What Accuracy and Gradient Cosine Miss: Evaluating Feedback Alignment via Scale Stability, Reference Validity, and Depth Utility

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite the success of deep learning, training deep networks in biologically plausible and hardware-efficient ways remains an open challenge. Feedback alignment (FA) methods address this by replacing backpropagation's symmetric backward weights with fixed random matrices, but their effectiveness depends critically on whether they can be accurately evaluated. The standard evaluation relies on two quantities: task accuracy and cosine similarity between the method's credit signal and the backpropagation gradient. We show that this reporting pair is insufficient by identifying two independent failure modes, both silent under current reporting: (1) measurement degeneracy, where the BP reference gradient collapses to the numerical floor in terminal-LayerNorm residual architectures, rendering cosine uninterpretable; and (2) aggregation collapse, where the aggregate cosine masks layerwise heterogeneity that concentrates credit at one end of the network. To address these limitations, we propose a diagnostic evaluation protocol based on three checks—scale stability, reference validity, and depth utility—together with per-layer rather than aggregate cosine reporting. Across five architectures and 125 trained models, the standard reporting pair gives no signal of failure in any audited case, while our protocol identifies all failures with wide calibration margins. The two failure modes are causally independent: a per-block scale penalty alleviates Mode 1 (residual scale explosion driving reference collapse) without affecting Mode 2 (cosine ranking that contradicts every functional metric we measured).

1 Introduction

The backpropagation algorithm trains neural networks by propagating error gradients layer by layer from the output back to the input. Despite its effectiveness, BP relies on two mechanisms that are both biologically implausible and computationally constraining: weight symmetry between the forward and backward paths, and sequential layer-by-layer updates that prevent parallelization [4, 6, 9, 13, 19]. Feedback alignment (FA) methods address these constraints by replacing the backward weights with fixed random matrices [3, 5, 13, 15, 17, 20]. Direct feedback alignment (DFA) further removes the sequential dependence by projecting the output error directly to each hidden layer through independent random connections [16], and has been shown to train modern architectures including transformers and graph networks with performance approaching BP [1, 12]. These properties make the FA family a candidate for both biologically plausible learning and hardware-efficient training [8]. Yet as these methods scale to deeper residual architectures, a prior question becomes pressing: how should we evaluate whether they actually work?

The standard evidence for a feedback alignment method is two quantities: task accuracy, which measures whether the network learned, and cosine similarity between the method's credit signal and the backpropagation gradient, which measures whether the learning signal points in roughly the right direction [3, 12, 13, 16, 17]. A method that reaches nontrivial accuracy and reports positive alignment

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

is typically interpreted as having trained the network with useful credit assignment. This pair has been the primary reporting convention across a decade of FA research, and the alignment angle in particular has been explicitly recommended as the basis for training best practices [11, 13, 17].

The reporting pair gives no signal when a method has not actually trained the network. On a representative residual setting (Figure 1), both FA and DFA report non-trivial accuracy and positive aggregate cosine—the kind of numbers a reader would interpret as evidence of credit reaching the network. Yet both fall below an architecture-matched frozen-blocks baseline whose residual blocks were never trained at all: neither method's trained network outperforms the same architecture with random blocks. The standard reporting pair shows no sign of this, for two independent reasons. First, the cosine measurement itself can be invalid: in terminal-LayerNorm residual architectures, residual-stream scale explosion compresses the BP reference gradient to the numerical floor through the LayerNorm Jacobian, so the reported cosine is computed against floating-point noise rather than a meaningful direction; and even when the reference is meaningful, an aggregate cosine can appear positive because the per-layer contributions are concentrated at one end of the network. Second, accuracy alone cannot distinguish whether deep blocks help or hurt the network's prediction.

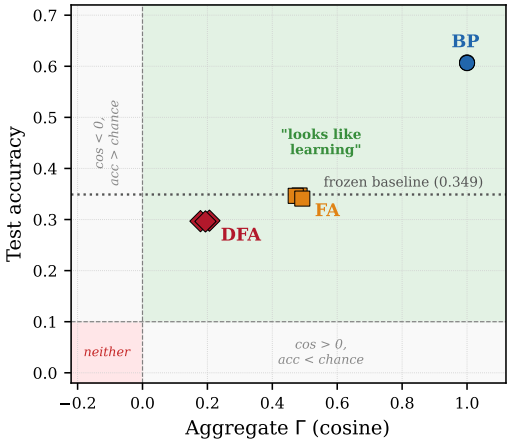

Figure 1: Standard reporting pair on the representative setting (d=512, L=2 ResMLP, three seeds): both FA and DFA report non-trivial accuracy and positive aggregate cosine.

Method	Test acc	Γ	vs Frozen	Verdict
BP	0.607	≈ 1.0	+25.7 pp	pass
FA	0.345	+0.48	−0.4 pp	fail
DFA	0.297	+0.19	−5.2 pp	fail

Table 1: Same data as Figure 1; trained blocks of FA and DFA fall below an architecture-matched frozen-random-blocks baseline.

We demonstrate these failures through a controlled audit of three methods—backpropagation, feedback alignment [13], and direct feedback alignment [16]—on residual architectures. FA and DFA share the same local loss and differ only in how credit reaches the deep layers, providing a controlled comparison on which our protocol identifies both failures with wide calibration margins while the standard pair does not.

Contributions.

- We identify two independent failure modes of the standard FA evaluation pair (accuracy + aggregate cosine) on deep residual architectures: *measurement degeneracy*, where the BP reference gradient collapses to the numerical floor and makes cosine uninterpretable, and *low intrinsic credit-direction quality*, where deep-layer credit is essentially unaligned with BP even when the reference is meaningful.
- We provide a three-diagnostic evaluation protocol that detects both modes. On a representative setting where both FA and DFA report non-trivial accuracy and positive cosine alignment, the standard pair gives no signal that either method's deep blocks are unused; the protocol identifies both as failures with wide calibration margins.
- We show that cosine alignment to the BP gradient, even when measured validly, does not predict depth utility: under matched conditions, three independent functional metrics rank the audited methods in an order that cosine contradicts.
- We validate the protocol across 5 architectures, 125 trained models, and 17 experimental settings, and release a reference implementation.

