BETTER ALIGNED, TRAINS WORSE: INTERVENING ON GRADIENT ALIGNMENT IN LOCAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Local learning rules replace backpropagation's transposed weights with a feedback matrix and are routinely checked by per-layer gradient alignment. A better-aligned matrix is read as one whose credit trains better, a premise tested so far mostly by correlation across runs. We show that alignment can prefer the feedback matrix that trains worse, and that following its preference lowers accuracy. We study the weight mirror, which learns its matrix from random probes. We run it at the published mirror learning rate with few probes and at a tuned setting with a larger rate and more probes. Each run also carries the other setting's matrix without using it, so both matrices are measured on the same forward weights. Measured this way, in a residual network with and without shortcuts, alignment prefers the matrix of the setting that trains worse. We then fork tuned runs. From one snapshot, one branch keeps its matrix and the other lets the preferred matrix compute the weight updates. That branch ends with lower accuracy in each seed, by 1.12 percentage points with shortcuts and 2.73 without. Alignment scores for credit assignment should be validated by intervention, not by correlation.

1 INTRODUCTION

Backpropagation computes each layer's credit, the error signal that sets the layer's weight update, by sending the output error backwards through the transposed forward weights (Rumelhart et al., 1986). Local credit-assignment rules replace this weight transport with a feedback matrix that each layer holds on its own (Lillicrap et al., 2020). The matrix can be fixed and random, as in feedback alignment (Lillicrap et al., 2016), project the output error or the target directly to each layer (Nøkland, 2016; Frenkel et al., 2021), or copy the signs of the forward weights, as in sign symmetry (Liao et al., 2016; Xiao et al., 2019). The weight mirror of Akrout et al. (2019) instead learns the matrix from random probes so that it tracks the forward weights. A rule is checked by its per-layer gradient alignment, the angle between the credit it delivers to a layer and the credit the transposed weights would deliver.

Alignment is read both as a pass mark and as a ranking. Akrout et al. (2019) report that "weight mirrors kept these angles $< 25^\circ$ in all layers" and prescribe that "if mirroring ever fails to yield small enough angles, we can simply do more mirroring". Roy et al. (2026) find their rule's angles 15 to 30 degrees below those of direct feedback alignment and conclude that its "updates are better aligned with the true BP [backpropagation] direction". Lance & Kieu (2026) read the misalignment of random feedback in convolutional layers as a partial explanation of its accuracy gap to sign symmetry. Toolkits and other rules rank by alignment in the same way (Jiménez Sanfiz & Akrout, 2021; Li & Benna, 2024; Boeshertz et al., 2026), and alignment is read as evidence that learning happens at all (Launay et al., 2020). None of these works states a selection rule or claims that alignment suffices for accuracy, and none says how large an alignment difference must be before it orders two methods.

A ranking by alignment assumes that the better-aligned of two feedback matrices delivers the credit that trains better. This premise has been probed by correlation (Chu & Bacho, 2023) and by one intervention that sets alignment once, at initialisation (Cheon et al., 2025). A correlation across runs cannot settle it, because each matrix is measured on the weights its own run trained. Two settings of the weight mirror show the problem. The low-budget setting uses the published mirror learning rate with one small batch of probes, and the tuned setting uses a larger rate with more probes. At their

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

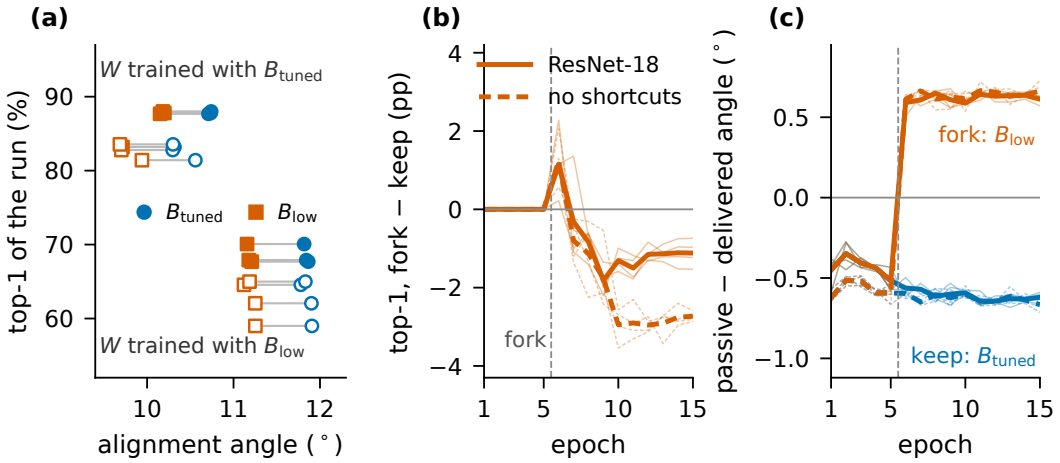


Figure 1: **Alignment prefers the matrix that trains worse, and following it lowers accuracy.** (a) Each marker pair is one run's two matrices measured on its W: B_{low} (squares) is better aligned than B_{tuned} (circles), yet runs trained by B_{low} are less accurate (open: no shortcuts). (b, c) After epoch 5, tuned runs hand the credit to B_{low} (fork) or keep B_{tuned} (keep); thin lines are seeds.

own weights the two are aligned alike, at angles of $10.81°$ and $10.95°$, while the tuned setting trains 16.36 percentage points (pp) better (Section 3).

We therefore separate each matrix from the weights it is measured on. In each run we also carry the other setting's matrix, which receives its own mirror updates but never delivers credit. We call this matrix *passive*. Both matrices are then measured on the same forward weights. To test the resulting ranking by intervention, we *fork* the run. From one snapshot, one branch keeps its matrix and the other hands the credit to the matrix that alignment prefers, with data order, augmentation and probes shared.

On shared weights, alignment prefers the low-budget matrix. It does so in ResNet-18 with and without its shortcuts, whichever setting trained the weights (Figure 1a). Its margin is more than $80\times$ the difference between two matrices of the same setting, which we measure in control runs that carry both. Handing the credit to the low-budget matrix lowers final accuracy in each forked seed, by 1.12 pp with shortcuts and 2.73 pp without (Figure 1b).

