L = 0.0021(g\|\bar{e}_0\|)^{-1.75}, \tag{20}$$

with log–log correlation -0.989. Changing the initial prediction also changes the effective output slope, so this fit is not a universal consequence of the initialization estimate. For the 139 uncentered runs with $\kappa_L > 3$, the median product $\kappa_L \min_t p_L$ is 1.17, compared with the large-dose quadrature limit of 1.50.

Table 9 separately checks the initial-error formula across class counts. The binary case has no drive at exactly uniform prediction, but the random readout leaves a small nonzero mean error.

C.3 Measured limits of the mean-drive approximation

The exact decomposition does not guarantee that its residual and upstream terms are small. We measure them at fixed checkpoints using virtual full-probe SGD updates, as defined in Appendix A.2. All before/after quantities use the same validation examples, and the original training trajectory is unchanged.

Gate–error residual. We evaluate the residual after 0, 50, 100, 200 and 300 updates on a fixed 2,048-example validation probe (sampler seed 999), using initialization–feedback pairs $(0, 0)$, $(1, 1)$ and $(2, 2)$. Baseline and ReLU runs use MNIST; the CIFAR-10 control uses a tanh MLP with the base protocol otherwise unchanged. Table 10 reports ratios to the leading mean-teaching term across

Table 8: Plain-SGD learning rate in the MNIST baseline, with equal hidden and readout rates. Columns give diverged runs (non-finite loss, or final loss above the initial loss), the participation minimum within 300 updates, the median first update reaching 50% and 80% probe accuracy, and accuracy after 3,000 updates, over non-diverged runs with seed pairs $(0, 0)$, $(1, 1)$ and $(2, 2)$. The 10^{-3} rows reuse the headline runs with initialization seeds 0–2 (BP feedback seeds are unused). Prior bias sets all sigmoid biases to $\mathrm{logit}(1/C)$. Probes are logged every ten updates at 10^{-3} and every two at larger rates. Each new run completes 3,000 updates.

η	Rule	n	Diverged	Min. p_3	50% acc.	80% acc.	Accuracy
0.001	DFA	3	0	0.2143 ± 0.0191	690	1150	0.902 ± 0.005
0.001	DFA, prior bias	3	0	0.8545 ± 0.1410	420	800	0.904 ± 0.003
0.001	BP	3	0	0.9504 ± 0.0054	210	1040	0.863 ± 0.006
0.003	DFA	3	0	0.2053 ± 0.0203	236	406	0.925 ± 0.005
0.003	DFA, prior bias	3	0	0.8471 ± 0.1342	138	270	0.928 ± 0.004
0.003	BP	3	0	0.9481 ± 0.0058	66	324	0.904 ± 0.004
0.01	DFA	3	0	0.1614 ± 0.0242	80	132	0.954 ± 0.001
0.01	DFA, prior bias	3	0	0.8173 ± 0.1341	42	82	0.954 ± 0.001
0.01	BP	3	0	0.9296 ± 0.0058	22	108	0.928 ± 0.004
0.03	DFA	3	0	0.0623 ± 0.0105	40	66	0.967 ± 0.003
0.03	DFA, prior bias	3	0	0.7554 ± 0.1711	16	36	0.968 ± 0.003
0.03	BP	3	0	0.8981 ± 0.0082	10	40	0.958 ± 0.002
0.1	DFA	3	0	0.0038 ± 0.0004	104	130	0.971 ± 0.003
0.1	DFA, prior bias	3	0	0.5043 ± 0.2070	14	28	0.974 ± 0.003
0.1	BP	3	0	0.8485 ± 0.0097	6	20	0.974 ± 0.003

Table 9: Initial mean error under sigmoid outputs as the number of MNIST classes changes. The ideal value assumes a constant prediction of one half. The measured value uses a random readout. Class restriction also changes training-set size; this comparison tests initialization, not later learning speed.

C	n	Ideal $\|\bar{e}_0\|$	Measured $\|\bar{e}_0\|$
2	3	0.000	0.057 ± 0.013
3	3	0.289	0.266 ± 0.088
5	3	0.671	0.700 ± 0.083
10	3	1.265	1.289 ± 0.055

all three layers. The residual is small during initial tanh collapse but becomes comparable later, as the leading error decreases. Its relative contribution is larger for ReLU. Absolute norms are retained with the measurements, since a large ratio can result from a small denominator.

Hidden mean drift. For the SGD rule in Equation (1), write $\mu_\ell = \mathbb{E}_x a_\ell$ for the vector of unit means. The virtual full-probe updates in Appendix A.2 give the exact finite-step decomposition:

$$\begin{aligned}\Delta\mu_\ell = & -\eta(H_{\ell-1}+1)\bar{\gamma}_\ell \odot B_\ell \bar{e} - \eta(H_{\ell-1}+1) r_\ell - \eta\,\mathrm{Cov}(\delta_\ell, h_{\ell-1})\bar{h}_{\ell-1} \\ & + W_\ell \Delta\bar{h}_{\ell-1} + \Delta W_\ell \Delta\bar{h}_{\ell-1}. \end{aligned} \tag{21}$$

All unincremented quantities are evaluated before the step; $\Delta\bar{h}_{\ell-1}$ is the change in mean presynaptic activity. The first three terms are local, the fourth propagates upstream changes, and the fifth is their finite-step interaction. Their vector sum reconstructs the measured shift to floating-point precision; their norms need not sum.

A large omitted vector need not explain a large fraction of net motion. For each contribution d_k to $\Delta\mu$, we therefore also report the signed fraction $\langle d_k, \Delta\mu\rangle / \|\Delta\mu\|^2$. These fractions can be negative or exceed one when terms cancel; the complete decomposition sums to one. Upstream motion dominates initially, whereas the leading mean-error term dominates during the collapse (Figure 6a). Panel e gives the corresponding absolute norms; Table 11 includes the norm of the net displacement. Upstream motion may itself originate in mean-driven updates, so this comparison does not identify independent causal sources.

Readout calibration. Readout adaptation helps limit the duration of the drive, so we check it separately. Put $\bar{z} = W_{\mathrm{out}}\bar{h}_L + b_{\mathrm{out}}$ and $e_{\mathrm{cl}} = \sigma(\bar{z}) - \bar{y}$, the mean-logit approximation to $\bar{e}$. The

Table 10: Gate–error residual norm relative to the leading mean-teaching norm, mean ± standard deviation over three seed pairs. Absolute norms are retained in the supplementary data.

Setting	Step	Layer 1	Layer 2	Layer 3
MNIST tanh	0	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
	50	0.22 ± 0.02	0.23 ± 0.02	0.11 ± 0.03
	100	1.00 ± 0.10	0.83 ± 0.09	0.37 ± 0.09
	300	1.89 ± 0.93	1.57 ± 0.82	1.48 ± 0.89
MNIST ReLU	0	0.07 ± 0.00	0.06 ± 0.01	0.05 ± 0.00
	50	0.94 ± 0.13	0.42 ± 0.05	0.19 ± 0.07
	100	1.11 ± 0.13	0.41 ± 0.01	0.14 ± 0.05
	300	1.13 ± 0.43	0.74 ± 0.29	0.57 ± 0.22
CIFAR-10 tanh	0	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
	50	0.43 ± 0.12	0.06 ± 0.00	0.02 ± 0.00
	100	1.06 ± 0.42	0.24 ± 0.04	0.07 ± 0.02
	300	1.75 ± 0.64	1.09 ± 0.20	0.62 ± 0.26

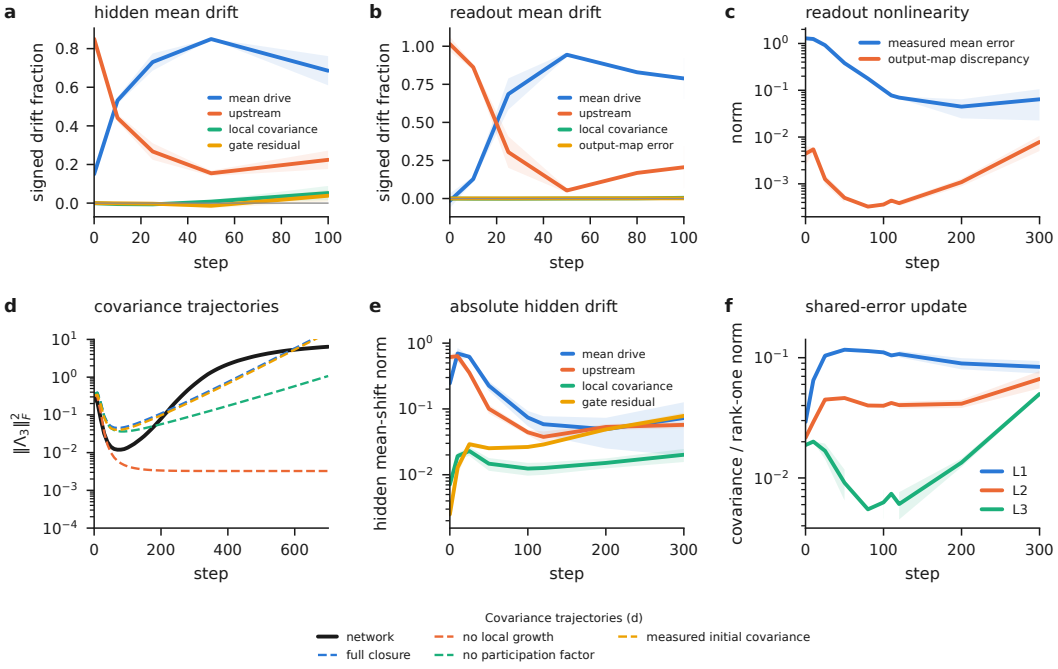

Figure 6: **Checks of the collapse and recovery approximations.** (a,b) Signed contributions along the measured virtual displacement of hidden layer-3 means and readout mean logits; finite cross terms are retained in the data. (c) Measured mean-error norm and the discrepancy from applying the sigmoid to the mean logit. (d) Layer-3 covariance energy in the network and the four recovery approximations defined in Appendix C.5 (legend below). (e) Absolute norms of the hidden layer-3 drift contributions in a. (f) Norm ratio of the covariance and rank-one contributions of shared output error in Equation (11). All panels use three baseline seed pairs; curves show means, and bands where shown one standard deviation. These are snapshot checks and model comparisons, not independent causal interventions.

same virtual full-probe step gives

$$\begin{aligned}\Delta\bar{z} = & -\eta_{\text{out}}(H_L+1)e_{\text{cl}} - \eta_{\text{out}}(H_L+1)(\bar{e}-e_{\text{cl}}) - \eta_{\text{out}}\,\text{Cov}(e,h_L)\bar{h}_L \\ & + W_{\text{out}}\Delta\bar{h}_L + \Delta W_{\text{out}}\Delta\bar{h}_L. \end{aligned} \tag{22}$$

The second term measures the output-nonlinearity approximation, the third error–activity covariance, and the last two feature motion and its finite-step interaction. Together they reconstruct the measured

Table 11: Signed projections onto the measured virtual mean displacement, divided by its squared norm. The complete components sum to one; the displayed columns omit the gate residual or output-map error and the finite cross term, so they need not sum to one. Absolute displacement norms are shown because large fractions can arise when net motion is small. Means and standard deviations use three seeds.