2 Motivating Observations

We fix a common training recipe across all primary-audit experiments: pre-LayerNorm [2, 21] ResMLPs [18] on CIFAR-10 [10], trained for 100 epochs with AdamW [14] (learning rate 10^{-3}, weight decay 0.01, cosine schedule, batch size 128) across three seeds. Architecture depth L and

width d are specified per experiment. Three methods are trained on this identical setup, differing only in how each layer's credit signal a_l is computed.

Backpropagation (BP) computes the exact gradient via the chain rule: $a_l = \partial L / \partial h_l$.

Feedback Alignment (FA) [13] propagates credit sequentially through fixed random matrices: starting from the exact output-layer gradient $\partial L / \partial h_L$, each layer receives $a_l = B_l\, a_{l+1}$, where $B_l \in \mathbb{R}^{d \times d}$ is fixed at initialization.

Direct Feedback Alignment (DFA) [16] projects the output error directly to each layer, bypassing all intermediate layers: $a_l = B_l^\top e_T$, where $B_l \in \mathbb{R}^{C \times d}$ is a fixed random matrix and $e_T = \hat{y} - y$ is the output error.

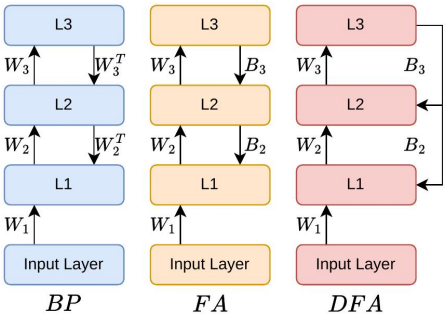

Figure 2: Backward architectures

FA and DFA share the same local loss: each block f_l is updated by reducing $-\langle f_l(h_l), a_l \rangle$. Neither loss contains a penalty on $\|f_l(h_l)\|$. The only difference between the two methods is how a_l is computed: FA preserves sequential structure in the credit path, DFA does not. A frozen-blocks baseline trains only the embedding, LayerNorm, and classification head while holding all residual blocks fixed at their random initialization; it reaches 0.349 ± 0.002 across the same three seeds.

The failure shown in Figure 1 has two independent sources internal to the standard reporting pair. To isolate them, we turn to the per-layer state of a related setting at $d = 256, L = 4$ (Figure 3). Two independent inconsistencies emerge from the per-layer data, each one severing the link between the standard reporting pair and the question it is meant to answer.

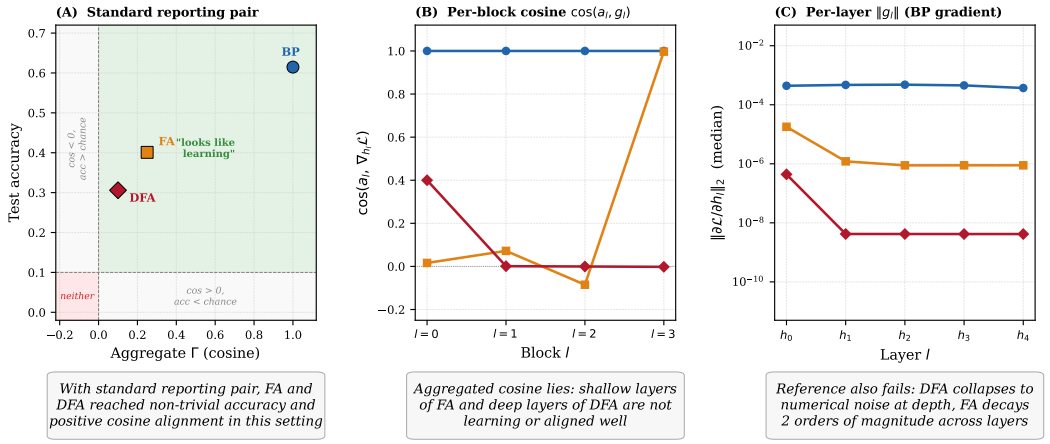

Figure 3: Per-layer state on the standard primary-audit setting (ResMLP $d = 256$, $L = 4$, CIFAR-10). (A) Standard reporting pair: all three methods report non-trivial accuracy and positive aggregate cosine. (B) Per-block cosine reveals that FA's shallow blocks and DFA's deep blocks contribute no aligned credit. (C) Per-layer BP gradient norm decays across FA's depth and collapses to numerical noise in DFA, leaving cosine measurements at depth without a meaningful reference.

The aggregate cosine masks opposite per-layer patterns. On this setting, FA reports an aggregate cosine of $\Gamma \approx +0.23$ and DFA reports $\Gamma \approx +0.10$; both look like partial alignment to the BP gradient. Per-block cosine reveals that these aggregates are produced by inverse distributions (Figure 3B). FA's shallowest block contributes near-zero cosine ($\approx +0.01$) while its deepest block contributes $+0.95$; the aggregate is carried almost entirely by the deep end of the network. DFA shows the opposite shape: its shallowest block contributes $+0.39$ while its deepest block contributes ≈ 0; the aggregate is carried almost entirely by the embedding. Two networks whose credit reaches opposite ends of the depth axis produce aggregate cosines that read as the same kind of evidence. Aggregation collapses this distinction, so the aggregate Γ cannot in principle answer whether credit reaches the deep layers.

The reference gradient can collapse below the cosine clamp. The BP gradient norm $\|g_l\|$ along depth tells the second story (Figure 3C). BP holds $\|g_l\|$ within an order of magnitude of $\sim 4 \times 10^{-4}$ across all layers, providing a reference of consistent scale. FA decays from $\sim 2 \times 10^{-5}$ at the embedding to $\sim 9 \times 10^{-7}$ at the deepest layer, but every layer remains above PyTorch's default

cosine-similarity denominator floor of $\varepsilon = 10^{-8}$. DFA collapses by three orders of magnitude after the embedding—from $\sim 4 \times 10^{-7}$ at h_0 to $\sim 4 \times 10^{-9}$ at h_1 and below—and remains beneath ε for every deeper layer. When the denominator is supplied by the clamp rather than the reference vector, $\cos(a_l, g_l)$ is no longer comparing the credit direction to a meaningful BP direction; it is comparing it to a normalized floating-point residual.