More mirroring, which Akrout et al. (2019) prescribe for angles that are too large, recovers only a small part of the accuracy gap when applied to the low-budget setting. We conclude that alignment can rank two feedback matrices backwards and that a choice made by it can lower accuracy. Before an alignment score chooses between feedback matrices, an intervention such as the fork should show that the better-aligned matrix trains better.

Contributions.

- In two ResNet-18 configurations, alignment on shared weights ranks two weight-mirror matrices backwards, and following it lowers accuracy in each forked seed (Sections 4–5).
- At their own weights, two settings that train up to 19.11 pp apart differ in alignment by at most $0.82°$ across nine measurements (Section 3).
- More mirroring, prescribed for large angles, leaves most of the accuracy gap in place when applied to the low-budget setting (Section 6).
- The fork, which continues one run from a shared snapshot with a different feedback matrix, tests an alignment-based choice by intervention (Section 5).

2 THE WEIGHT MIRROR AND PER-LAYER ALIGNMENT

The credit at a layer is the error signal that the layer multiplies by its own input to form its weight update. Backpropagation computes it by multiplying the error arriving from the layer above by $W^\top$, the transpose of the forward weights. A transport-free rule never reuses $W^\top$ and multiplies instead by the transpose of a feedback matrix B that the layer holds locally. We say that B delivers the credit. The weight mirror (Akrout et al., 2019) learns B from noise. After each optimiser step, layer l draws k minibatches of N Gaussian probes $n_j \sim \mathcal{N}(0, I)$ and passes them through its updated forward weights. It forms the estimate $\hat{H}_l = \frac{1}{kN} \sum_j (W_l n_j)\, n_j^\top$, whose expectation is W_l up to zero-padding effects at the borders of a convolution. The feedback matrix moves towards this estimate,

$$B_l \leftarrow (1 - \eta_B)\, B_l + \eta_B\, \|W_l\|_F\, \hat{H}_l / \|\hat{H}_l\|_F, \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm. Apart from the norm of W_l, the rule never uses the stored values of W.

The rule has two knobs, the probe budget kN and the mirror learning rate η_B. A large rate makes B follow each new, noisy estimate closely, while a small rate averages many estimates but lags behind W as it changes. We compare two settings, whose values Table 1 lists. The low-budget setting uses the published η_B of Akrout et al. (2019) with one minibatch of probes, and the tuned setting a four times larger η_B with four minibatches, the most accurate cell of the grid in Table 4. Both budgets lie below the 2048 probes per mirror update of the original work. We write B_{tuned} and B_{low} for the two settings' feedback matrices.

We measure per-layer alignment at each convolution inside the network's residual blocks. At each such layer we compare the credit the rule computes with $B^\top$ against the credit $W^\top$ would compute from the same incoming error. We take their cosine over fixed batches of test images, flattened into one vector, and average it over layers and epochs. This is the global form of the batch average, as opposed to the per-sample form, which averages per-sample cosines. We report the average as an angle, $\theta = \arccos(\cos)$, so a smaller angle means better alignment. An alignment difference "A minus B" is B's angle minus A's, positive when A is better aligned. Because both credits start from the same error, this measure scores a single feedback hop. The works cited in Section 1 report per-layer alignment either for one hop, as the angle between B and $W^\top$ (Akrout et al., 2019; Jiménez Sanfiz & Akrout, 2021), or for the whole backward pass (Lillicrap et al., 2016; Roy et al., 2026), which also compounds the mismatch of the hops above the layer.

We train ResNet-18 (He et al., 2016) without normalisation layers on CIFAR-10 (Krizhevsky, 2009) for 15 epochs with AdamW (Loshchilov & Hutter, 2019). Each setting's learning rate is selected from a grid, and each contrast is paired by seed (Appendix A). We use two independent implementations. The sequential trainer takes an ordinary backward sweep with B on each hop and runs all experiments from Section 4 on. In it, backpropagation reaches 88.79% top-1 accuracy in this schedule (one run). The iterative trainer computes all layers' credits at once and refines them over a truncated number of sweeps. It produces Table 1, and Appendix B shows that the result of Section 3 holds with the exact number of sweeps and in the sequential trainer. We write $\pm$ for the across-seed standard deviation and $[\cdot, \cdot]$ for a two-sided 95% confidence interval (CI) on seed-paired differences.

3 TWO SETTINGS THAT TRAIN FAR APART ALIGN ALIKE

Measured at each setting's own weights, as the cited works measure it, alignment does not separate the two settings. In Table 1 the tuned setting trains 16.36 pp better and wins each seed pair. The two angles differ by $0.14°$. Measured per layer in credit space, against the bound Akrout et al. (2019) report, both settings stay inside that $25°$ range at each layer in the sequential trainer (Appendix B).

The accuracy gap holds at a shared learning rate and seed by seed. In Table 1 the two settings select the same learning rate, and the tuned setting wins even the closest seed pair by more than 10 pp. Within a setting, alignment tracks accuracy. Across eighteen further seeds of the low-budget setting in the sequential trainer, it ranks the seeds by accuracy with a Spearman correlation of 0.74, although their alignments span less than the difference between the two settings.

We collect nine measurements of the two settings in Figure 2a, and alignment stays nearly equal across all of them. They vary the trainer, the dataset, the shortcuts, the depth and the schedule, and

Table 1: **Two settings of the weight mirror are aligned alike and train far apart.** ResNet-18 on CIFAR-10, iterative trainer, 15 epochs, mean ± across-seed standard deviation over n seeds. The angle is the arccosine of the mean cosine. Sign symmetry, far less aligned, trains between the two settings.

setting	η_B	probes	n	cosine	angle (°)	top-1 (%)
tuned	0.4	512	18	0.98224 ± 0.00019	10.81	78.38 ± 0.72
low budget	0.1	128	18	0.98179 ± 0.00020	10.95	62.02 ± 7.38
sign symmetry	–	–	18	0.66563 ± 0.00139	48.27	66.23 ± 0.99