Location	Step	Mean term	Upstream	Covariance	$\|\Delta\mu\|$ or $\|\Delta\bar{z}\|$
Hidden layer 3	0	0.15 ± 0.01	0.85 ± 0.01	-0.00 ± 0.00	0.6715 ± 0.0644
Hidden layer 3	50	0.85 ± 0.00	0.15 ± 0.01	0.01 ± 0.01	0.2405 ± 0.0426
Hidden layer 3	100	0.69 ± 0.08	0.22 ± 0.05	0.05 ± 0.04	0.0837 ± 0.0187
Hidden layer 3	300	0.30 ± 0.38	0.35 ± 0.15	-0.02 ± 0.05	0.1083 ± 0.0242
Readout	0	-0.02 ± 0.05	1.01 ± 0.05	-0.00 ± 0.00	0.1611 ± 0.0378
Readout	50	0.94 ± 0.02	0.05 ± 0.02	0.00 ± 0.00	0.0778 ± 0.0082
Readout	100	0.79 ± 0.13	0.21 ± 0.14	0.00 ± 0.00	0.0248 ± 0.0031
Readout	300	0.13 ± 0.30	0.85 ± 0.36	0.02 ± 0.01	0.0162 ± 0.0052

Table 12: Participation prediction across distinct physical conditions. Intervals resample whole conditions, retaining their seed and layer observations (2,000 bootstrap draws). Condition means average over seeds and layers; deeper layers contain only two conditions and have no reported interval.

Summary	Conditions	Points	r	95% interval	MAE
Pooled	48	450	0.89	$[0.81, 0.94]$	0.105
Condition means	48	48	0.91	$[0.83, 0.97]$	0.101
Layer 1	48	144	0.93	$[0.87, 0.96]$	0.128
Layer 2	48	144	0.90	$[0.82, 0.95]$	0.112
Layer 3	48	144	0.86	$[0.73, 0.95]$	0.087
Layer 4	2	6	-0.16	–	0.027
Layer 5	2	6	0.41	–	0.010
Layer 6	2	6	0.89	–	0.015

mean-logit shift. Upstream motion is substantial initially and again later; the leading mean-route fraction is 0.94 at step 50 (Figure 6b). The direct output-map discrepancy is much smaller than the early mean error (panel c).

Finally, panel f measures the covariance contribution of shared output error relative to its rank-one contribution in Equation (11). The former is not generally zero. Error centering therefore tests removal of shared output error, rather than isolating the rank-one term. Together, these checks support an effective description of early collapse without establishing uniformly negligible omitted terms.

C.4 Reduced-model accuracy and detection limits

Model agreement must be assessed across distinct physical settings and must include detection failures. We therefore identify each condition by its training parameters, normalize equivalent defaults, and keep initialization and feedback seeds as separate replication indices. Run labels and logging choices do not define new conditions. The comparison uses 48 distinct settings and three paired seeds per setting, giving 144 trajectories without fitted parameters (Figure 2d,e).

Table 14 reports whether the network and model criteria detect a plateau in each trajectory (Appendix A). This is distinct from detecting representation collapse. Timing agreement is conditional on detecting a plateau in both trajectories; missed and extra plateaus are therefore reported separately. Onsets at zero remain in timing comparisons but cannot enter relative onset errors.

For the gain, output-bias, learning-rate, width and batch-size sweeps, correlations between predicted and measured participation minima over setting-by-layer means range from 0.98 to 0.99. The class-count sweep is an exception ($r = 0.10$); it includes the two-class setting, where the model predicts almost no drive. On the 37 settings covered by the closed-form estimate, correlations between condition means, each averaged over seeds and layers, are $r = 0.99$ for the reduced model and $r = 0.67$ for the estimate. Table 13 also reports the pooled condition–seed–layer comparisons.

As a null model, the closed-form collapse number of Equation (6), evaluated at initialization and mapped through the large-dose asymptote, tracks the same participation minima much less well,

Table 13: Participation minima predicted by the reduced model and by the closed-form initialization estimate, $\min(1, 1.5/\hat{\kappa}_\ell)$ with $\hat{\kappa}_\ell$ from Equation (6), on condition–seed–layer points. The closed form uses the sigmoid effective slope; the second block restricts both predictors to the settings its derivation covers. Condition means average seeds and layers, as in Table 12. Within single-parameter sweeps, the reduced-model correlation over setting-by-layer means is: gain 0.98, output bias 0.99, learning rate 0.99, width 0.98, batch size 0.99, class count 0.10; the class-count sweep includes the two-class setting, where the model predicts almost no drive.

Settings	Predictor	Points	r	MAE	r, means	MAE, means
All 48 settings	Reduced model	450	0.89	0.105	0.91	0.101
	Closed-form estimate	450	0.39	0.333	0.64	0.307
37 tanh, sigmoid, $C > 2$ settings	Reduced model	351	0.96	0.077	0.99	0.072
	Closed-form estimate	351	0.43	0.318	0.67	0.293

Table 14: Plateau detection and timing coverage. Both, network only, model only and neither partition all trajectories by detected onset. Censored model exits are included in Both but excluded from numerical exit errors. Plateau detection uses loss, not hidden-state collapse.

Comparison	Runs	Both	Network only	Model only	Neither	Exit pairs	Censored
Parameter sweeps	144	77	1	13	53	76	1
Width–depth grid	48	39	4	3	2	39	0

including on the settings its derivation covers (Table 13). The comparison supports evolving network statistics rather than relying on the initial drive alone.

We retain exits not reached within the integration horizon as censored observations. Exit errors use only uncensored pairs and state their denominator. In the narrow-network condition, the missing model exit means "not reached by step 1500." The gain-0.1 condition also has substantial timing error despite the high aggregate correlation.

Numerical resolution. To distinguish closure error from numerical error, we repeat three selected predictions with 32, 64 and 128 Gauss–Hermite nodes and with one, two and four Euler substeps per update. Initial states and training horizons remain fixed. Quadrature order has negligible effects here; finer integration shifts exits and participation minima only slightly (Table 15). These checks support the default resolution in these settings, rather than a convergence guarantee across every condition.

Width and depth. To test width and depth jointly, we use widths $\{100, 300, 600, 900\}$ and depths $\{1, 3, 5, 7\}$, each with three paired seeds and 3,000 steps for both network and model. We initialize each prediction from that network's initial state and retain every grid cell. Across all condition–seed–layer observations, participation has correlation 0.96 and mean absolute error 0.07. Figure 7a–c displays the deepest layer cell by cell; panels d,e compare exit timing. In the separate three-layer width sweep, layer-1 participation rises with width while top-layer participation remains low, and exit becomes earlier (panel f). The model misses 4 detected network plateaus and predicts 3 extra ones. Among 39 paired exits, median relative error $|t_{\text{model}} - t_{\text{network}}|/t_{\text{network}}$ is 0.24. Three grid settings also occur in the parameter sweeps, so the grid is not wholly independent validation. Coverage, censoring and per-run errors are retained in the analysis outputs (Appendix A.5).

C.5 Alternative recovery approximations

Exit-time agreement alone does not establish a particular recovery mechanism. We compare Equation (19) with three alternatives on identical per-update collapse trajectories: no accumulated local term ($J_\ell = 0$), local growth without the participation factor ($\bar{u}_\ell$ replaces $\bar{u}_\ell/\sqrt{p_\ell}$), and transmission initialized from measured hidden-layer label covariances. In the last version, the coefficient ρ_ℓ is replaced by

$$\rho_\ell^{\text{init}} = \frac{\|\Lambda_\ell(0)\|_F^2}{\bar{u}_\ell(0)\|\Lambda_{\ell-1}(0)\|_F^2}, \qquad \Lambda_0 = \text{Cov}(x, y), \tag{23}$$

so the initial transmitted amplitudes equal their measured values. These denominators are nonzero in the tested image tasks. No trajectory coefficients are fitted. Each comparison runs to the network's

Table 15: Numerical resolution in three selected settings, each with initialization/feedback seeds $(0, 0)$. Exit is in update steps for one, two or four Euler substeps. The last column is the largest absolute change in the first-300-update participation minimum across layers from one to four substeps. These use 64 quadrature nodes; orders 32 and 128 give the same displayed minima and timing. All detailed values and initial states accompany the code.

Setting	Exit: 1	2	4	Max. $\lvert\Delta \min p_\ell\rvert$
Baseline	570	567	565	0.0053
Gain 10	213	209	208	0.0054
Width 100, depth 7	2809	2801	2797	0.0023

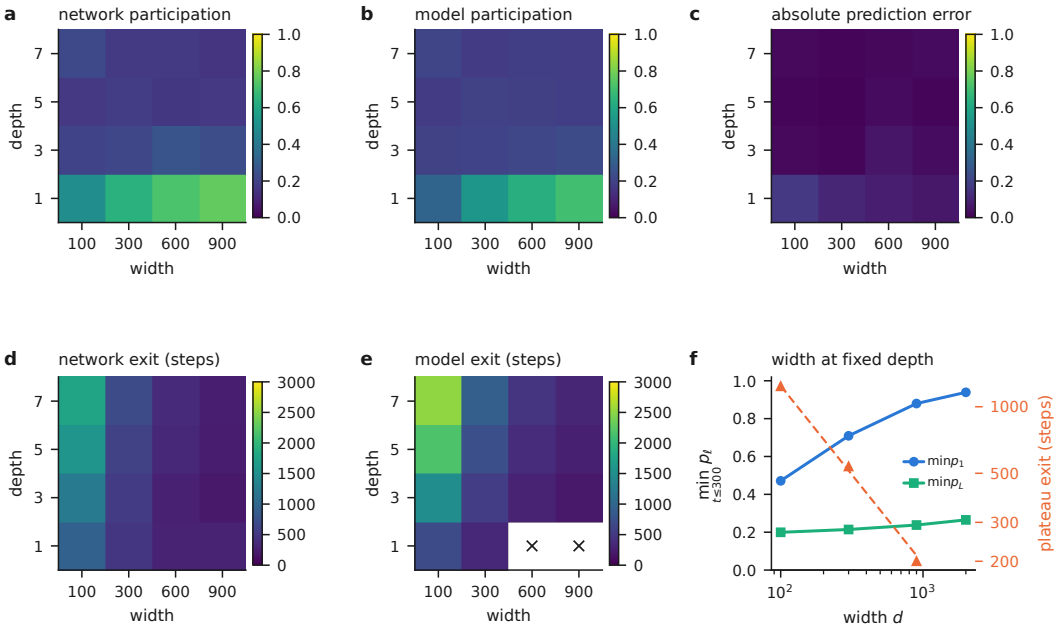

Figure 7: **Width and depth test the reduced model jointly.** (a,b) Deepest-layer participation minima within the first 300 updates in the network and model, averaged over three seeds per cell. (c) Mean absolute error, computed per seed before averaging. (d,e) Plateau exit in update steps, averaged over observed exits; a cross marks a cell with no detected plateau, and an open triangle a cell with a detected plateau but no exit within the horizon. Table 14 separates detection failures from censored exits; per-cell counts are saved with the comparison. (f) A separate three-layer sweep with widths 100, 300, 900 and 2000: layer-1 and top-layer participation minima, and plateau exit on the right axis (orange triangles; dashed line: power-law fit).

recorded training horizon and uses the original model's first persistent prior-loss band and exit criterion. Network detections retain their smoothed-loss and chance-accuracy conditions.