The two failures are independent. On this setting, FA fails the first check (its aggregate is dominated by a single end of the network) but passes the second (every reference norm remains measurable); DFA fails both. The aggregate-dominance failure does not require reference collapse, and reference collapse does not require any particular layerwise distribution of cosine. Each one severs the standard reporting pair from its intended meaning on its own. Neither method universally avoids either mode—Section 4 shows that both modes can be triggered by either method in other settings, and that interventions on one mode do not resolve the other.

3 Failure Modes

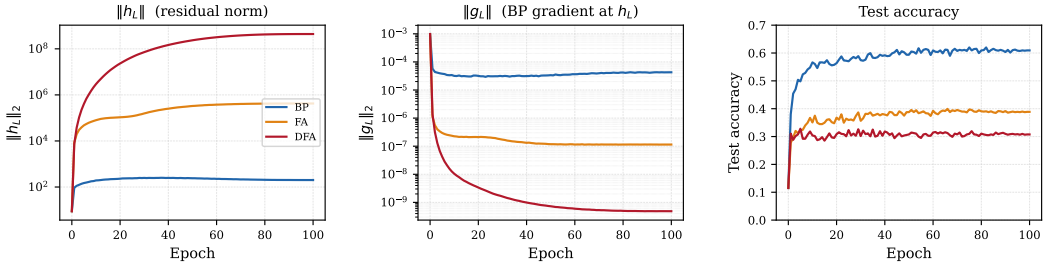

Figure 4: Temporal evolution of $\|h_L\|$, $\|g_L\|$, and test accuracy on ResMLP $d = 256$, $L = 4$, terminal LN. DFA's residual norm grows four orders of magnitude and its BP reference gradient collapses to $\sim 10^{-9}$, the regime where cosine similarity becomes a comparison against floating-point noise. FA exhibits the same growth direction at attenuated magnitude, with $\|g_L\|$ stabilizing two orders of magnitude above this regime.

3.1 Mode 1: Residual scale explosion drives reference collapse

In Mode 1, the BP reference gradient at deep layers is compressed below the numerical floor used by the cosine implementation, so the cosine reported by the standard pair is computed against a normalized floating-point residual rather than against the BP direction. The mechanism is a chain of three steps connecting the local loss form to this measurement collapse.

The local loss $-\langle f_l(h_l), a_l \rangle$ is unbounded in $\|f_l(h_l)\|$: any direction in which the block output grows along a_l reduces the loss further. In a pre-LN residual block $h_{l+1} = h_l + f_l(h_l)$, growth in $\|f_l(h_l)\|$ accumulates directly into the residual stream, so $\|h_L\|$ increases along training. The terminal LayerNorm rescales h_L by its norm, giving a Jacobian whose spectrum scales as $1/\|h\|$ [21]. The deepest BP reference gradient $\|g_L\| = \|\partial L/\partial h_L\|$ inherits this $1/\|h\|$ factor, so $\|g_L\|$ shrinks at the same rate that $\|h_L\|$ grows.

Figure 4 shows this chain in the DFA trajectory on the primary-audit setting: $\|h_L\|$ grows from $\sim 10^2$ at initialization to $\sim 10^8$ by epoch 100, while $\|g_L\|$ collapses from $\sim 10^{-3}$ to $\sim 10^{-9}$ over the same window—four orders of magnitude in each direction, in tandem. By epoch 20, $\|g_L\|$ is already below PyTorch's default cosine-similarity denominator floor of $\varepsilon = 10^{-8}$, and remains there for the rest of training.

FA shares DFA's local loss form but exhibits a markedly attenuated chain on the same architecture (Figure 4): $\|h_L\|$ grows to $\sim 10^5$ rather than $\sim 10^8$, and $\|g_L\|$ stabilizes at $\sim 10^{-7}$, within the regime where cosine is computed against a meaningful reference. The local loss form is therefore not sufficient for Mode 1 to become a measurement failure; the credit propagation rule determines whether the chain reaches the cosine clamp. FA is not generally safe from Mode 1—the same mechanism produces full reference collapse on terminal-LN architectures with different geometry (Section 4).

3.2 Mode 2: Aggregation collapses layerwise heterogeneity

In Mode 2, the per-layer cosine values along depth follow distributions that concentrate at one end of the network, and the aggregate cosine averages over this concentration; the standard pair's Γ no longer indicates whether credit reaches the deep layers, only that some layers are aligned somewhere along the depth. Section 2 already showed two such concentrations on the same architecture: FA's per-block cosine rises from $\approx +0.01$ at the embedding to $+0.95$ at the deepest block, while DFA's drops from $+0.39$ at the embedding to ≈ 0 at the deepest block. Both produce aggregate cosines that read as partial alignment.

The two concentrations follow from the credit propagation rule. DFA's $a_l = B_l^\top e_T$ is a fixed random projection of the output error and contains no information about the forward state at layer l. The embedding sits one block away from the output and receives credit whose correlation with the BP gradient is preserved by the projection's geometry, but every deeper block receives a signal generated independently of its forward state, and the cosine reflects that independence—near zero in expectation. FA's $a_l = B_l\, a_{l+1}$ inherits the deepest block's exact gradient $\partial L/\partial h_L$ and degrades upstream by an additional random matrix product per layer. The two rules produce alignment patterns that are inverse but symmetric in their effect on the aggregate.

Per-layer cosine reporting [13, 17] addresses this specific aggregation failure but does not recover the standard pair: it carries no information about whether each layer's reference gradient is numerically valid (Mode 1), and it does not indicate whether the trained depth as a whole contributes to the network's prediction (Section 5). Section 4.2 establishes that Mode 1 and Mode 2 are causally independent failure modes, both observationally and under a penalty intervention that alleviates Mode 1 without affecting Mode 2.