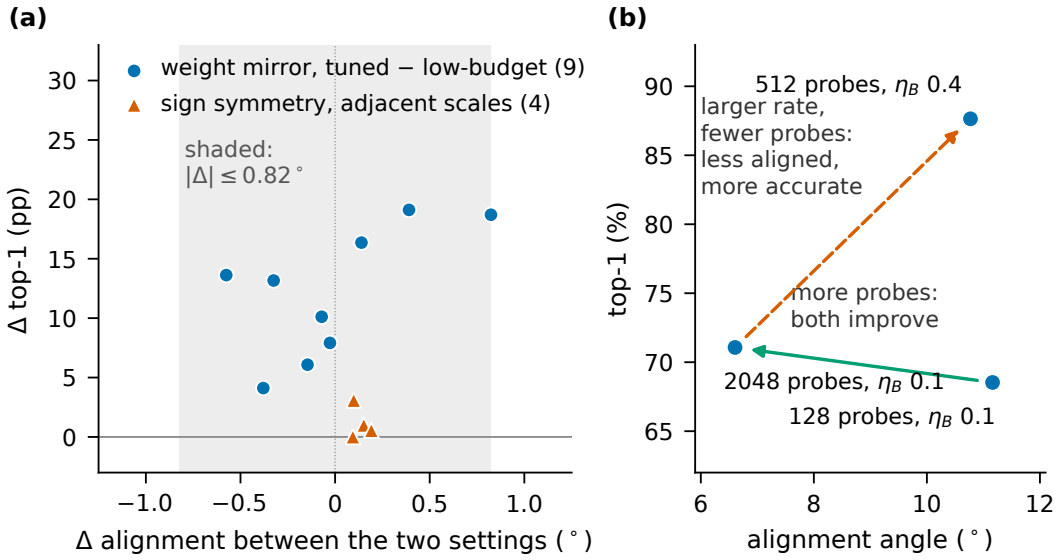


Figure 2: **Alignment orders the settings along one knob and backwards across the other.** (a) Cross-setting comparisons at the settings' own weights: the alignment difference against the paired top-1 difference, for the nine weight-mirror measurements (appendix Table 6) and the four pairs of neighbouring feedback scales of sign symmetry, larger minus smaller (appendix Table 8). (b) Three mirror settings in the sequential trainer. Raising the probe budget at a fixed mirror learning rate improves alignment and accuracy together; the larger rate with fewer probes is less aligned and trains far better.

include two smaller networks trained for longer, a 10-layer residual network and the 9-layer chain left when its shortcuts are removed (Appendix B). The alignment difference stays within $0.82°$. The tuned setting still trains 4.11 to 19.11 pp better. At this size even the sign of an alignment difference between two runs can change with the form of the batch average (Appendix B).

The same failure appears along a knob of sign symmetry, the scale of its feedback. The rule sets $B = \text{sign}(W) \odot |F|$ elementwise, where a fixed random matrix F sets the feedback magnitudes, and multiplying F by a scale gives the knob. At fixed weights, rescaling B leaves each cosine nearly unchanged, since only the residual shortcuts break the invariance (Appendix C). An alignment difference between two scales therefore arises mainly through the weights each scale trains. We trained five scales on ResNet-18 in the sequential trainer, with eighteen seeds each. Two pairs of neighbouring scales differ in alignment by about $0.1°$, neither with a CI that excludes zero. Yet one pair trains 3.09 pp apart and the other 0.01 pp apart (Appendix C).

Across rules, alignment differences as large as those quoted in Section 1 can also fail to order accuracy. On the 9-layer chain both mirror settings are about $35°$ better aligned than sign symmetry, yet sign symmetry trains between them (Table 2). With the shortcuts restored it trains below both once one diverged run is replaced (Appendix G). Whether a difference this large orders two rules therefore depends on the network.

Table 2: **Two weight-mirror settings, both far better aligned than sign symmetry, fall on opposite sides of it in accuracy.** 9-layer chain, CIFAR-10, sequential trainer, 150 epochs, $n = 6$ seeds per setting, each at its own selected learning rate. The worst layer is the maximum angle over layers, epochs and seeds; the mean angle is the arccosine of the mean cosine over sixteen test batches.

setting	worst layer (°)	mean angle (°)	top-1 (%)
weight mirror, tuned	21.08	10.27	92.15
weight mirror, low-budget	21.48	10.20	82.04
sign symmetry	47.65	44.92	89.17

Table 3: **On shared weights alignment orders the two feedback matrices backwards, and handing the credit to the matrix it prefers lowers accuracy.** Sequential trainer, CIFAR-10, 15 epochs, ResNet-18 with and without its shortcuts. (a) Each run trains W with one setting and carries the other setting's matrix passively; angles averaged over all fifteen epochs; Δ is tuned minus low-budget alignment, negative when B_{low} is better aligned. (b) From one epoch-5 state, the fork branch hands the credit to the passive matrix and the keep branch continues unchanged; Δ top-1 is fork minus keep; *seeds* counts the seeds with the mean's sign.

(a) One W, both feedback matrices read: angle to the credit $W^\top$ would deliver

configuration	W trained by	B_{tuned} (°)	B_{low} (°)	Δ (°)	95% CI	top-1 (%)
ResNet-18	tuned	10.73	10.18	-0.55	$[-0.58, -0.52]$	87.82 ± 0.16
	low budget	11.84	11.18	-0.66	$[-0.67, -0.65]$	68.55 ± 1.32
no shortcuts	tuned	10.37	9.76	-0.61	$[-0.63, -0.58]$	82.73 ± 0.94
	low budget	11.85	11.20	-0.65	$[-0.66, -0.64]$	62.65 ± 2.76

(b) The fork: from epoch 6 the passive matrix delivers the credit, $n = 4$ per row

configuration	handed over	Δ top-1 (pp)	95% CI	seeds
ResNet-18	B_{low} to the tuned run	-1.12	$[-1.66, -0.58]$	4/4
	B_{tuned} to the low-budget run	$+1.62$	$[-2.25, +5.49]$	3/4
no shortcuts	B_{low} to the tuned run	-2.73	$[-2.93, -2.53]$	4/4
	B_{tuned} to the low-budget run	$+5.26$	$[+1.02, +9.50]$	4/4

4 On shared weights, alignment prefers the worse-training matrix

Measured on the same forward weights, alignment prefers B_{low} in both configurations we tested, whichever setting trained the weights. The own-weight differences of Section 3 compare matrices measured on different weights after different training histories, so the runs can move them as much as the matrices can. To remove the difference between the runs, each run trains W with one setting's matrix and carries the other setting's matrix passively. The passive matrix starts from the same random initial matrix and receives its own mirror updates throughout, but it never delivers credit. We measure both matrices against the one W on the same test batches. The configurations are ResNet-18 and the same network without its residual shortcuts, both in the sequential trainer, with three or four runs for each setting that trains the weights.