The full approximation predicts substantially more observed exits than the version without local growth. Removing participation dependence delays many predictions, while substituting measured initial covariances changes them less (Table 16). Covariance trajectories remain approximate: even the baseline's magnitude is not reproduced exactly (Figure 6d). This supports the extension as an exit-time approximation, with limited evidence for its full covariance dynamics.

To test configurations outside the original sweeps, we use width 450, depths 2, 4 and 6, and paired initialization–feedback seeds 10, 11 and 12. All four predictions are saved before each 3,000-update training run. These nine runs form the fresh group in Table 16. The earlier parameter and width–depth groups are shown separately because some settings overlap. Coverage retains every run; timing errors are conditional on observed pairs.

Table 16: Recovery alternatives at each network's recorded training horizon. Both, missed and extra refer to plateau onset detection. Exit pairs count observed model/network exits relative to all observed network exits; timing error is conditional on paired exits. Each variant uses the same collapse trajectory and detector. The fresh group uses new width/depth combinations and seeds, with predictions saved before training.

Group / variant	Runs	Both	Missed	Extra	Exit pairs	Median rel. error
fresh / full	9	9	0	0	9/9	0.19
fresh / no local	9	9	0	0	0/9	–
fresh / no participation	9	9	0	0	9/9	0.96
fresh / measured init	9	9	0	0	9/9	0.20
grid / full	48	39	4	3	39/43	0.24
grid / no local	48	43	0	5	7/43	1.77
grid / no participation	48	40	3	5	33/43	0.75
grid / measured init	48	40	3	3	40/43	0.25
sweeps / full	144	77	1	13	76/78	0.14
sweeps / no local	144	78	0	40	4/78	0.73
sweeps / no participation	144	77	1	13	71/78	0.49
sweeps / measured init	144	77	1	13	76/78	0.14

Table 17: Two-rate recovery fit after merging equivalent SGD gain/rate settings. The fit is conditional on observed plateau durations; intervals resample whole effective-rate conditions (2,000 draws).

Predictor	Exponent	95% interval
$a = \eta_{\mathrm{hid}} g$	-0.40	[-0.42, -0.38]
η_{out}	-0.60	[-0.62, -0.57]

The scalar closure has an additional analytical boundary: $A_0 = 0$ and $J_\ell(0) = 0$ force every A_ℓ and J_ℓ to remain zero. It cannot represent recovery driven by nonlinear features on a task with zero population input–label covariance. Such a task could nevertheless be learnable; we do not claim to have tested that possibility here.

Under plain SGD, gain and hidden learning rate combine as $a = \eta_{\mathrm{hid}} g$ for fixed normalized feedback and output rate. The recovery law therefore concerns (a, η_{out}). After merging equivalent rate settings, a joint log–log fit gives exponents -0.40 for a and -0.60 for η_{out} ($R^2 = 0.98$), using 43 observed durations among 72 unique effective-rate/seed combinations. Table 17 gives condition-bootstrap intervals. Gain and hidden-rate sweeps are therefore not independent confirmations. The exponents in Equation (8) are empirical; their intervals exclude an equal square-root split, while their sum is close to the -1 required by joint rate scaling.

Measurement-window sensitivity. The first-300-update participation minimum provides a common reporting window, but not a common dynamical phase across learning rates. We repeat the model comparison with window $T = \min[T_{\mathrm{run}}, \max(300, 300 \times 10^{-3}/a)]$, where T_{run} is the recorded horizon. Correlation is 0.89 in the fixed-window reintegration and 0.89 with this rescaled window; mean absolute error changes from 0.105 to 0.107. All pairs and windows are saved. This is a sensitivity check, not a universal phase detector.

D Decodability and gate recovery

Section 5 distinguishes retained class information from the speed of readout learning. We give the fitting and continuation protocol here, then examine how sensitivity and useful updates change during recovery.

D.1 Frozen-feature decoding and readout continuation

The frozen-feature experiment separates what a representation can support from how quickly the current readout can use it. We replay three baseline initialization–feedback pairs, $(0, 0)$, $(1, 1)$ and $(2, 2)$, with the original minibatches. Saved participation values agree with the original runs to numerical precision. Checkpoints are initialization, each run's deepest-layer participation minimum

Table 18: Frozen-feature diagnostics with 10,000 training examples for fitting and 5,000 separate validation examples for evaluation. A fixed trace-relative ridge of 10^{-3} is used for least-squares one-hot decoding. Here R uses the fitting set and decoder ridge, whereas Table 19 uses the probe estimator. The collapse checkpoint is each run's layer-3 participation minimum within its first 300 updates. Values average three seeds.

Checkpoint	Layer	Covariance energy	R	Feature RMS	Decoder accuracy
initial	1	0.472 ± 0.012	5.99 ± 0.01	0.274 ± 0.002	0.877 ± 0.007
initial	2	0.372 ± 0.029	6.14 ± 0.03	0.244 ± 0.003	0.890 ± 0.006
initial	3	0.300 ± 0.016	6.21 ± 0.02	0.222 ± 0.003	0.893 ± 0.007
collapse	1	0.440 ± 0.026	6.37 ± 0.05	0.227 ± 0.003	0.892 ± 0.006
collapse	2	0.103 ± 0.020	6.31 ± 0.06	0.105 ± 0.007	0.896 ± 0.005
collapse	3	0.014 ± 0.005	5.94 ± 0.11	0.038 ± 0.005	0.886 ± 0.005
recovery	1	4.222 ± 0.245	7.62 ± 0.03	0.498 ± 0.013	0.927 ± 0.003
recovery	2	7.994 ± 0.341	8.21 ± 0.04	0.632 ± 0.013	0.929 ± 0.001
recovery	3	10.565 ± 0.287	8.41 ± 0.02	0.696 ± 0.010	0.924 ± 0.002

within the first 300 updates, and step 3000. The geometric checkpoint rule is fixed before evaluating decoders.

For each seed, we sample 10,000 distinct training examples for fitting and 5,000 distinct validation examples for evaluation, using the same examples at every checkpoint. The two sets are disjoint. A least-squares readout fits one-hot targets from centered features with ridge $10^{-3}\,\mathrm{tr}(\Sigma_h)/d$. Its intercept is the fitting-set target mean; predictions use the largest output coordinate. The ridge is fixed, without selecting it on the evaluation set. Table 18 reports every layer, the covariance statistics on the fitting data, and held-out accuracy. Its normalized covariance statistic uses the fitting examples and the same ridge as the decoder; Appendix A.2 uses a smaller probe and a different regularization level.

To test optimization on the same representation, we freeze the features and continue training the checkpoint's sigmoid readout for 3,000 SGD updates at learning rate 10^{-3} and batch size 128. Each control receives identical sampled minibatches. Let m be the fitting-set feature mean and $s = [\mathbb{E}\|h - m\|^2/d]^{1/2}$ its scalar root-mean-square variation. The controls use h, $h - m$, or $(h - m)/s$. For the last control we initialize $W' = sW$ and $b' = b + Wm$; for centering alone, $W' = W$ and the same shifted bias. Thus all three begin with exactly the checkpoint's logits. Their subsequent updates differ because ordinary SGD depends on the feature coordinates. Means and scales are estimated only on the fitting set and remain fixed. Figure 4b reports evaluation accuracy; all checkpoints and loss curves are included in the data. These controls establish a difference in learning over a fixed budget.

Scalar rescaling has an exact optimizer interpretation. Put $c = h - m$ and write the weights in centered coordinates as $U = W'/s$, so the logit is $Uc + b'$. With fixed m, s, plain SGD on c/s gives

$$\Delta U = -\frac{\eta_{\mathrm{out}}}{s^2}\langle ec^\top\rangle, \qquad \Delta b' = -\eta_{\mathrm{out}}\langle e\rangle. \tag{24}$$

Thus it is equivalent to centered-feature SGD with weight rate η_{out}/s^2 and unchanged intercept rate, using identical minibatches. This separates an amplitude bottleneck at the chosen optimizer scale from loss of decodability. A single scalar leaves the eigenvalue ratios of the centered feature covariance unchanged; it does not correct anisotropy within that covariance. The code tests the equality of the two update trajectories.

Covariance amplitude and normalization. Table 19 compares raw and normalized label covariance at the same checkpoint, using the probe statistic defined in Appendix A.2. The marked amplitude decrease in baseline DFA coexists with a much smaller change in normalized covariance. This motivates the separate held-out decoding test; the probe statistic itself is not an accuracy estimate.

D.2 Reconstructing participation from unit means and spreads

The full Gaussian reconstruction uses both the location and spread of each unit's preactivation distribution. To assess their contributions to the early participation decline, we repeat the reconstruction

Table 19: Covariance amplitude and its normalized statistic at matched layer-3 checkpoints. Within each run, $t_* \geq 30$ minimizes covariance energy among checkpoints with a recorded R_3. The energy ratio is $\|\Lambda_3(0)\|_F^2 / \|\Lambda_3(t_*)\|_F^2$. A large amplitude loss need not imply a comparable loss of normalized covariance. R_3 is not held-out accuracy.