4 Validation

4.1 Scope of Mode 1 across architectures

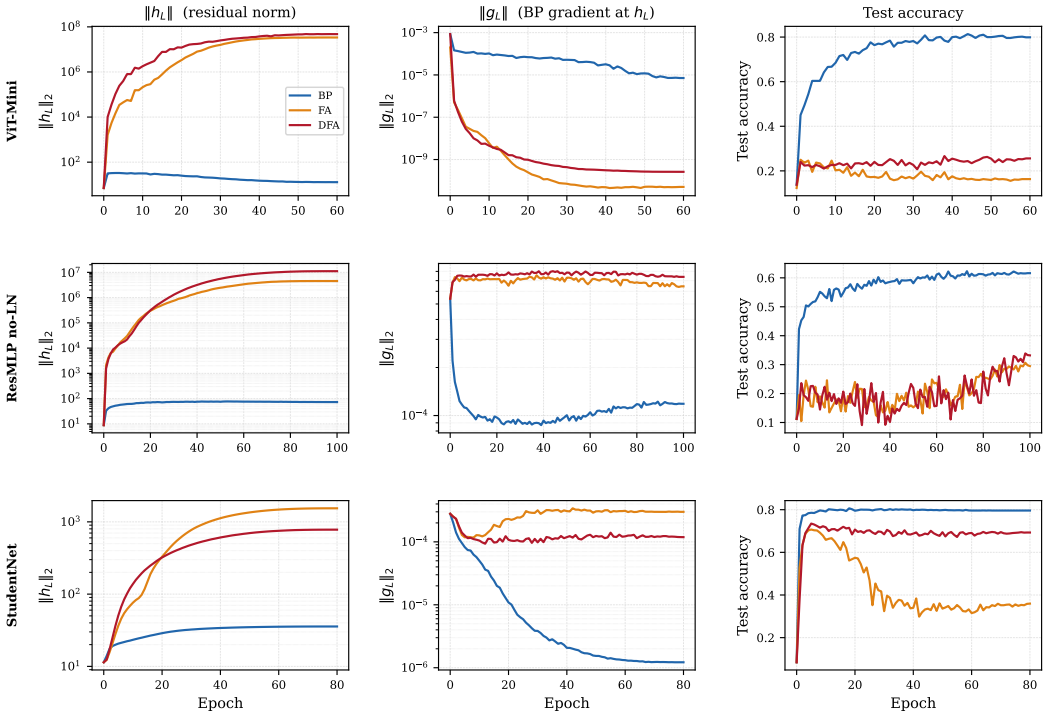

Figure 5: Temporal evolution of $\|h_L\|$, $\|g_L\|$, and test accuracy across three architectures: ViT-Mini (terminal LN), ResMLP $d = 256$, $L = 4$ with terminal LN removed, and StudentNet (no LN). The $\|g_L\|$ y-axis ranges differ across rows to expose within-row dynamics; the qualitative split is between collapsing trajectories on ViT-Mini and stable trajectories on the two no-LN architectures.

The two failure modes documented in Section 3 on ResMLP have different cross-architecture scopes. We examine three additional settings that vary terminal LayerNorm presence while holding

the other components of the architecture roughly fixed (Figure 5): a vision transformer ViT-Mini ($d = 128, L = 4$, cls token + terminal LN), the same ResMLP $d = 256, L = 4$ used in the primary audit but with the terminal LayerNorm removed, and StudentNet ($d = 128, L = 4$, no LN). Across these three settings, Mode 1a (residual scale growth under FA and DFA) appears in all three; Mode 1b (reference gradient collapse) appears only in the architecture with terminal LayerNorm.

On ViT-Mini, both FA and DFA reproduce the full Mode 1 chain seen on ResMLP: $\|h_L\|$ grows to $\sim 10^7$ and $\|g_L\|$ collapses to $\sim 10^{-9}$ within 30 epochs, well below the regime where cosine similarity computes against a meaningful reference. The transformer architecture provides no protection from Mode 1 once terminal LN is in place. On ResMLP with terminal LN removed and on StudentNet (no LN), the residual stream still grows under FA and DFA—$\|h_L\|$ reaches $\sim 10^7$ on no-LN ResMLP and $\sim 10^3$ on StudentNet—but $\|g_L\|$ stabilizes at or above $\sim 10^{-4}$ on both, in the regime where cosine remains a valid measurement. The unbounded growth incentive of the local loss is architecture-agnostic, but its translation into measurement collapse requires terminal LayerNorm to compress the BP reference gradient.

Table 2 (✓=passes, ✗=fails, FA/DFA) shows Mode 1a failing universally—the unbounded growth incentive is architecture-agnostic—while Mode 1b tracks terminal LN presence. The same-backbone control isolates this: removing terminal LN from the primary-audit ResMLP flips the Mode 1b verdict for both methods without changing Mode 1a. The depth-utility consequence on ViT-Mini is severe—the frozen-blocks baseline reaches 0.570 ± 0.003 while FA and DFA reach 0.163 and 0.256, trained blocks underperforming random blocks by more than 30 percentage points.

Architecture	LN	Scale stable	Reference valid
ResMLP	yes	✗ / ✗	✓ / ✗
ResMLP, no LN	no	✗ / ✗	✓ / ✓
ViT-Mini	yes	✗ / ✗	✗ / ✗
StudentNet	no	✗ / ✗	✓ / ✓

Table 2: Mode 1 verdict per architecture (FA/DFA).

4.2 Penalty intervention separates the two modes

The two failure modes can be intervened on independently. Adding a per-block scale penalty $\lambda \cdot \|f_l(h_l)\|^2$ to the local loss on each residual block—without modifying the credit rule, the optimizer, or any other component—suppresses Mode 1 in a dose-response manner on methods where Mode 1 is severe; at the dose where Mode 1 is fully alleviated but Mode 2 is not yet engaged, the deep-layer cosine remains at the vanilla value, separating the two modes causally.