On ResNet-18, B_{low} is better aligned by $0.55°$ on the tuned runs' weights and by $0.66°$ on the low-budget runs' weights, while the tuned runs train 19.27 pp better (Table 3a). Without shortcuts the pattern repeats with a larger accuracy gap. The ranking holds in each of the fourteen runs, at the final epoch as well as on average (Appendix D).

These within-run differences are more than $80\times$ their own noise. In control runs that carry a second, passive tuned matrix, two matrices of the same setting differ by $0.006°$ on ResNet-18 and $0.003°$ without shortcuts. Within one run the two matrices share the weights, the test batches and the form of the average.

In these runs, measuring each setting at its own weights hides the difference between the matrices, because the difference between the runs is larger and points the other way. Write $\theta(B; W_s)$ for the angle of matrix B measured on the weights W_s of the run that setting s trains (t tuned, l low budget).

The own-weight difference then splits exactly into a between-run term and a between-matrix term, each positive when it makes the tuned setting look better aligned:

$$\begin{aligned}\theta(B_{\text{low}}; W_{\text{l}}) - \theta(B_{\text{tuned}}; W_{\text{t}}) = [\,\theta(B_{\text{tuned}}; W_{\text{l}}) - \theta(B_{\text{tuned}}; W_{\text{t}})\,] \\ + [\,\theta(B_{\text{low}}; W_{\text{l}}) - \theta(B_{\text{tuned}}; W_{\text{l}})\,].\end{aligned} \tag{2}$$

In these runs on ResNet-18, the own-weight difference of $+0.45°$ splits into $+1.11°$ between runs and $-0.66°$ between matrices. The terms have the same signs without shortcuts (Appendix D). The between-run term compares the tuned matrix on its own run's weights with the tuned matrix carried passively in the low-budget run. It therefore mixes the effect of different weights with that of the different path the matrix followed in each run.

5 Following alignment's preference lowers accuracy

Handing the credit to B_{low}, the matrix that alignment prefers, lowers final accuracy in both configurations. We fork tuned runs after epoch 5. On the fork branch the passive B_{low}, still updated by its own low-budget mirror, delivers the credit from then on, while the keep branch continues with B_{tuned}. Both branches continue in one process from one snapshot at the tuned run's learning rate, so data order, augmentation and probes coincide and only the delivering matrix differs. Final top-1 accuracy falls by 1.12 pp on ResNet-18 and by 2.73 pp without shortcuts, in each forked seed (Table 3b). The fork branch falls behind (Figure 1b), while B_{low} remains the better aligned of the two (Figure 1c).

Forking alone costs almost nothing. When we restore the snapshot without a hand-over, final accuracy moves by at most 0.29 pp relative to an uninterrupted run, so the fork isolates a cost of following alignment at the level of the feedback matrix. The reverse hand-over gives the low-budget run B_{tuned} and changes accuracy in the opposite direction. Without shortcuts it raises final accuracy by 5.26 pp in each seed. On ResNet-18 its CI includes zero (Table 3b).

Each forked seed also loses under a longer schedule and an earlier fork (Appendix D). In separate 50-epoch runs on ResNet-18 the fork branch loses in each of six seeds ($p = 0.0312$, exact sign test), and forking the shorter runs after epoch 2 loses in each of four seeds. One forked run in each variant destabilises, and the CI of each mean includes zero. In the longer runs that forked run diverges to chance while the alignment of both matrices improves.

6 More mirroring leaves most of the gap

Applied to the low-budget setting, more mirroring recovers only 2.54 pp of the gap. Akrout et al. (2019) prescribe it for angles that are too large, which these are not. Raising the low-budget setting's probe budget sixteenfold, to their 2048 probes, still leaves it 16.57 pp behind the tuned setting over six seed pairs (Appendix F).

Along this budget knob alignment orders the settings correctly. More probes improve alignment and accuracy together (Figure 2b). Across the other knob it orders them backwards. The 2048-probe setting is the best aligned of the three, by $4.17°$ over the tuned setting, which still trains better.

With two or more minibatches of probes, the gap tracks the mirror learning rate (Table 4). Raising η_B fourfold gains about 16 pp at two or four minibatches. With one minibatch, where the low-budget setting sits, the larger rates instead lower mean accuracy and make training unstable across seeds.

Kunin et al. (2020) trace the weight mirror's sensitivity to its hyperparameters to the stability of its estimate rather than to misalignment, and our grid is consistent with their account. A simpler account holds that an alignment averaged over a batch sees a matrix's bias but not its variance. On the 9-layer chain, zero-mean noise added to the delivering matrix, redrawn each step at 10% of its norm, lowers alignment by more than any own-weight difference of Section 3. Yet it costs at most 0.21 pp (upper 95% bound, six seeds), which suggests that this account is incomplete.

Setting a weight mirror's alignment directly, by rotating each layer's credit towards or away from the credit $W^{\top}$ would deliver, does move accuracy, but the gain per unit of alignment varies $15.3\times$ across three configurations (Appendix E).

Table 4: **With two or more probe minibatches, accuracy tracks the mirror learning rate.** Top-1 accuracy (%) over the mirror learning rate η_B and the number k of probe minibatches of 128, mean $\pm$ across-seed standard deviation over six seeds. ResNet-18, sequential trainer, 15 epochs, each row at its own selected learning rate. The low-budget setting is the top-left cell, the tuned setting the bottom-right.

η_B	$k = 1$	$k = 2$	$k = 4$
0.1	68.53 ± 1.03	70.79 ± 0.75	71.45 ± 0.85
0.2	65.00 ± 19.04	78.33 ± 1.83	79.74 ± 0.68
0.4	63.00 ± 29.03	86.74 ± 0.52	87.64 ± 0.32

7 Related work

Accuracy without alignment. Meta-learned feedback reaches the accuracy of a backpropagation-trained baseline with often weak per-layer alignment (Lindsey & Litwin-Kumar, 2020), and a broadcast global error trains with updates matched to the gradient only in sign (Clark et al., 2021). A convergence rate for random feedback in a two-layer model under gradient flow contains no alignment term (Song et al., 2021), and analytic accounts describe when random feedback suffices (Garg & Vempala, 2022; Refinetti et al., 2021).