Condition	n	t_*	Energy ratio	$R_3(0)$	$R_3(t_*)$
Plain DFA	3	80 ± 0	28.0 ± 6.6	7.64 ± 0.03	7.42 ± 0.10
Backpropagation	3	147 ± 12	1.2 ± 0.0	7.64 ± 0.03	7.74 ± 0.00
Pre-aligned DFA	3	40 ± 0	0.9 ± 0.0	7.64 ± 0.03	7.86 ± 0.01
Center teaching signal	3	40 ± 0	0.8 ± 0.0	7.64 ± 0.03	7.69 ± 0.01
Frozen readout	3	253 ± 50	7785.5 ± 5851.3	7.64 ± 0.03	5.63 ± 0.26
Feedback gain 3	3	40 ± 0	187.4 ± 56.5	7.64 ± 0.03	7.01 ± 0.11
Balanced softmax	3	40 ± 0	0.6 ± 0.0	7.63 ± 0.03	7.83 ± 0.01
Imbalanced softmax	3	40 ± 0	1.6 ± 0.7	8.06 ± 0.16	8.96 ± 0.08

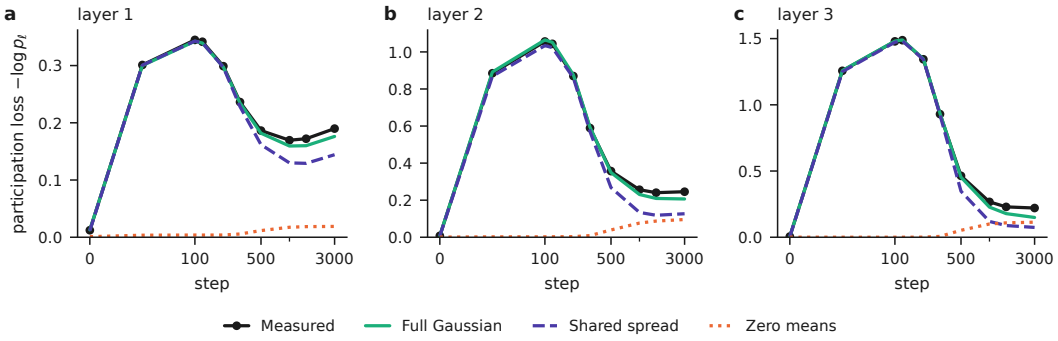

Figure 8: **Unit means and spreads in the Gaussian reconstruction.** Panels (a–c) show layers 1–3. Measured participation loss is compared with the full per-unit Gaussian reconstruction and two moment substitutions: one shared root-mean-square spread with measured unit means, or zero means with measured unit spreads. Curves average $-\log p_\ell$ across 15 baseline trajectories at each saved snapshot. The horizontal axis is linear through step 100 and logarithmic thereafter. These substitutions change the reconstruction, not the network's training.

defined in Appendix A.2 with two substitutions. The *shared-spread* reconstruction retains every measured $\mu_{\ell i}$ but replaces each $\sigma_{\ell i}$ by $s_\ell = (d_\ell^{-1} \sum_j \sigma_{\ell j}^2)^{1/2}$, computed separately for each run and snapshot. The *zero-mean* reconstruction sets every $\mu_{\ell i} = 0$ while retaining the measured $\sigma_{\ell i}$. Both recompute gate energies and then participation using Equation (2).

Figure 8a–c compares these reconstructions with measured participation loss, $-\log p_\ell$, in all three layers. The full reconstruction matches this loss closely during early collapse (median reconstructed-to-measured ratio 1.00 over steps 50–300), but departs later. At step 100, the shared-spread reconstruction gives a median reconstructed-to-measured loss ratio of 0.994 across runs and layers; the zero-mean reconstruction gives only 0.0022. Thus, differences between unit means suffice to reproduce the early concentration when a common within-unit spread is retained. Differences in spread alone produce little concentration at that time. Later, both sets of moments matter: at step 3000 the ratios are 0.515 and 0.401, compared with 0.84 for the full reconstruction.

These are conditional comparisons within the Gaussian approximation. They do not partition a causal effect, because the moments are measured after training and participation depends nonlinearly on both. They also do not remove the late mismatch of the full reconstruction.

D.3 Gate turnover and useful descent

Participation rebounds as learning resumes (Figure 1f), but the identities of sensitive units change. Figure 9a follows the top-fifth gate-energy set selected at step 120. Its overlap with the currently selected set falls toward the chance reference, arguing against recovery through one persistent selected channel. This does not exclude predictive value in early gate statistics.

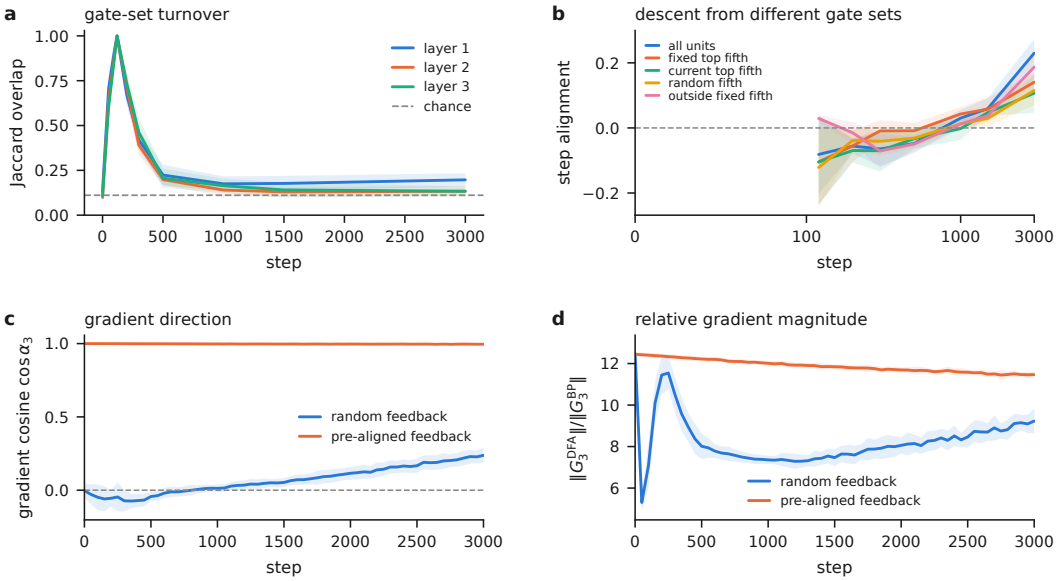

Figure 9: **Gate turnover and the direction and size of learning signals.** (a) Jaccard overlap of the current top-fifth gate-energy set with its step-120 set; dashed line: chance overlap. (b) Layer-3 alignment of norm-matched virtual weight-and-bias updates from all units, current or fixed top-fifth sets, a random fifth, and the fixed set's complement; time is linear through step 100 and logarithmic thereafter. (c,d) Layer-3 raw weight-gradient cosine and DFA-to-BP norm ratio for random and pre-aligned feedback. Curves and bands show means and one standard deviation: 15 trajectories in a,c,d and three paired seeds in b. The complement in b contains more units, but every virtual step has the same norm.

Which units support descent? Turnover alone does not identify the units responsible for learning. Norm-matched, single-layer virtual updates test the direction of descent independently of their raw magnitude (Appendix A.2). The fixed early set retains useful direction, but its complement also supports descent later (Figure 9b). These snapshot tests do not establish recovery after permanent ablation.

Direction and magnitude. The large projected descent under pre-aligned feedback reflects both a nearly BP-aligned direction and a large gradient-norm ratio (panels c,d). Gate participation determines neither factor. Figure 4c shows their distinction from sensitivity during the baseline trajectory; Table 2 summarizes the different measurements.

E Interventions, Generality and Boundary Cases

The controls in this section ask when the mechanism matters and when interventions based on it fail. We compare intervention costs and instabilities, then test changes in optimizer, architecture, readout and data. Complete condition summaries accompany the code.

E.1 Intervention Comparisons

Preventing collapse need not improve accuracy at every training time, because an intervention can also slow readout adaptation. Error centering lowers top-layer mean-activity energy at step 300 from 182 to 6.0. At step 1500, mean-error norm remains 0.61 and loss 3.35 (baseline: 0.05 and 1.31). Thus, its accuracy cost depends on when performance is measured (Figure 10a), even when early gate concentration is reduced. Comparing accuracy with participation across architectures makes the distinction explicit (panels b,c).

The local-error conditioner offers another trade-off (Figure 10d). Relative ridge is selected from $\{0.1, 0.3, 1, 3, 10\}$ by mean validation accuracy at step 3000, using the same three seed pairs reported

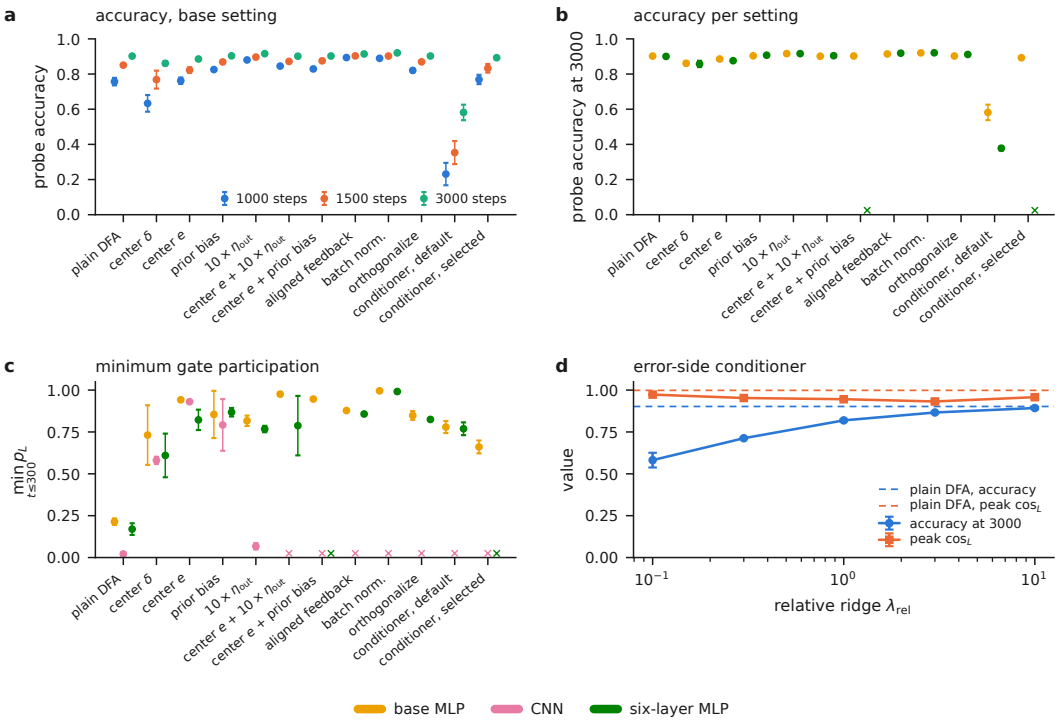

Figure 10: **Interventions.** All networks, including the CNN, are trained on MNIST. (a) Probe accuracy at 1,000, 1,500 and 3,000 steps on the base network. (b) Accuracy at 3,000 steps in the base and six-layer MLPs; CNN runs end earlier. (c) Minimum top-layer participation within 300 steps in both MLPs and the CNN. Points and bars show means and one standard deviation; crosses mark unmeasured conditions. (d) Accuracy at step 3000 and peak cosine versus relative ridge λ_{rel}; dashed lines show plain DFA. The conditioner appears at its default ridge 0.1 and, for the base MLP, at the ridge 10 selected by the same step-3000 validation accuracy. The *orthogonalize* control uses Muon-style orthogonalization, not full Muon. All measured conditions use three paired seeds.

in the figure. The default and selected settings are shown separately; selection is descriptive because there is no independent evaluation set. The selected ridge retains high cosine. A plateau is detected in only one run at the selected ridge, so Table 21 compares a common sustained-loss learning criterion across all runs. The conditioner also costs more than centering (Table 3).