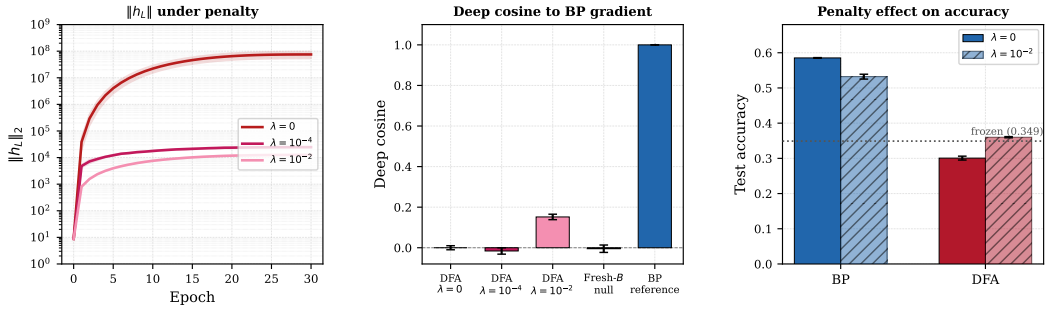

Figure 6: Penalty intervention separates Mode 1 from Mode 2 on DFA, ResMLP $d = 256, L = 4$, three seeds, 30 epochs. **Left:** residual norm under $\lambda \in \{0, 10^{-4}, 10^{-2}\}$ shows dose-response containment. **Middle:** deep cosine across conditions. At $\lambda = 10^{-4}$ the penalty contains $\|h_L\|$ and lifts $\|g_L\|$ above the cosine clamp, but deep cosine remains at the vanilla value; at $\lambda = 10^{-2}$ deep cosine recovers partially to $+0.152 \pm 0.013$. **Right:** BP and DFA accuracy with and without $\lambda = 10^{-2}$ penalty against the frozen-blocks baseline. BP loses 5.3 pp under penalty (capacity cost) but stays 18.3 pp above frozen; DFA gains 5.9 pp but only reaches 1.1 pp above frozen.

We sweep $\lambda \in \{0, 10^{-4}, 10^{-2}\}$ on DFA, ResMLP $d = 256, L = 4$, three seeds, 30 epochs (Figure 6, left and middle panels). Vanilla DFA reaches $\|h_L\| \sim 4 \times 10^8$ and $\|g_L\| \sim 5 \times 10^{-10}$, with deep cosine at the vanilla zero. At $\lambda = 10^{-4}$, the residual stream is contained ($\|h_L\| \sim 2.3 \times 10^4$) and $\|g_L\| \sim 5 \times 10^{-6}$ recovers well above the clamp—Mode 1 is alleviated—while deep cosine remains at -0.016 ± 0.016, indistinguishable from zero. At $\lambda = 10^{-2}$, $\|h_L\|$ is further contained and deep cosine rises to $+0.152 \pm 0.013$, indicating partial recovery of Mode 2. The two modes respond at

different intervention strengths, and Mode 2 does not follow Mode 1's recovery passively. FA on ResMLP exhibits only mild Mode 1 and produces no dose response (Appendix A).

Two control experiments rule out alternative explanations of the deep-cosine recovery: a BP control shows the penalty does not artificially lift cosine on a method whose credit is already exact (Appendix C.1); a fresh-B null calibration shows the recovered cosine is specific to the matrices used during training rather than trivial adaptation (Appendix C.2).

Two observational checks corroborate the intervention. Across methods on ResMLP $d = 256, L = 4$, FA's $\|g_l\|$ remains above the cosine clamp at every layer (Figure 4, middle panel), so its standard pair fails only through Mode 2; DFA on the same architecture fails both. Along training time, DFA's deep-layer cosine is already -0.008 ± 0.013 at epoch 1 while $\|g_L\| \sim 10^{-7}$ remains above ε: Mode 2 is present before Mode 1 develops. Together with the penalty intervention, this establishes Mode 1 and Mode 2 as causally independent failure modes the protocol must diagnose separately. Neither method universally avoids either mode—FA on ViT-Mini [7] exhibits both despite passing Mode 1b on ResMLP—so the modes are properties of (architecture, credit-rule) pairings rather than of methods in isolation.

4.3 Cosine cannot predict depth utility

Mode 1 and Mode 2 together explain why the standard reporting pair fails on settings where the measurement is invalid or the aggregate masks layerwise heterogeneity. A natural follow-up is whether cosine alignment, when measured validly and reported per-layer, suffices on its own—that is, whether the depth-utility check (the frozen-blocks comparison) is redundant once Mode 1 and Mode 2 are addressed. We show it is not: even on settings where Mode 1 has been alleviated and the cosine measurement is valid, deep-layer cosine to the BP gradient does not predict whether depth is being used.

To establish this, we introduce two diagnostic probes designed to vary credit quality in the cosine space:

State Bridge (SB) learns a state predictor $G_\psi(h_l, t_l)$ that estimates the deepest hidden state h_L from the current layer's state; the credit signal is the gradient of a cross-entropy loss computed by passing $G_\psi(h_l, t_l)$ through the classification head, $a_l = \nabla_{h_l} \mathrm{CE}(\mathrm{head}(G_\psi(h_l, t_l)), y)$.

Credit Bridge (CB) learns a value network $V_\phi(h_l, t_l)$ that directly estimates a scalar value at the current layer; the credit signal is its input gradient, $a_l = \nabla_{h_l} V_\phi(h_l, t_l)$.

SB and CB are not proposed as competitive FA methods; they exist to produce credit signals whose deep-layer cosine to the BP gradient differs systematically from DFA's, so that we can ask whether these cosine differences correspond to functional differences in credit quality. Both probes train under the same per-block penalty $\lambda = 10^{-2}$ used in Section 4.2 to alleviate Mode 1, the predictor and value network themselves are unpenalized.

Table 3: Four methods under matched penalty rescue ($\lambda = 10^{-2}$, ResMLP $d = 256, L = 4$, three seeds, 30 epochs). Three functional metrics agree on SB $\gg$ rest; deep cosine ranks the methods in a different order.

Method	Test acc	Deep cosine	Nudging ($\eta = 0.01$)	Train loss Δ
SB + pen	$\mathbf{0.453 \pm 0.003}$	$+0.322 \pm 0.008$	$\mathbf{-1.93 \times 10^{-3}}$	$\mathbf{-0.447}$
FA + pen	0.369 ± 0.003	$+0.423 \pm 0.006$	-2.09×10^{-4}	-0.128
CB + pen	0.360 ± 0.004	$\mathbf{+0.679 \pm 0.010}$	-4.26×10^{-4}	-0.121
DFA + pen	0.360 ± 0.002	$+0.152 \pm 0.013$	-4.98×10^{-5}	-0.095

Table 3 reports four methods under matched penalty rescue. The functional metrics agree on a single ranking: SB $\gg$ FA $\approx$ CB $\approx$ DFA. SB reaches 0.453 accuracy, 9 percentage points above the others; its nudging effect (the change in test loss after a small step in the credit direction) is 4 to 40 times larger than the other three; its training loss decreases by -0.447 over 30 epochs, 3 to 5 times more than the others. Deep cosine, by contrast, ranks the methods CB $>$ FA $>$ SB $>$ DFA. CB has the highest deep cosine ($+0.679$), more than double SB's ($+0.322$), yet matches DFA's accuracy (0.360). FA has higher deep cosine than SB ($+0.423$ vs $+0.322$) but 9 percentage points lower accuracy and an order-of-magnitude smaller nudging effect. The cosine ranking does not track the functional ranking on any of the three functional metrics.