Alignment without accuracy. Chu & Bacho (2023) find no trend between alignment and accuracy across one rule's initialisations in multilayer perceptrons, and Moskovitz et al. (2018) note comparable angles beside unequal accuracy. These comparisons are across runs. We compare two settings whose alignment is nearly equal, measure both matrices on shared weights, and fork the feedback matrix. The nearest interventional prior, Cheon et al. (2025), sets the angle between B and $W^\top$ once, at initialisation, over four angles. They find that alignment helps but saturates, consistent with our finding that exact alignment adds under a quarter of a point on the 9-layer chain (Appendix E). Their alignment is never re-imposed, whereas we rotate the credit itself at each step.

Audits and uses of alignment. Hao et al. (2026) audit the same reporting pair, accuracy and gradient cosine, and ask whether a method trained the network at all. We adopt their safeguards: no normalisation layer, a trainer that matches automatic differentiation when $B = W$, per-layer measurement, and a depth-utility check. That check compares a trained network with one whose hidden blocks stay frozen, and both our settings pass it (Appendix H). We then ask whether alignment can choose between two settings of a method that did train. Several rules rank themselves by alignment or optimise it directly (Li & Benna, 2024; Roy et al., 2026; Caillon et al., 2026; Lansdell et al., 2020), and a toolkit ships it as a standard metric (Jiménez Sanfiz & Akrout, 2021). Our measurements bear on these works where alignment is used to choose between feedback matrices. Local losses, forward-only training and energy-based learning change more than the feedback path and lie outside our scope (Belilovsky et al., 2019; Nøkland & Eidnes, 2019; Jaderberg et al., 2017; Dellaferrera & Kreiman, 2022; Hinton, 2022; Scellier & Bengio, 2017).

8 Limitations

The findings are established for the weight mirror and sign symmetry on small-image convolutional networks without normalisation, mostly at 15 epochs. Product feedback alignment (Li & Benna, 2024), the direct feedback method of Caillon et al. (2026) and ImageNet scale lie outside what we measured. Feedback alignment with fixed random feedback is excluded because, on its source's own network and recipe, it trains far below its published accuracy in two independent implementations. The shared-weight measurement and the fork rest on one pair of settings in two configurations of one architecture family. They use three or four runs per setting and four forked seeds per direction, and they measure only the global form of the credit cosine, not the angle between B and $W^\top$. They show that alignment can rank two feedback matrices backwards, not how often it does. The fork isolates a cost at the level of the feedback matrix, at the tuned run's learning rate, and does not account for the pair's accuracy gap. We trace that gap to the mirror learning rate and an unstable

one-minibatch estimate, without identifying a quantity through which they act. The accuracy value of a unit of alignment is measured only with a rotation that uses $W^\top$.

9 CONCLUSION

Per-layer gradient alignment can rank two feedback matrices backwards. On shared weights it prefers the matrix that trains worse in both configurations we forked. Following that preference lowers accuracy. In both configurations, measuring each setting at its own weights hides this reversal, because the difference between the runs sets the sign. Which property of a feedback matrix, readable before training ends, predicts which of two settings will train better remains open. Until such a property is known, we recommend choosing between settings of a rule on held-out accuracy, and using alignment to rank feedback matrices only where a fork or a similar intervention confirms the ranking.

AI USE STATEMENT

AI assistants based on large language models were used for part of the execution of this work, under the authors' direction: implementing the learning rules and the training and measurement code. All research decisions were made by the authors, who verified every AI-assisted output and every reported result. The authors take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

Section 2 and Appendix A give the architectures, optimiser, schedules, learning-rate selection, seed pairing and the mirror update as run in both trainers. Appendix D states the shared-weight measurement, the in-process fork and the rotation instrument exactly, since the central results act on them. The anonymized code supplement contains the rule implementations, both trainers, the instruments, the scoring scripts and one script per table and figure, each of which reads the per-run result files.

REFERENCES

Mohamed Akrout, Collin Wilson, Peter C. Humphreys, Timothy Lillicrap, and Douglas B. Tweed. Deep learning without weight transport. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to ImageNet. In *International Conference on Machine Learning*, 2019.

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

Paul Caillon, Erwan Fagnou, Blaise Delattre, and Alexandre Allauzen. Scaling direct feedback learning with Jacobian alignment guarantees. In *International Conference on Learning Representations*, 2026.

Jeonghwan Cheon, Jaehyuk Bae, and Se-Bum Paik. One-time soft alignment enables resilient learning without weight transport. *arXiv preprint arXiv:2505.20892*, 2025.

Dominique Chu and Florian Bacho. Random feedback alignment algorithms to train neural networks: Why do they align? *arXiv preprint arXiv:2306.02325*, 2023.

David G. Clark, L. F. Abbott, and SueYeon Chung. Credit assignment through broadcasting a global error vector. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

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

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.

Shivam Garg and Santosh S. Vempala. How and when random feedback works: A case study of low-rank matrix factorization. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Yuren Hao, Xiang Wan, and ChengXiang Zhai. What accuracy and gradient cosine miss: Evaluating feedback alignment via scale stability, reference validity, and depth utility. *arXiv preprint arXiv:2606.21126*, 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022.

Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. In *International Conference on Machine Learning*, 2017.

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

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

Daniel Kunin, Aran Nayebi, Javier Sagastuy-Breña, Surya Ganguli, Jonathan M. Bloom, and Daniel Yamins. Two routes to scalable credit assignment without weight symmetry. In *International Conference on Machine Learning*, 2020.

Jake Lance and Larry Kieu. Biological plausibility and representational alignment of feedback alignment in convolutional networks. *arXiv preprint arXiv:2605.08564*, 2026.

Benjamin James Lansdell, Prashanth Ravi Prakash, and Konrad Paul Kording. Learning to solve the credit assignment problem. In *International Conference on Learning Representations*, 2020.

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, 2020.

Ji-An Li and Marcus K. Benna. Deep learning without weight symmetry. *arXiv preprint arXiv:2405.20594*, 2024.