Table 20 reports the large finite losses seen with some non-saturating activations; these should not be equated with an early stop from nonfinite loss. Because a linear unit's gate is constant, the gate–error residual cannot explain its failure. Even with zero mean teaching signal, local covariance can shift unit means through nonzero presynaptic means (Appendix B.2). The logs show large finite growth consistent with this route, rather than establishing its contribution to the full trajectory. With raw pixels, top-layer mean-activity energy grows from 17 to 3.5×10^{14} in the linear network and reaches 3.6×10^{6} with ReLU and 3.3×10^{7} with GELU. With per-pixel input centering at the larger rate 10^{-3} rather than 3×10^{-4}, mean activity stays small in the linear network (mean per-run peak energy 0.6). ReLU and GELU retain larger activity means. These controls do not isolate covariance-driven motion from upstream changes or residual terms, and finite-horizon growth does not establish mathematical divergence.

E.2 Optimizer dependence

To test optimizer dependence, we compare Adam (Kingma & Ba, 2015) and SGD at matched initializations, feedback draws, minibatches and update counts. Adam uses learning rates 10^{-3} and 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$, where β_1, β_2 are the moving-average decay factors and ϵ stabilizes the denominator; weight decay is zero. Each rate is tested with plain DFA, centered error,

Table 20: Final loss with non-saturating units after 3000 updates. All 27 activation–protocol combinations are retained, with three seeds each. Entries give the mean (standard deviation) in scientific notation. The hidden learning rate is shown; a faster readout uses ten times that rate. Large finite losses indicate instability and are not nonfinite early stops. The constant-predictor reference is about 3.2491.

Protocol	η_{hid}	ReLU loss	GELU loss	Linear loss
Plain DFA	10^{-4}	$2.9(0.3)\times 10^{0}$	$2.8(0.15)\times 10^{0}$	$1.8(0.083)\times 10^{0}$
Center error	10^{-4}	$3.2(0.23)\times 10^{0}$	$3.7(0.15)\times 10^{0}$	$1.4(1.3)\times 10^{1}$
Plain DFA	3×10^{-4}	$1(0.027)\times 10^{0}$	$1(0.041)\times 10^{0}$	$9.8(0.3)\times 10^{-1}$
Center error	3×10^{-4}	$1.5(1.3)\times 10^{2}$	$1.4(1.2)\times 10^{3}$	$7.4(9.3)\times 10^{10}$
Center error + faster readout	3×10^{-4}	$2.3(2)\times 10^{2}$	$2.4(1.9)\times 10^{3}$	$1.3(1)\times 10^{5}$
Backpropagation	10^{-3}	$1.4(0.058)\times 10^{0}$	$1.7(0.091)\times 10^{0}$	$9.9(0.19)\times 10^{-1}$
Prior-bias initialization	10^{-3}	$5.8(0.19)\times 10^{-1}$	$5.7(0.16)\times 10^{-1}$	$7.2(0.13)\times 10^{-1}$
Center error + faster readout	10^{-3}	$4.6(4.7)\times 10^{9}$	$5.3(4.5)\times 10^{9}$	$5.1(3.9)\times 10^{12}$
Center error + inputs	10^{-3}	$5.5(0.056)\times 10^{-1}$	$1.3(1.2)\times 10^{0}$	$3.1(0.011)\times 10^{0}$

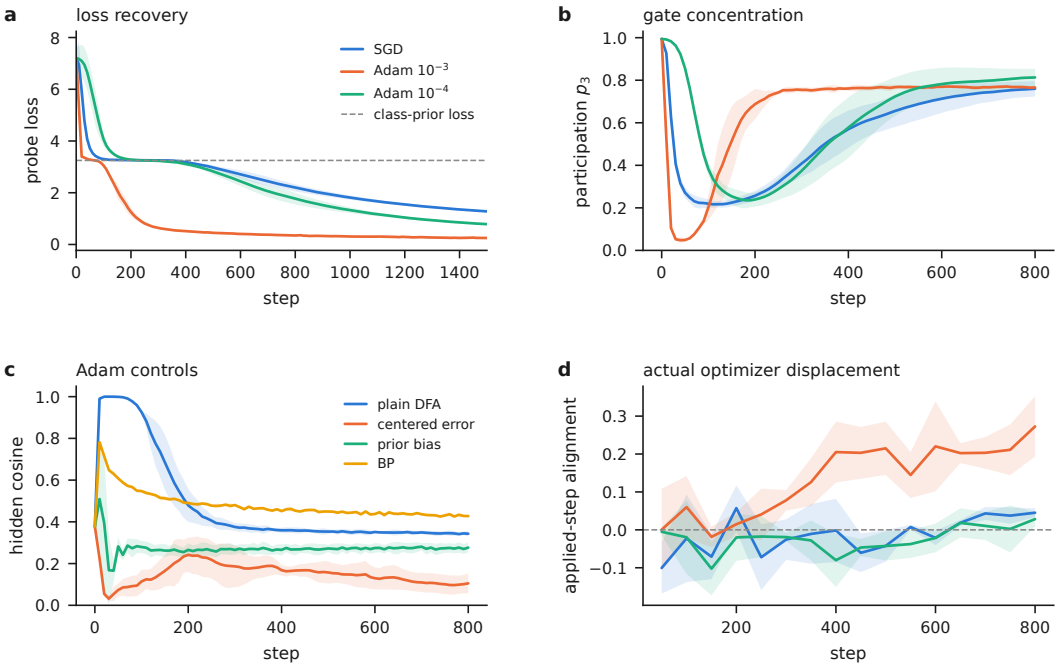

Figure 11: **Adam produces deeper gate concentration with faster loss recovery.** (a) Probe loss and the prior-loss reference. (b) Layer-3 participation. (c) Mean-error controls and BP at the larger Adam learning rate. (d) Layer-3 cosine between the actual displacement and negative true gradient, including hidden biases. Colours in b,d follow a; panel c has its own legend. Curves show means and bands show one standard deviation across three seed pairs. The SGD dose formula is not applied to Adam.

prior-bias initialization and BP, using three seed pairs and 3,000 updates. The matched SGD baselines also use three seeds; this is not a hyperparameter search.

The larger Adam rate produces deeper gate concentration but faster learning (Figure 11a,b). The plateau detector identifies a plateau in only one of three runs, so its exit at 110 is not a three-run mean. We instead compare the first logged loss below $0.9\mathcal{L}_{\text{prior}}$ that remains below it for at least 50 updates, without requiring a preceding plateau. Every Adam and SGD baseline run reaches this criterion: means are 117 for larger-rate Adam, 480 for smaller-rate Adam, and 537 for SGD (Table 21). Minimum participation is 0.046 for larger-rate Adam versus 0.214 for SGD. Centering and

Table 21: Learning time without requiring a detected plateau: first probe loss below 90% of the constant-predictor loss, remaining below it for 50 updates. All runs use the same criterion and seed pairs $(0, 0)$, $(1, 1)$ and $(2, 2)$; the shared SGD baseline is shown once, so its accuracy can differ slightly from the 15-trajectory value in Table 4. Values are means $\pm$ standard deviations; counts state how many reach it within 3,000 updates.

Condition	n	Time	Reached/n	Final accuracy
Plain SGD	3	537 ± 47	3/3	0.902 ± 0.005
Adam, $\eta = 10^{-3}$	3	117 ± 6	3/3	0.969 ± 0.004
Adam, $\eta = 10^{-4}$	3	480 ± 36	3/3	0.928 ± 0.006
Conditioner, $\lambda_{\mathrm{rel}} = 0.1$	3	1897 ± 67	3/3	0.582 ± 0.044
Conditioner, $\lambda_{\mathrm{rel}} = 3$	3	433 ± 21	3/3	0.866 ± 0.005
Conditioner, $\lambda_{\mathrm{rel}} = 10$	3	387 ± 23	3/3	0.893 ± 0.007

prior-bias initialization suppress the high-cosine transient (panel c). The complete control summary retains all detected exits with their denominators.

An adaptive optimizer changes the direction and size of the raw DFA update. To measure its effect on useful descent (Figure 11d), let θ_ℓ collect hidden layer ℓ's weights and biases. We record its actual displacement $\Delta\theta_\ell$, norm and first-order loss decrease $-(\nabla_{\theta_\ell}\mathcal{L})^\top \Delta\theta_\ell$ on the current mini-batch. Directional alignment is $\cos(\Delta\theta_\ell, -\nabla_{\theta_\ell}\mathcal{L})$, so positive values favour descent. The larger-rate Adam update becomes positively aligned during early recovery, while SGD and smaller-rate Adam remain near zero or negative for longer (Figure 11d). These diagnostics include momentum and coordinatewise scaling. They are distinct from the raw weight-gradient projection in Equation (9).

E.3 Reconstructing the original DFA protocol

Our baseline shares the readout, loss, hidden nonlinearity and input scaling of Nøkland (2016) but uses plain SGD and Xavier initialization. To reconstruct that setting, we train the original 3×800 tanh MNIST network with logistic outputs and binary cross-entropy, inputs in $[0, 1]$, RMSprop without momentum or weight decay, and minibatches of 64. DFA starts from zero weights and biases with feedback entries drawn uniformly from $[-1/\sqrt{d_\ell}, 1/\sqrt{d_\ell}]$; BP weights and biases are drawn uniformly from $[-1/\sqrt{n}, 1/\sqrt{n}]$ for fan-in n. A third arm keeps our Xavier initialization and Gaussian feedback under the same optimizer. The original learning rate, decay and ϵ are not reported; we use the PyTorch implementation with decay 0.99 and $\epsilon = 10^{-8}$, which, like the original RMSprop, has no bias correction, and test rates 10^{-3} and 10^{-4}, each with three seed pairs and 6,000 updates, the example count of the baseline (Table 22).

At zero initialization, hidden layers beyond the first have zero weight gradients and raw bias gradient $G_{b,\ell} = B_\ell \bar{e}$, a purely common-mode signal. RMSprop rescales this gradient using an exponential average of its square. With zero-initialized second moments and negligible ϵ, the first displacement of a bias coordinate i with nonzero gradient is

$$\Delta b_{\ell i} \simeq -\frac{\eta}{\sqrt{1 - 0.99}} \operatorname{sign}((B_\ell \bar{e})_i) = -10\eta \operatorname{sign}((B_\ell \bar{e})_i).$$

For a constant gradient, the displacement approaches magnitude η. More generally, the initial coordinate scaling permits a local change in mean preactivation of up to $10\eta(\|\bar{h}_{\ell-1}\|_1 + 1)$, where $\|\cdot\|_1$ sums absolute entries. When the mean term determines the gradient signs, this shift can be large even for small feedback scale or $\|\bar{e}\|$.

At the larger rate, at least 99.9% of top-layer units saturate within ten updates in all three arms, including BP. DFA leaves chance accuracy after 40 updates, whereas BP stays at chance longer (223 updates on average, varying widely across seeds). A plausible reason is that BP's error reaches a hidden layer only through every saturated layer above it, whereas DFA delivers error to each layer through a single derivative gate. At the smaller rate, few top-layer units pass $|h| > 0.9$, and DFA leaves chance after 47 updates, against 10 for BP. In fully saturated runs, gate participation stays high because all gate energies are equally small (Appendix A.2); the saturated fraction is the informative measure.