The dissociation rules out the interpretation that deep-layer cosine to the BP gradient measures whether the credit signal is useful for training depth. What cosine measures is angular agreement: the credit direction a_l matches the BP gradient direction g_l to varying degrees. CB is designed to produce a credit signal close to g_l in angle, and it succeeds—its deep cosine is the highest of the four. But angular agreement at a single point does not guarantee that an update in that direction reduces loss across the relevant region of state space, nor that the resulting weight updates compose into a working forward computation. SB's credit signal has lower angular agreement with the BP gradient but produces larger functional effects per step (nudging) and over training (loss decrease, accuracy). The two are different quantities, and cosine measures only the first.

This has a direct consequence for the protocol. The depth-utility check in Section 5 cannot be replaced by a per-layer cosine check, even one that conditions on Mode 1 being alleviated and Mode 2 being addressed by per-layer reporting. Whether the trained depth contributes to the network's prediction is a question about functional behavior, not angular agreement, and it requires its own measurement: the architecture-matched frozen-blocks baseline. The standard reporting pair fails at the cosine axis through Modes 1 and 2; it also fails at the accuracy axis by giving no signal that the trained network underperforms an architecture whose deep blocks were never trained at all. The protocol therefore requires three separate diagnostics, not a refinement of the cosine alone.

5 Recommended evaluation protocol

The audit and the failure-mode analysis converge on three checks that together address how the standard reporting pair fails. We state them as a protocol that an evaluator can run on any FA training run; the design rationale follows the structure of Sections 3 and 4.

5.1 Three diagnostic checks

Diagnostic 1 (Scale stability). Compute the maximum per-block residual growth $\rho = \max_l \|h_{l+1}\|/\|h_l\|$ at the end of training. Flag the run if $\rho > 50$. This detects the residual-stream explosion that drives Mode 1a. The threshold is calibrated against the audit: BP and FA on no-LN architectures stay below 10, while DFA on every audited architecture exceeds 90, giving a calibration gap of nearly an order of magnitude.

Diagnostic 2 (Reference validity). Compute the deepest BP reference gradient norm $\|g_L\|$ at the end of training. Flag the run if $\|g_L\| < 10 \cdot \varepsilon$, where ε is the cosine implementation's denominator floor for the training dtype (PyTorch's default is $\varepsilon = 10^{-8}$ for fp32, giving a threshold of 10^{-7} in our setup). At this threshold, the floor contributes more than $\sim 10\%$ of the cosine denominator, making the reported angle a mixture of the true reference direction and the implementation's clamp. In the audit, BP and FA on ResMLP $d = 256, L = 4$ stay above 10^{-6}, while DFA collapses to $\sim 5 \times 10^{-10}$, well below the threshold.

Diagnostic 3 (Depth utility). Compute the gap between the trained model's test accuracy and an architecture-matched frozen-blocks baseline (residual blocks frozen at random initialization, embedding/LN/head trained). Flag the run if the gap is below 2 percentage points. This detects whether the trained depth contributes to the network's prediction, the question that Section 4.3 showed cosine cannot answer even when measured validly.

Verdict logic. A run fails the protocol if either Mode 1 (Diagnostics 1 and 2 both flag) or Depth Utility (Diagnostic 3 flags) is triggered. Diagnostic 1 alone does not constitute a Mode 1 failure—scale growth without reference collapse leaves cosine measurable, as observed for FA on ResMLP. Diagnostic 2 alone is unlikely without Diagnostic 1 by the mechanism in Section 3.1. Diagnostic 3 stands on its own: a run can pass both Mode 1 diagnostics and still fail at depth utility, which is precisely what Section 4.3 establishes.

5.2 Decision-utility ablation

To verify that each diagnostic group contributes to detecting failures the previous strategies miss, we ablate the protocol on two representative cases. Each case is a method-architecture pair where the standard reporting pair gives a misleading verdict, and we ask which reporting strategy first identifies the failure. Case 1 is DFA on ResMLP $d = 256, L = 4$ (the primary audit), where the failure is driven by Mode 1 measurement degeneracy. Case 2 is FA on ResMLP $d = 512, L = 2$ (the representative setting from Section 2), where Mode 1 is not triggered but the trained depth fails to

outperform the frozen baseline. Together they exercise both the Mode 1 check and the depth utility check independently.

Table 4: Decision-utility ablation. Each cell shows the reporting strategy's verdict on the case; ✗ marks a missed failure. Both cases are genuine failures (the trained network underperforms or matches a frozen-blocks baseline).

Reporting strategy	Case 1: DFA, $d = 256, L = 4$	Case 2: FA, $d = 512, L = 2$
S0: acc only	✗ pass (acc 0.306)	✗ pass (acc 0.345)
S1: acc + aggregate Γ	✗ pass ($\Gamma = +0.10$)	✗ pass ($\Gamma = +0.48$)
S2: + Mode 1 check (D1 & D2)	✓ fail	✗ pass (D1, D2 both ok)
S3: + Depth utility (D3)	✓ fail	✓ fail

Table 4 reports the verdicts. The standard reporting pair (S0, S1) misses both failures: DFA reports above-chance accuracy and a positive aggregate cosine, FA does the same on the shallower setting; nothing in the standard pair indicates that either method's trained residual blocks underperform a frozen baseline. Adding the Mode 1 check (S2) catches Case 1 but not Case 2: DFA's Mode 1 is fully triggered ($\|h_L\| \sim 5 \times 10^8$, $\|g_L\| \sim 5 \times 10^{-10}$, both diagnostics flag), while FA's Mode 1 on the shallower setting is not triggered ($\|h_L\|$ stays moderate, $\|g_L\|$ stays above the clamp), so S2 reports FA as passing. The depth utility check (S3) catches Case 2: FA's accuracy of 0.345 is below the frozen baseline of 0.349, flagging the run regardless of Mode 1 status. The full protocol catches both failures.