Qianli Liao, Joel Z. Leibo, and Tomaso Poggio. How important is weight symmetry in backpropagation? In *AAAI Conference on Artificial Intelligence*, 2016.

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.

Timothy P. Lillicrap, Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21:335–346, 2020.

Jack Lindsey and Ashok Litwin-Kumar. Learning to learn with feedback and local plasticity. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Theodore H. Moskovitz, Ashok Litwin-Kumar, and L. F. Abbott. Feedback alignment in deep convolutional networks. *arXiv preprint arXiv:1812.06488*, 2018.

Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Arild Nøkland and Lars Hiller Eidnes. Training neural networks with local error signals. In *International Conference on Machine Learning*, 2019.

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*, 2021.

Arani Roy, Marco P. Apolinario, Shristi Das Biswas, and Kaushik Roy. Feedback alignment meets low-rank manifolds: A structured recipe for local learning. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2026.

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

Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in Computational Neuroscience*, 11:24, 2017.

Ganlin Song, Ruitu Xu, and John Lafferty. Convergence and alignment of gradient descent with random backpropagation weights. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Will Xiao, Honglin Chen, Qianli Liao, and Tomaso Poggio. Biologically-plausible learning algorithms can scale to large datasets. In *International Conference on Learning Representations*, 2019.

A PROTOCOL AND THE MIRROR UPDATE AS RUN

Trainers. The iterative trainer updates the credit at every layer from its neighbours' current values, all layers at once, and repeats the sweep K times. At K equal to the credit depth of the network, the number of sweeps the error needs to reach every layer (17 for ResNet-18), it reproduces a depth-ordered backward sweep exactly. Rows 1, 3, 4 and 5 of Table 6 use $K = 9$, and row 2 uses the exact $K = 17$. The sequential trainer takes an explicit depth-ordered backward sweep and substitutes B on each hop between layers. With $B = W$ its sweep matches automatic differentiation of the same loss to a relative error of 2.2×10^{-15}. At $B \neq W$ its feedback hop matches an independent reference implementation to a relative error of 2.0×10^{-14}. The two trainers share only the data loader.

The mirror update as run. Both trainers implement Equation 1 with the same steps. The update runs once after every optimiser step, on the weights that step has just produced, and never changes W. For each layer it draws k independent minibatches of $N = 128$ probes, the training batch size, from $\mathcal{N}(0, I)$ with the shape of the layer's input (for a convolution, the input feature map at the layer's own stride and padding). Each probe passes through the current W_l with the gradient stopped, and each minibatch contributes the input-output correlation $\frac{1}{N} \sum_j (W_l n_j)\, n_j^\top$. For a convolution this is the gradient, with respect to the kernel, of the inner product between the probe convolved by that kernel and its stored response. The mean over the k minibatches is rescaled to the Frobenius norm of W_l, and B_l moves towards it with step η_B. The mirror update adds no decay term to B, and W decays only through the optimiser's decoupled weight decay (below). The final dense layer takes the same update. B starts from a draw independent of W, with He-scaled magnitudes and random signs. Within this update the two settings differ only in η_B and k. Akrout et al. (2019) average their mirroring update over 2048 noisy examples at $\eta_B = 0.1$, and only that value of η_B is taken from their paper.

Networks and optimisation. All networks are convolutional, carry no normalisation layer, and are trained with a single global cross-entropy loss and crop-and-flip augmentation with an 8-pixel cutout. The four architectures are ResNet-18 with and without its residual shortcuts, and a 10-layer residual network with its shortcut-free 9-layer chain. The two smaller networks were each validated against automatic differentiation. Every run uses AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and decoupled weight decay 10^{-4}, a batch size of 128, and global-norm clipping at 1.0; the iterative trainer also softly clips each weight matrix's spectral norm after every step. The learning rate rises linearly from zero over the first two epochs and then follows a cosine decay to 10^{-6} at the last step. The last 5000 of the 50000 CIFAR training images form a validation split, and the other 45000 are trained on. Most comparisons run 15 epochs; the 9-layer chain, the 10-layer network and the cross-rule rows run 150.

Learning rates. Each setting's peak learning rate is selected on seed 0 by a two-sided bracket over a grid whose neighbouring rates differ by a factor of two to three, so no selected rate sits at the edge of its grid. The one exception is the common rate imposed on the $K = 17$ row, described below. The selected rate is then used for every seed. Table 5 lists every grid and selected rate, and which split the selection read. The eighteen-seed pair, the sequential ResNet-18 configuration on which the first shared-weight measurement and fork run, and four other configurations select on validation accuracy. The others, including the no-shortcut ResNet-18 of the second configuration, select on the seed-0 run's test accuracy, and that run also enters the reported mean. In each of those, validation accuracy selects the same rate, except for sign symmetry on the 9-layer chain, where it would select 3×10^{-4} instead of 7×10^{-4}. With $K = 17$ the tuned setting's accuracy still rises at the top of its grid, so that row imposes the low-budget setting's optimum on both settings. The shared-weight measurement, the fork and the rotations reuse the rates selected for their configuration, and the 9-layer chain's backpropagation reference is a single run at 7×10^{-4}.

Estimands and statistics. The estimand is the final-epoch accuracy for every run in the sequential trainer and for the eighteen-seed pair; the remaining iterative-trainer rows use the best epoch, which moves their gap downward by a fraction of a point. Every contrast is paired by seed. Intervals are two-sided 95% t-intervals on paired differences, and p-values are exact paired sign-flip tests, whose floor is 8×10^{-6} at $n = 18$ and 0.0312 at $n = 6$. At one common learning rate the one-minibatch

Table 5: **Every learning-rate grid and the rate selected from it.** Peak weight learning rates in units of 10^{-4}, bracketed on seed 0. *Validation*: the selected rate maximises accuracy on the held-out validation split. *Test, seed 0*: it maximises the seed-0 run's test accuracy. For sign symmetry the setting is the feedback scale.