The saturation shared by BP and DFA at the larger rate is consistent with RMSprop's large initial normalized steps. With Adam at the same rate (Appendix E.2), BP does not exhibit the same near-

Table 22: Reconstruction of the original DFA protocol (Nøkland, 2016): 3×800 tanh MLP, logistic outputs with binary cross-entropy, inputs in $[0, 1]$, RMSprop and minibatches of 64, with zero initialization and uniform feedback for DFA and fan-in initialization for BP; the last rows keep Xavier initialization and Gaussian feedback. Saturated is the largest fraction of top-layer units with $|\bar{h}_{3i}| > 0.9$. Leaves chance is the first update with probe accuracy above the majority-class rate plus 0.05; learning time uses the criterion of Table 21; the 80% column is the first update reaching that accuracy; the last column is accuracy after 6,000 updates. At zero initialization, layers above the first are input-independent until their weights grow, so their peak cosine is one by construction. Means $\pm$ standard deviations over three seed pairs.

Protocol	Rate	n	Peak $\cos_3$	Saturated	Leaves chance	Learning time	80% acc.	Accuracy
DFA, zero initialization	10^{-3}	3	1.000 ± 0.000	1.00 ± 0.00	40 ± 0	40 ± 0	60 ± 0	0.973 ± 0.005
BP, fan-in initialization	10^{-3}	3	1.000 ± 0.000	1.00 ± 0.00	223 ± 162	273 ± 189	467 ± 151	0.974 ± 0.004
DFA, Xavier initialization	10^{-3}	3	1.000 ± 0.000	1.00 ± 0.00	40 ± 0	40 ± 0	60 ± 0	0.972 ± 0.010
DFA, zero initialization	10^{-4}	3	1.000 ± 0.000	0.12 ± 0.01	47 ± 6	70 ± 0	187 ± 6	0.959 ± 0.002
BP, fan-in initialization	10^{-4}	3	0.954 ± 0.002	0.25 ± 0.00	10 ± 0	10 ± 0	43 ± 6	0.969 ± 0.003
DFA, Xavier initialization	10^{-4}	3	0.987 ± 0.004	0.29 ± 0.12	40 ± 0	80 ± 10	220 ± 10	0.967 ± 0.001

Table 23: Sign-error feedback in the MNIST baseline: hidden layers receive $\mathrm{sign}(e)$ through the baseline feedback, and the readout keeps its gradient. Participation minima use the first 300 updates; final values and accuracy are at step 3000. Learning time uses the criterion of Table 21. Plain sign feedback and prior-bias initialization give identical hidden updates, because $\mathrm{sign}(e) = \mathbf{1} - 2y$ does not depend on the readout. Means $\pm$ standard deviations over three seed pairs.

Condition	n	Peak $\cos_3$	Final $\cos_3$	Min. p_3	Final p_3	Learning time	Accuracy
Sign	3	1.000 ± 0.000	0.957 ± 0.010	0.013 ± 0.003	0.055 ± 0.008	1833 ± 398	0.750 ± 0.031
Sign, prior bias	3	1.000 ± 0.000	0.957 ± 0.010	0.013 ± 0.003	0.055 ± 0.008	1830 ± 393	0.750 ± 0.031
Sign, centered	3	0.378 ± 0.021	0.026 ± 0.001	0.892 ± 0.021	0.872 ± 0.009	1073 ± 31	0.896 ± 0.005

unit-cosine transient (peak top-layer cosine 0.78, minimum participation 0.70), whereas DFA shows strong gate concentration (minimum participation 0.046). The Adam comparison uses the baseline architecture, initialization and batch size, so this contrast does not isolate optimizer bias correction. Both adaptive-optimizer experiments shorten the DFA transient relative to the hundreds of updates observed with plain SGD in the main experiments.

E.4 Sign-error feedback

Sign-error DFA broadcasts $\mathrm{sign}(e)$ instead of e (Frenkel et al., 2021). With independent sigmoid outputs and binary targets, $0 < \hat{y}_c < 1$ gives $\mathrm{sign}(e_c) = 1 - 2y_c$. Before derivative gating, the broadcast is therefore a fixed target projection plus the constant $B_\ell \mathbf{1}$. The population mean of the sign vector is $\mathbf{1} - 2\pi$, with norm $\sqrt{C}\,|1 - 2/C|$ for balanced one-hot targets. Batch means fluctuate around this fixed vector instead of shrinking as the readout learns the prior. Derivative gates still change during training, so the gated teaching signal is not fixed. The readout keeps its true gradient. We train the MNIST baseline with this signal, with prior-bias initialization, and with the batch mean of $\mathrm{sign}(e)$ subtracted before projection, using three seed pairs and 3,000 updates (Table 23).

Because $\mathrm{sign}(e)$ does not depend on the readout, prior-bias initialization leaves every hidden update unchanged, and only readout-driven differences in accuracy remain. The plain signal collapses the top layer within the first updates and keeps it collapsed through training, with low participation and slow learning. Subtracting the batch mean prevents sustained collapse and improves learning to the level of error-centered DFA. Both versions use the same discrete sign values before centering. The comparison therefore supports persistent mean drive as a cause of the sustained collapse, rather than quantization alone.

E.5 Changes in architecture, activation and objective

Architecture and feedback routing do not confine the transient to a shallow DFA network. The CNN, six-layer MLP and layerwise FA networks show strong activity convergence (Figure 12a–d), which persists at greater width (panel o). In the width-300 six-layer tanh MLP (panel b), layers 4–6 reach

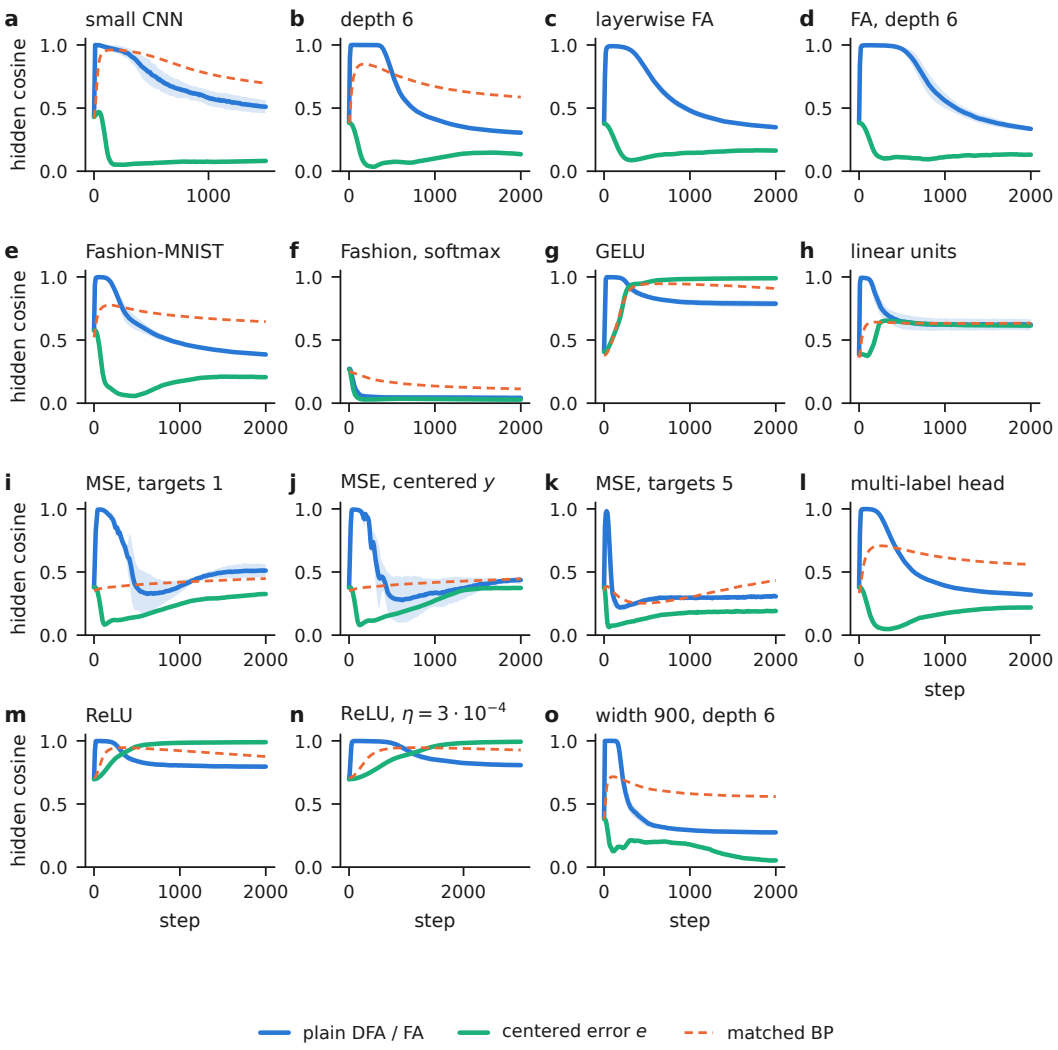

Figure 12: **Generality across protocols.** Deepest-layer cosine under plain DFA (layerwise FA in c,d), centered error, and matched BP where available; each BP curve is a single run. Panel titles specify the protocols; MSE denotes squared error. Curves show run means; bands show one standard deviation for plain DFA/FA. ReLU and GELU cosines require complementary gate and loss measurements.

cosine 1.000 and the plateau lasts 520 steps. The CNN's first block reaches cosine 0.951 and its deepest block minimum participation 0.021 (one matched BP run: 0.634 and 0.84).

The Fashion-MNIST comparison shows the effect of sigmoid versus softmax readouts on the same dataset (panels e,f). Squared-error controls separate target centering from prediction centering and show how target scale changes the relative input-dependent error (panels i–k). The augmented twelve-output classification target also exhibits a transient (panel l). Large-target squared error gives a mean-to-input-dependent error ratio of 0.36, compared with 1.28 for baseline sigmoid outputs. Across from-scratch runs, this ratio at step 10 correlates with peak cosine at $r = 0.85$ (95% CI 0.30 to 0.96; 102 trajectories across 18 settings). The percentile interval resamples whole physical settings with all their runs retained, using 2,000 draws; it describes uncertainty across settings rather than independent run-level replication.

GELU, linear and ReLU units lack tanh's two-sided saturation (panels g,h,m,n). Positive activity bias can raise cosine without suppressing input dependence; GELU is not strictly nonnegative. Gate and loss measurements are therefore needed. Error centering can also destabilize these networks, with the ReLU comparison showing learning-rate dependence (panels m,n).

Table 24: Readout initialization and target centering in the baseline architecture. Zero readout means zero output weights; sigmoid predictions are then one half at initialization. Peak cosine uses the 2000-update trajectory, and participation minima use the first 300 updates.