Each of the three checks catches at least one case the others miss. Appendix B extends this ablation to additional architectures and methods, with consistent results.

5.3 Scope, recommendations, and discussion

Scope. The thresholds are calibrated on supervised image classification with small to medium residual architectures (ResMLP, ViT-Mini, StudentNet) on CIFAR-10 in fp32. On other architecture families, datasets, or precisions (fp16/bf16 raise the effective denominator floor), the threshold values may need recalibration; the measurements themselves remain well-defined wherever a forward pass, a BP reference gradient, and a frozen-blocks baseline can be computed. The Mode 1b mechanism is similarly scoped to terminal-LayerNorm architectures; on architectures without it, Diagnostic 2 functions as a defensive check rather than a primary failure detector. Extending the audit to RL, generative training, or larger-scale NLP is open work.

Cross-architecture footprint. Diagnostic 1 applies anywhere the local loss form $-\langle f_l(h_l), a_l \rangle$ is used without a norm penalty. Diagnostic 2 is informative only on architectures where the BP reference can mechanistically collapse; on no-LN architectures it adds little signal but costs nothing. Diagnostic 3 is architecture-agnostic.

Recommendations. (i) Report raw diagnostic values, not only pass/fail, to allow recalibration. (ii) Report cosine alignment per-layer rather than aggregate. (iii) Treat the protocol as a minimum bar: passing rules out the failures we audited but does not validate the method as a BP alternative.

6 Conclusion

The standard reporting pair for feedback alignment—task accuracy and aggregate cosine alignment to the BP gradient—can fail to indicate that a method has trained the network. We identified two independent ways the cosine half can fail: the BP reference gradient can collapse below the cosine implementation's denominator floor in terminal-LayerNorm residual architectures (Mode 1), and the aggregate can mask layerwise heterogeneity that concentrates credit at a single end of the network (Mode 2). We further showed that even when cosine is measured validly, it does not predict whether trained depth contributes to the network's prediction; this is a separate failure of the accuracy half, requiring an architecture-matched frozen-blocks baseline to detect. The two failures are causally independent, both observationally and under a penalty intervention that alleviates Mode 1 without affecting Mode 2.

The recommended protocol consists of three diagnostic checks—scale stability, reference validity, and depth utility—together with per-layer rather than aggregate cosine reporting. Across the audited architectures and methods, the standard reporting pair gives no signal of failure where our protocol identifies failure. The protocol does not certify a method as universally effective; it rules out the

specific class of silent failures the audit revealed, and leaves the standard pair's interpretation intact when all three diagnostics pass.

References

[1] Mohamed Akrout, Collin Wilson, Peter C. Humphreys, Timothy P. Lillicrap, and Douglas B. Tweed. Deep learning without weight transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. arXiv:1904.05391.

[2] Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[3] Sergey Bartunov, Adam Santoro, Blake A. Richards, Geoffrey E. Hinton, and Timothy P. Lillicrap. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. arXiv:1807.04587.

[4] Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to imagenet. In *International Conference on Machine Learning (ICML)*, 2018. arXiv:1812.11446.

[5] Brian Crafton, Abhinav Parihar, Evan Gebhardt, and Arijit Raychowdhury. Direct feedback alignment with sparse connections for local learning. *Frontiers in Neuroscience*, 2019. doi: 10.3389/fnins.2019.00525. arXiv:1903.02083.

[6] Giorgia Dellaferrera and Gabriel Kreiman. Error-driven input modulation: Solving the credit assignment problem without a backward pass. In *International Conference on Machine Learning (ICML)*, 2022.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2010.11929.

[8] 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*, 2021. doi: 10.3389/fnins.2021.629892.

[9] Geoffrey E. Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022. doi: 10.48550/arXiv.2212.13345.

[10] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[11] Julien Launay, Iacopo Poli, and Florent Krzakala. Principled training of neural networks with direct feedback alignment, 2019.

[12] 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 (NeurIPS)*, 2020.

[13] Timothy P. Lillicrap, Daniel Cownden, Douglas B. Tweed, and Colin J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7(1):13276, 2016. doi: 10.1038/ncomms13276.

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[15] Theodore H. Moskovitz, Ashok Litwin-Kumar, and L. F. Abbott. Feedback alignment in deep convolutional networks, 2018.

[16] Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[17] Maria Refinetti, Stéphane d'Ascoli, Ruben Ohana, and Sebastian Goldt. Align, then memorise: The dynamics of learning with feedback alignment. In *International Conference on Machine Learning (ICML)*, 2021.

[18] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. doi: 10.1109/TPAMI.2022.3206148. arXiv:2105.03404.

[19] James C. R. Whittington and Rafal Bogacz. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural Computation*, 2017. doi: 10.1162/NECO_a_00949.

[20] Will Xiao, Honglin Chen, Qianli Liao, and Tomaso Poggio. Biologically-plausible learning algorithms can scale to large datasets. In *International Conference on Learning Representations (ICLR)*, 2018. arXiv:1811.03567.

[21] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning (ICML)*, 2020. arXiv:2002.04745.

A FA + penalty on ResMLP: no dose response

For completeness, we apply the same penalty intervention to FA on ResMLP $d = 256, L = 4$, three seeds, 30 epochs:

λ	Test acc	Deep cos	$\|h_L\|$	$\|g_L\|$
0	0.372 ± 0.007	$+0.325 \pm 0.015$	$\sim 9 \times 10^4$	$\sim 2 \times 10^{-6}$
10^{-4}	0.377 ± 0.006	$+0.298 \pm 0.031$	$\sim 9 \times 10^3$	$\sim 1 \times 10^{-5}$
10^{-2}	0.369 ± 0.003	$+0.423 \pm 0.006$	$\sim 1 \times 10^4$	$\sim 2 \times 10^{-5}$

FA on this architecture has mild Mode 1 even at $\lambda = 0$: the residual norm $\|h_L\| \sim 9 \times 10^4$ is well below DFA's $\sim 4 \times 10^8$, and the BP reference gradient $\|g_L\| \sim 2 \times 10^{-6}$ is already two orders of magnitude above the cosine clamp. The penalty has no Mode 1 to alleviate; accordingly, accuracy and deep cosine remain in narrow windows (0.37–0.38 and $+0.30$–$+0.42$ respectively) across all three λ values. This is consistent with the cross-architecture finding (Section 4.1) that FA's Mode 1 severity depends on the architecture, and the intervention is most informative where Mode 1 is fully developed.