configuration	setting	grid (10^{-4})	selected (10^{-4})	selected on
ResNet-18, iterative (Table 6 rows 1, 5)	tuned	1, 3, 7, 15	3	validation
	low budget	1, 3, 7, 15	3	
$K = 17$ (row 2)	tuned	1, 3, 7, 15	1 (imposed)	test, seed 0
	low budget	0.3, 1, 3, 7	1	
CIFAR-100 (row 3)	tuned	1, 3, 7, 15	3	validation
	low budget	0.3, 1, 3, 7	3	
credit norm equalised (row 4)	tuned	0.3, 1, 3, 7, 15	3	validation
	low budget	0.3, 1, 3, 7, 15	1	
ResNet-18, sequential (row 6; Secs. 4, 5)	tuned	1, 3, 7, 15, 30, 60	7	validation
	low budget	0.3, 1, 3, 7, 15, 30	1	
no shortcuts, sequential (row 7; Secs. 4, 5)	tuned	0.3, 1, 3, 7, 15, 30	3	test, seed 0
	low budget	0.3, 1, 3, 7, 15, 30	1	
9-layer chain (row 8)	tuned	3, 7, 15	7	test, seed 0
	low budget	0.3, 1, 3, 7	1	
10-layer residual (row 9)	tuned	3, 7, 15, 30	7	test, seed 0
	low budget	0.3, 1, 3, 7	3	
sign symmetry, iterative	–	1, 3, 7, 20	3	validation
sign symmetry, feedback scales	1.0	1, 3, 7, 15, 30	7	test, seed 0
	1.2	3, 7, 15	7	
	1.35	3, 7, 15	7	
	1.5	3, 7, 15	7	
	2.0	1, 3, 7, 15, 30	7	
sign symmetry, 9-layer chain	–	3, 7, 15, 30	7	test, seed 0
sign symmetry, 10-layer residual	–	0.3, 1, 3, 7, 15, 30	3	test, seed 0
backpropagation, sequential	–	3, 7, 15, 30	7	validation
2048 probes (Section 6)	low budget	0.3, 1, 3, 7	1	test, seed 0
$\eta_B \times k$ grid (Section 6)	$\eta_B = 0.1, k = 1, 2, 4$	0.3 to 60, 6 to 7 rates	1	test, seed 0
	$\eta_B = 0.2, k = 1, 2, 4$	1 to 15, 4 rates	3	
	$\eta_B = 0.4, k = 1, 2, 4$	0.3 to 60, 4 to 7 rates	7	

setting of the grid in Section 6 still destabilises, with $93.2\times$ the across-seed standard deviation of its two-minibatch neighbour, so the instability comes from the one-minibatch estimate.

Run-to-run variation. A seed fixes initialisation, data order and mirror probes but not the augmentation draws. Five repeats of one seed therefore span 2.29 pp, while with the draws shared, hardware non-determinism moves a 15-epoch run of the sequential ResNet-18 configuration by at most 0.14 pp. The fork of Section 5 shares every draw between its branches.

B THE NINE MEASUREMENTS, IN BOTH FORMS OF THE BATCH AVERAGE

Table 6 lists every measurement of the two weight-mirror settings behind Figure 2a. The rows are not nine independent replications: rows 1 and 5 share runs, rows 2 to 4 are variants in one implementation, and rows 6 to 9 are independent of those and of each other. Row 2 runs the iterative trainer to the exact $K = 17$, row 3 trains on CIFAR-100, and row 4 rescales each layer's credit to unit norm per sample before the weight update. In rows 1, 3 and 5 both settings' selected learning rates coincide, and row 2 fixes one common rate. Rows 4 and 6 to 9 let each setting take its own bracketed rate, which adds the retuning headroom of the better setting to its lead. The table gives each alignment difference both as the recorded cosine difference and in degrees, read at the low-budget setting's own level (the arccosine of its mean cosine), as in the main text. The difference takes both signs, three rows positive and six negative. More test batches do not narrow its range. On the 10-layer residual pair the cosine difference reads within 0.00003 at one test batch and at sixteen. Row 6, run again on fresh trajectories with sixteen batches, reads $+0.00144$ while its settings differ by 19.44 pp, and within those runs the batch count moves the cosine difference by less than 0.00002.

Both forms. Two forms of the batch average are in use. The global form takes one cosine over the flattened batch, and the per-sample form averages per-sample cosines. Table 7 recomputes the five iterative-trainer rows in both forms from the per-epoch, per-layer cosines. The magnitude stays within 0.00269 in cosine in both forms, and the sign flips with the form in three of the five rows.

Table 6: **Across nine measurements the alignment difference stays within $0.82°$ and takes both signs, while the accuracy gap never changes sign.** Tuned minus low-budget weight mirror, paired by seed, grouped by the form of the batch average each measurement used; a positive alignment difference means the tuned setting is better aligned. Rate: *same*, both settings' selected learning rates coincide; *fixed*, one imposed common rate; *own*, each setting at its own selected rate. Rows 1 to 5 appear in both forms in Table 7. Rows 6 to 9 were recomputed in the global form only. In rows 8 and 9 the alignment difference is read over three and four of the six seeds.

	network	trainer	epochs	variant	rate	n	Δ top-1 (pp)	95% CI	Δ cosine	Δ angle (°)
per-sample form, one test batch										
1	ResNet-18	iterative	15	–	same	6	$+13.62$	$[+11.01, +16.23]$	-0.00197	-0.58
2	ResNet-18	iterative	15	$K = 17$	fixed	6	$+6.08$	$[+4.85, +7.32]$	-0.00050	-0.15
3	ResNet-18	iterative	15	CIFAR-100	same	6	$+7.92$	$[+7.40, +8.43]$	-0.00009	-0.03
4	ResNet-18	iterative	15	unit-norm credit	own	6	$+13.17$	$[+12.22, +14.12]$	-0.00111	-0.32
global form, one test batch										
5	ResNet-18	iterative	15	–	same	18	$+16.36$	$[+12.63, +20.09]$	$+0.00046$	$+0.14$
6	ResNet-18	sequential	15	–	own	6	$+19.11$	$[+17.88, +20.34]$	$+0.00130$	$+0.39$
7	ResNet-18	sequential	15	no shortcuts	own	6	$+18.70$	$[+17.60, +19.81]$	$+0.00269$	$+0.82$
global form, sixteen test batches										
8	9-layer chain	sequential	150	–	own	6	$+10.12$	$[+9.52, +10.71]$	-0.00022	-0.07
9	10-layer residual	sequential	150	–	own	6	$+4.11$	$[+3.84, +4.39]$	-0.00112	-0.38

Changing the implementation alone moves this pair's two alignments differentially by an amount of the same order as the difference itself. The main text therefore reads only the magnitude of a difference of this size between two runs. The shared-weight differences of Section 4 are differences within one run, against one W and on the same test batches, and are read in the global form only.