Readout and targets	n	Initial $\|\bar{e}\|$	Peak cosine	Min. p_3
Centered squared error, random readout	3	0.803 ± 0.346	0.995 ± 0.002	0.366 ± 0.066
Centered squared error, zero readout	3	0.022 ± 0.006	0.378 ± 0.020	0.987 ± 0.001
Same zero readout, BP	3	0.022 ± 0.006	0.378 ± 0.021	0.994 ± 0.001
Uncentered squared error, zero readout	3	0.317 ± 0.000	0.752 ± 0.014	0.944 ± 0.007
Sigmoid, zero readout	3	1.265 ± 0.000	0.998 ± 0.000	0.243 ± 0.014
Softmax, zero readout	3	0.022 ± 0.006	0.250 ± 0.007	0.940 ± 0.010

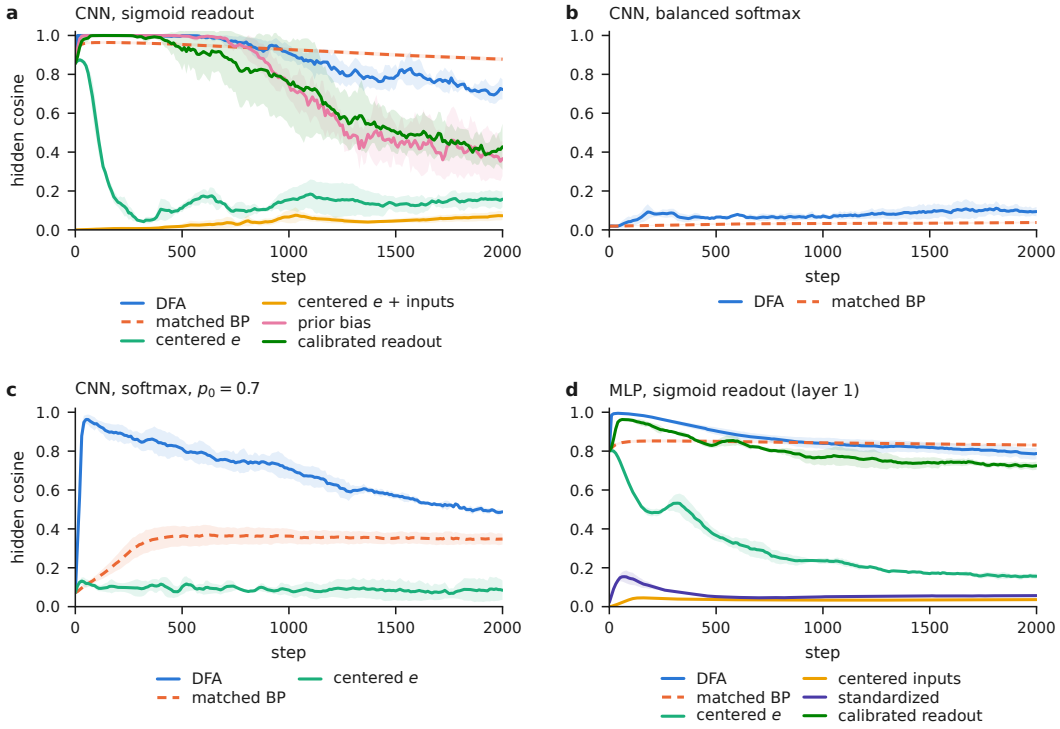

Figure 13: **CIFAR-10.** Hidden cosine of CNN block 3 (the 128-unit fully connected hidden layer; a–c) or MLP layer 1 (d) under DFA, the matched BP network and the interventions. (a) Small CNN with sigmoid readout, comparing prior-bias initialization, mean-logit calibration and centering controls. (b) CNN with balanced softmax and standardized inputs. (c) CNN with softmax and class-0 mixture weight $p_0 = 0.7$. (d) Flattened-input MLP with error centering, per-pixel centering, global standardization or mean-logit calibration. BP uses matched initialization and minibatches. Curves show means across three seed pairs, bands one standard deviation. High cosine alone does not establish collapse; Table 25 reports complementary measurements.

Mean predictions with centered targets. A random readout acting on nonzero mean hidden activity can produce nonzero mean predictions, and hence a drive even with centered targets. In the squared-error control, centered targets still give $\|\bar{e}_0\| = 0.80$, dose 1.02 and peak cosine 0.995. Zeroing the output weights reduces the initial error to 0.02 and the cosine to 0.378. The same intervention does not rescue sigmoid outputs, whose zero-logit prediction is one half at initialization (cosine 0.998; Table 24).

Table 25: CIFAR-10 controls. Conditions shared with Figure 13 use the same runs and layers. CNN geometry uses hidden block 3, the 128-unit fully connected layer after two convolutional blocks; MLP geometry uses layer 1. Entries average per-run peak cosine over 4000 updates and minimum participation within 300 updates, with sample standard deviations. Accuracy is measured at step 3000. Softmax CNNs use globally standardized inputs. Other rows use raw pixels except the stated centering or standardization controls. Imbalanced accuracy should be read with Table 26.

Condition	n	Peak cosine	Min. participation	Accuracy
CNN: plain DFA	3	1.000 ± 0.000	0.009 ± 0.001	0.276 ± 0.021
CNN: BP	3	0.964 ± 0.008	0.756 ± 0.026	0.343 ± 0.006
CNN: prior bias	3	1.000 ± 0.000	0.016 ± 0.002	0.349 ± 0.039
CNN: calibrated logits	3	1.000 ± 0.000	0.016 ± 0.010	0.340 ± 0.015
CNN: center error	3	0.876 ± 0.013	0.905 ± 0.015	0.358 ± 0.031
CNN: center error + inputs	3	0.139 ± 0.030	0.850 ± 0.023	0.349 ± 0.019
CNN: balanced softmax	3	0.150 ± 0.025	0.854 ± 0.043	0.431 ± 0.024
CNN: balanced softmax, BP	3	0.054 ± 0.007	0.992 ± 0.003	0.404 ± 0.011
CNN: imbalanced softmax	3	0.965 ± 0.029	0.146 ± 0.090	0.729 ± 0.007
CNN: imbalanced softmax, BP	3	0.375 ± 0.059	0.983 ± 0.004	0.727 ± 0.008
MLP: plain DFA	3	0.995 ± 0.001	0.111 ± 0.011	0.333 ± 0.016
MLP: BP	3	0.853 ± 0.009	0.804 ± 0.005	0.335 ± 0.006
MLP: calibrated logits	3	0.965 ± 0.006	0.346 ± 0.063	0.358 ± 0.008
MLP: center error	3	0.805 ± 0.008	0.842 ± 0.027	0.312 ± 0.021
MLP: center inputs	3	0.046 ± 0.003	0.940 ± 0.017	0.375 ± 0.005
MLP: standardized inputs	3	0.159 ± 0.036	0.937 ± 0.003	0.414 ± 0.006
MLP: center error + inputs	3	0.002 ± 0.001	0.933 ± 0.009	0.351 ± 0.004

E.6 CIFAR-10 CONTROLS AND A LEARNED MNIST PRIOR SHIFT

The larger input mean on CIFAR-10 tests whether reducing mean output error is sufficient to prevent collapse. It is not sufficient in the tanh CNN with a sigmoid readout: prior-bias initialization and mean-logit calibration leave substantial activity convergence, whereas combining input and error centering reduces it (Figure 13a). With standardized inputs, balanced softmax largely avoids this transient, but an imbalanced prior restores it (panels b,c). Balanced-softmax benchmarks (Sanfiz & Akrout, 2021) therefore resemble our low-dose control more than the uncalibrated sigmoid baseline.

The flattened MLP makes the role of the input mean visible already in layer 1 (Figure 13d). Unlike MNIST, global standardization lowers CIFAR-10 input mean energy. Even here, no single cosine threshold separates all conditions: BP can also have high cosine on uncentered images. Table 25 supplies participation and accuracy for that reason.

Label-prior shift after learning. To test whether imbalance can restart collapse after learning, we return to globally standardized MNIST with a three-layer, width-300 tanh MLP and softmax cross-entropy. Training initially samples the empirical class distribution; at the shift, each example is drawn from class 0 with probability 0.7 and from the original training distribution otherwise. The fixed probe retains the original distribution. With imbalance introduced at step 1500, the post-shift peak hidden cosine is 0.03 and balanced accuracy is 0.91 after the first shifted update (step 1500) and 0.89 at step 1750. Table 27 separates pre-shift and post-shift windows, including earlier shifts and matched BP controls. The first shifted batch has mean-error norm 0.11, falling to 0.04 within 50 updates. Confident, correct predictions contribute little error regardless of class frequency.

Calibrating an imbalanced softmax readout. Three paired runs of the same MNIST MLP use $p_0 = 0.7$ and standardized inputs. Setting the initial mean logits to the log sampling prior reduces the initial mean error to 0.03 and peak top-layer cosine from 0.96 to 0.36. However, balanced accuracy after 450 updates is 0.21 rather than 0.30; both approach 0.89 by update 3,000. BP also shows a modest early delay. Thus suppressing the high-cosine transient does not guarantee faster balanced-accuracy improvement under imbalance. These measurements average class accuracies; they do not separately resolve the learning trajectory of each minority class.

Table 26: Imbalanced softmax controls at their final recorded update (MNIST: 3000; CIFAR-10: 4000). The majority-class reference uses the training sampling prior; balanced accuracy averages class recall on the fixed probe. Raw accuracy alone can be close to the majority predictor.

Dataset / rule	n	Majority reference	Accuracy	Balanced accuracy
CIFAR-10, DFA	3	0.73 ± 0.00	0.74 ± 0.00	0.21 ± 0.03
CIFAR-10, BP	3	0.73 ± 0.00	0.73 ± 0.01	0.18 ± 0.02
CIFAR-10, centered error	3	0.73 ± 0.00	0.71 ± 0.00	0.21 ± 0.01
MNIST, DFA	15	0.73 ± 0.00	0.96 ± 0.01	0.88 ± 0.01
MNIST, BP	5	0.73 ± 0.00	0.95 ± 0.01	0.83 ± 0.01

Table 27: Class-prior shifts in the MNIST softmax MLP described in Appendix E.6. Peak cosine uses the 100 updates before and 300 updates after the shift. Error norms use the first shifted training batch and the batch 50 updates later. The fixed validation probe retains its pre-shift distribution.

Condition	n	Shift	Cos. before	Cos. after	$\|\bar{e}\|$: first	+50
DFA, early shift	3	300	0.09 ± 0.07	0.06 ± 0.04	0.473 ± 0.033	0.082 ± 0.005
BP, early shift	3	300	0.22 ± 0.01	0.21 ± 0.01	0.414 ± 0.067	0.098 ± 0.014
DFA, late shift	3	1500	0.03 ± 0.01	0.03 ± 0.01	0.105 ± 0.019	0.035 ± 0.017
BP, late shift	3	1500	0.11 ± 0.00	0.11 ± 0.00	0.172 ± 0.025	0.069 ± 0.006

E.7 Batch fluctuations and the number of classes

To separate sampling fluctuations from learning, we measure errors over the training split at fixed initialization. Let m_e and Σ_e denote their population mean and covariance, and $\bar{e}_n$ the error mean of a sampled batch of size n. The expectation below is over sampled batches at this fixed network state.