B Extended decision-utility ablation

B.1 Protocol verdict on CIFAR-10

We extend the decision-utility ablation to additional architecture-method pairs to verify that the Section 5.2 pattern (each diagnostic group catching at least one case the others miss) is not specific to the two cases shown in the main text.

Table 5: Extended decision-utility ablation. Each cell shows the reporting strategy's verdict; ✗ marks a missed failure (silently passing a method whose trained blocks underperform a frozen-blocks baseline). Healthy reference cases (BP) appear with ✓ throughout. Cases include the two main-text cases (Section 5.2) plus three additional method-architecture pairs.

Reporting strategy	DFA, RM-d256-L4	FA, RM-d512-L2	DFA, ViT-Mini	FA, ViT-Mini	BP, RM-d256-L4
S0: acc only	✗	✗	✗	✗	✓
S1: acc + Γ	✗	✗	✗	✗	✓
S2: + Mode 1 (D1 & D2)	✓	✗	✓	✓	✓
S3: + Depth utility (D3)	✓	✓	✓	✓	✓

The pattern from the main text holds: the standard pair (S0, S1) silently passes all four genuine failure cases; the Mode 1 check (S2) catches the three cases where measurement is degenerate but does not catch the case where measurement is valid (FA on $d = 512, L = 2$); the depth utility check (S3) closes the remaining gap. The healthy case (BP on the primary audit) passes all reporting strategies, confirming that the protocol does not flag working methods. ViT-Mini provides the strongest depth-utility evidence in the extended set: the frozen-blocks baseline reaches 0.570 ± 0.003 on this architecture, while DFA and FA reach 0.256 and 0.163 respectively—trained transformer blocks underperform random blocks by more than 30 percentage points.

StudentNet, the no-LN architecture used in Section 4.1 as a Mode 1b control, is excluded from this ablation: its frozen-blocks baseline reaches 0.908 ± 0.009 on the synthetic task, exceeding even fully trained BP (0.796). The task is solvable by a linear probe on random features, which makes Diagnostic 3 uninformative regardless of the training method. We use StudentNet only as a mechanism control for Mode 1b's architectural scope and not as a depth-utility test case.

B.2 Protocol verdict on CIFAR-100

To check whether the protocol's verdict pattern depends on the audited dataset, we replicate the primary-audit setup on CIFAR-100: ResMLP $d = 256, L = 4$, three seeds, 100 epochs, with the same optimizer, schedule, and training implementation (Section 2). We also compute an architecture-matched frozen-blocks baseline on CIFAR-100, which reaches 0.178 ± 0.001—close to the linear-probe ceiling on CIFAR-100 pixels (0.177).

Table 6: Protocol verdict on CIFAR-100 (ResMLP $d = 256, L = 4$, three seeds, 100 epochs). The pattern from the primary CIFAR-10 audit is reproduced: BP passes; DFA fails Mode 1 and Depth Utility; FA passes Mode 1 (growth $15.6\times$, $\|g_L\| = 1.3 \times 10^{-6}$ above clamp) but fails Depth Utility (-4.5 pp vs frozen).

Method	Test acc	D1 (Growth)	D2 ($\|g_L\|$)	D3 (vs Frozen)	Verdict
BP	0.321 ± 0.002	$1.0\times$ ✓	9.5×10^{-4} ✓	$+14.3$ pp ✓	pass
FA	0.133 ± 0.013	$15.6\times$ ✓	1.3×10^{-6} ✓	-4.5 pp ✗	fail (D3)
DFA	0.088 ± 0.001	$1004\times$ ✗	9.1×10^{-9} ✗	-9.0 pp ✗	fail (D1+D2+D3)

The CIFAR-100 verdict pattern matches CIFAR-10 exactly. BP passes all three diagnostics; DFA triggers Mode 1 (both D1 and D2) and depth utility, with the same residual-scale-explosion-to-reference-collapse mechanism developed in Section 3; FA's Mode 1 stays mild (growth below $50\times$, $\|g_L\|$ above the clamp) but its trained depth fails to outperform random blocks. FA on CIFAR-100 is a second instance of the depth-utility-only failure pattern (FA on $d = 512, L = 2$ ResMLP being the first in Section 5.2), strengthening the case that Diagnostic 3 catches a class of failures distinct from those caught by Mode 1 alone. The protocol verdict is dataset-invariant on this architecture.

C Controls for the penalty intervention

C.1 BP control: penalty does not artificially lift cosine

A potential alternative explanation for the deep-cosine recovery on DFA at $\lambda = 10^{-2}$ (Section 4.2) is that the penalty itself shifts cosine measurement statistics independent of any change in credit quality. We rule this out with a BP control: BP under the same penalty schedule reaches 0.585 ± 0.001 accuracy at $\lambda = 0$ and 0.532 ± 0.007 at $\lambda = 10^{-2}$, a capacity cost of 5.3 percentage points (Figure 6, right panel). BP's deep cosine remains ≈ 1.0 throughout: the penalty does not artificially lift cosine on a method whose credit is already exact. The penalty is not a free intervention—it costs accuracy on every method we tested—but BP+penalty still exceeds the frozen-blocks baseline by 18.3 percentage points, so the penalty does not turn working methods into broken ones.

C.2 Fresh-B null calibration

A second alternative is that the recovered deep cosine reflects trivial adaptation to the fixed feedback matrices B_l rather than an improvement in credit quality. We rule this out with a fresh-B null calibration: holding the trained DFA network at the $\lambda = 10^{-2}$ checkpoint (seed 42) fixed and replacing each B_l with a freshly drawn random matrix gives deep cosine -0.005 ± 0.018 over 20 draws, against the trained-B value of $+0.166$ on the same checkpoint—a 9.3σ separation. The recovered cosine is well above the noise floor and is specific to the matrices used during training.

D Related work

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: **[TODO]**

   Justification: **[TODO]**

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: **[TODO]**

    Justification: **[TODO]**

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.