For independent sampling with replacement,

$$\mathbb{E}\|\bar{e}_n\|^2 = \|m_e\|^2 + \mathrm{tr}(\Sigma_e)/n, \qquad \mathbb{E}\|\bar{e}_n - m_e\|^2 = \mathrm{tr}(\Sigma_e)/n. \tag{25}$$

We draw 512 batches at each size in $\{16, 32, 128, 512, 2048\}$, for three seeds per setting. Fitted centered-RMS exponents range from -0.503 to -0.493, consistent with $-1/2$ (Figure 14a,b). The raw norm need not follow that law when $m_e \neq 0$.

We separately train all batch-size conditions for 1,500 updates (panels c,d). MNIST has lower peak cosine, whereas calibrated CIFAR-10 stays close to one throughout the tested batch-size range. Its dependence on batch size is not monotonic. Thus the fixed-state noise identity does not establish a universal $n^{-1/2}$ collapse law or explain the CIFAR trajectory by sampling noise alone. Training also changes gates, calibration error and feedback–activity correlations; the comparison holds updates, not processed examples, fixed.

The MNIST batch sweep uses pixels divided by 255, whereas the headline softmax condition uses global standardization. Its peak cosine therefore differs from the headline value without contradicting it. Table 28 aligns horizons and seed pairs and includes sigmoid and softmax with the same raw-pixel preprocessing.

CIFAR-100 provides a larger-class-count check with the same three-layer tanh MLP. Softmax starts with much smaller mean error than sigmoid outputs, yet both readouts produce high-cosine transients; gate concentration is substantially milder with softmax (Table 29). Matched BP also has high cosine on these uncentered images, but DFA concentrates gate energy far more strongly under both readouts and learns more slowly; after 3,000 updates both rules are still early in training. The ideal sigmoid formula is $\sqrt{C}(1/2 - 1/C)$. Uniform softmax predictions give zero mean error for a balanced class prior; more generally, mean error vanishes whenever the mean prediction matches the target prior. These runs change dataset as well as class count, so they are generality checks rather than an isolated class-count test or competitive benchmark.

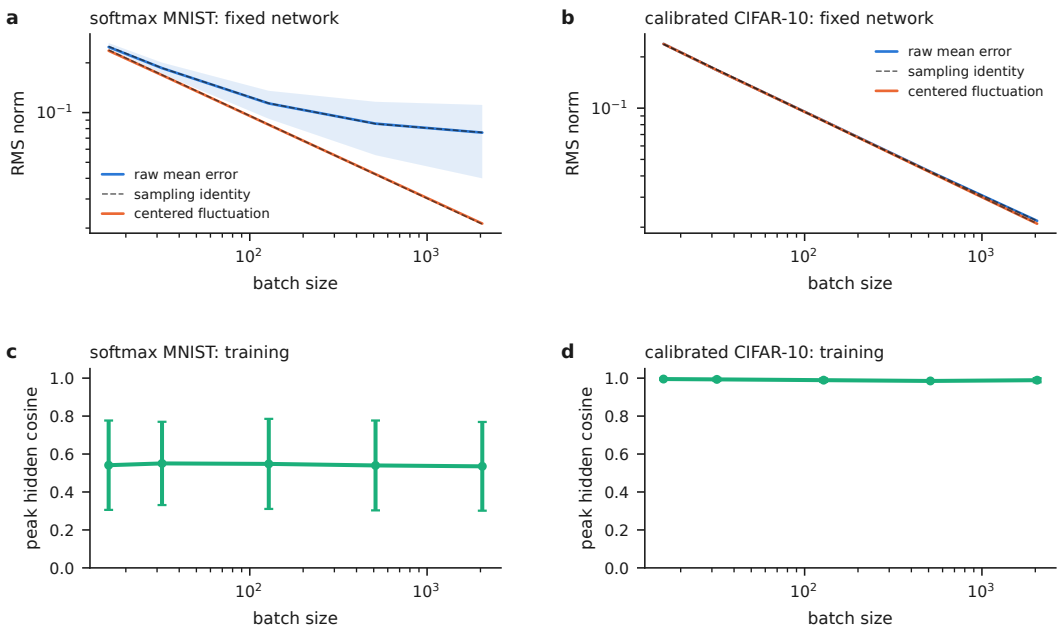

Figure 14: **Sampling noise and collapse are separate measurements.** (a,b) Root-mean-square batch-mean error at frozen initialization: raw norm and fluctuation about the population mean, with the sampling identity shown dashed. (c,d) Peak top-layer cosine during 1,500 updates with the same readout choices. Panels a,c use softmax MNIST; b,d use a mean-logit-calibrated sigmoid CIFAR-10 MLP. Curves show means and bands or bars one standard deviation across three seeds. Changing batch size also changes the number of examples processed.

Table 28: MNIST readout comparisons with preprocessing stated explicitly. All rows use the same three initialization–feedback pairs, $(0, 0)$, $(1, 1)$ and $(2, 2)$, batch size 128, and a 1,500-update window for peak cosine; participation minima use the first 300 updates. Table 4 reports larger 15-trajectory cohorts for the corresponding headline protocols; the raw-pixel softmax row is an additional preprocessing-matched comparison. Only the first two rows isolate the readout choice.

Protocol	n	Initial $\|\bar{e}\|$	Peak cosine	Min. p_3
Raw pixels, sigmoid	3	1.289 ± 0.055	0.998 ± 0.000	0.214 ± 0.019
Raw pixels, softmax	3	0.078 ± 0.032	0.548 ± 0.237	0.920 ± 0.045
Standardized, softmax	3	0.091 ± 0.002	0.297 ± 0.024	0.930 ± 0.011

Table 29: CIFAR-100 readouts in the three-layer tanh MLP under DFA and matched BP (same initializations and minibatches). High peak cosine also occurs under BP on these uncentered images; participation separates the rules, and sigmoid outputs concentrate gate energy most. Participation minima use the first 300 updates; peak cosine and final accuracy use the 3000-update run. Accuracy remains low for both rules after 3000 updates at this learning rate, so these runs test the early transient rather than final performance.

Readout, rule	n	Initial $\|\bar{e}\|$	Peak cosine	Min. p_3	Accuracy
Sigmoid, DFA	3	4.976 ± 0.015	1.000 ± 0.000	0.036 ± 0.007	0.011 ± 0.004
Sigmoid, BP	3	4.976 ± 0.015	0.999 ± 0.000	0.282 ± 0.006	0.018 ± 0.005
Softmax, DFA	3	0.046 ± 0.004	0.986 ± 0.011	0.632 ± 0.100	0.028 ± 0.004
Softmax, BP	3	0.046 ± 0.004	0.777 ± 0.020	0.954 ± 0.003	0.069 ± 0.007

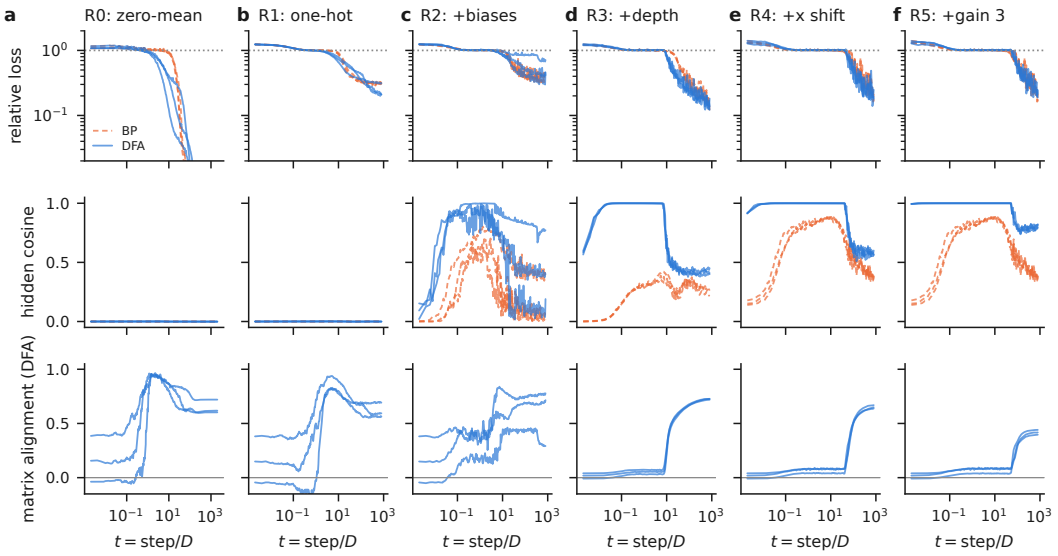

Figure 15: **The teacher–student ladder.** Test loss relative to the constant predictor (top), top-layer hidden cosine (middle) and DFA weight alignment (bottom) against $t = \text{step}/D$, where D is input dimension. DFA is solid and BP dashed; thin lines show the three individual seeds. Column a adapts Refinetti et al. (2021) to four outputs; b–f change the objective and targets, hidden biases, depth and activation, input statistics, then feedback gain. Weight alignment is the cosine between vectorized feedback and downstream weight matrices (Appendix A.2).

E.8 Comparison with align-then-memorise dynamics

The finite-width saturation transient complements analyses of lazy and feature-learning regimes (Bordelon & Pehlevan, 2023). It also differs from class-structured neural collapse (Papyan et al., 2020), self-supervised dimensional collapse (Jing et al., 2022), and previously studied BP plateaus (Fukumizu & Amari, 2000; Amari et al., 2018; Saxe et al., 2014). Saturation need not always hinder learning: Frenkel et al. (2021) exploit fixed random target projections interacting with saturating units.

A plateau in loss can have different mechanisms across learning problems. Figure 15 compares our stall with an adaptation of the scalar-regression setting of Refinetti et al. (2021): Gaussian inputs, error-function (erf) hidden units, zero-mean vector regression targets and online SGD, with a teacher of $K = 4$ hidden units, a student of $M = 4$, four outputs and input dimension $D = 500$. Column b switches to one-hot classification targets and sigmoid cross-entropy; c adds trainable hidden biases; d uses three tanh layers of width 300; e shifts and rescales inputs; and f raises feedback gain to 3. These are protocol comparisons, not single-factor interventions in every column.

The experiment uses three seeds per condition, fresh Gaussian samples at every step, hidden learning rate 0.5 and output rate one tenth as large; horizontal axes show updates divided by D. Plateau windows for the synthetic runs are detected with a $\pm 5\%$ band around the constant-predictor loss and no accuracy criterion, unlike the image-classification detector. Both window summaries are retained in the analysis outputs (Appendix A.5).

The collapse switches on at the rung that adds trainable hidden biases and is present at every rung after it; adding nonzero-mean targets alone, with zero-mean inputs and no biases, does not produce it. In rungs R3–R5, BP also stays near the constant predictor for a similar time without full collapse, so in this synthetic task collapse does not set the plateau length. In these controls, nonzero-mean targets supply a mean error and trainable biases create an additional mean-drive channel. A flat loss near the constant predictor need not involve collapse: the cosine and alignment rows distinguish a pre-alignment plateau with hidden cosine near one from the post-alignment plateau in the regression control, which is reached at high weight alignment with the cosine low; the two can also coexist within one trajectory. This comparison is empirical, not an if-and-only-if consequence of Proposition 1.