Table 7: **The sign of a cross-setting difference flips with the form of the batch average in three of five rows, while its magnitude stays small in both.** Tuned minus low-budget weight mirror, cosine difference, iterative trainer, ResNet-18, one test batch; rows and rate labels as in Table 6. Row 4 rescales each layer's credit to unit norm per sample.

	variant	rate	n	per-sample form	global form
1	–	same	6	-0.00197	$+0.00053$
2	$K = 17$	fixed	6	-0.00050	-0.00038
3	CIFAR-100	same	6	-0.00009	$+0.00020$
4	unit-norm credit	own	6	-0.00111	$+0.00116$
5	–	same	18	$+0.00044$	$+0.00046$

The per-layer reading. Akrout et al. (2019) read the angle per layer, so we also report the worst layer. Averaged over runs, it sits at $21.75°$ for the tuned setting and at $21.94°$ for the low-budget one, and the worst layer of any run reaches at most $22.11°$ over five seeds per setting, sixteen layers and fifteen epochs. The same readout gives $88.26°$ on fixed random feedback and at most $0.01°$ with $B = W$, so it is not saturated.

C SIGN SYMMETRY'S FEEDBACK SCALE

We trained five feedback scales from 1.0, the unscaled F, to 2.0 in the sequential trainer, each with its own learning-rate bracket and eighteen seeds. Their mean cosines run from 0.67829 to 0.68513. Table 8 gives the four adjacent pairs. The two smallest cosine differences differ by 0.00007, while their accuracy differences are $+0.01$ and $+3.09$ pp, the second of the same sign in all eighteen pairs. A second draw with sixteen test batches keeps the two cosine differences within 0.00007 of each other while their accuracy differences read -0.50 and $+2.17$ pp, in the same order. A single fixed test batch biases a between-setting difference by 0.00042 to 0.00131.

The knob barely moves the cosine. Scaling F rescales B uniformly, and a cosine is invariant to that. At fixed weights, doubling the scale moves the mean cosine by 0.00021 on ResNet-18 and by less than 10^{-7} on the chain, while the ratio of delivered to exact credit norms, read by the same probe, moves by 0.80. The residual shortcut breaks the cancellation by more than three orders of

Table 8: **Two nearly equal alignment differences between adjacent feedback scales mark accuracy differences of $+0.01$ and $+3.09$ pp.** Sign symmetry, sequential trainer, ResNet-18, eighteen paired seeds per pair, global form over one test batch; the angle difference is read at the smaller scale's level. *CI excludes 0*: the cosine difference's interval excludes zero. The last column is an independent second draw of the two pairs with the smallest cosine difference.

scales	Δ cosine	CI excl. 0	Δ angle ($^\circ$)	Δ top-1 (pp)	seeds, mean's sign	2nd draw (pp)
2.0 v 1.5	$+0.00119$	no	$+0.09$	$+0.01$	9/18	-0.50
1.2 v 1.0	$+0.00126$	no	$+0.10$	$+3.09$	18/18	$+2.17$
1.35 v 1.2	$+0.00194$	yes	$+0.15$	$+1.00$	13/18	–
1.5 v 1.35	$+0.00245$	yes	$+0.19$	$+0.55$	12/18	–

magnitude, and the between-scale differences are several times that direct effect. They are therefore set by the trained weights and not by the knob, which is why Section 4 measures both feedback matrices against one set of weights.

D THE INSTRUMENTS BEHIND THE SHARED-WEIGHT MEASUREMENT, THE FORK AND THE ROTATION

The shared-weight measurement. In the sequential trainer a second weight-mirror feedback matrix can be carried passively. It receives the same mirroring updates its setting would apply, from the same random initial matrix as the delivering matrix, but never delivers credit. The delivering and the passive matrix are measured against the one trained W on the same sixteen fixed test batches, in the global form, every epoch; the per-sample form of the passive matrix is not recorded. At the final epoch alone, the two shared-weight differences on ResNet-18 are -0.00201 and -0.00231 in cosine, and every one of the fourteen runs in both configurations keeps the sign of its fifteen-epoch mean. The control with two tuned matrices in one run uses separate probe streams for the two. Without shortcuts, the own-weight difference of Equation 2 is $+0.83^\circ$, just above the largest of the nine own-weight differences, and it splits into $+1.49^\circ$ between runs and -0.65° between matrices (rounded). The no-shortcut rows of Table 3a are measured on the keep branches of the forked runs of Section 5. The second configuration removes ResNet-18's residual shortcuts and keeps everything else, with each setting at the rate selected for that network (Table 5, the no-shortcut row).

The fork. Two branches continued from one snapshot must share every random stream, including library-global ones. We therefore continue both branches in one process from a snapshot of the complete epoch-5 state, including every generator. In cosine, averaged over epochs 6 to 15, the delivered B_{tuned} on the keep branch reads 0.00192 below the passive B_{low}, and on the fork branch the delivered B_{low} reads 0.00198 above the passive B_{tuned} (0.00195 without shortcuts). A restore-only control, which snapshots and restores without a hand-over, moves final accuracy by at most 0.14 pp over three seeds, and by at most 0.29 pp without shortcuts. Under a 50-epoch schedule on ResNet-18, with the same fork after epoch 5, the fork loses in all six seeds. Five lose between 2.20 and 3.27 pp, with mean -2.53 pp and CI $[-3.08, -1.98]$. In the remaining seed the forked run diverges to chance while both matrices' alignment improves, and loses 79.15 pp; with it the six-seed mean is -15.30 pp with CI $[-48.13, +17.53]$, which includes zero. In the 15-epoch runs forked after epoch 2 instead of epoch 5, all four seeds lose, by 9.09 to 66.31 pp, and one forked run falls to 21.94% accuracy. The mean, -26.50 pp with CI $[-69.69, +16.69]$, includes zero; a restore-only control at that epoch moves final accuracy by at most 0.32 pp over three seeds.

The rotation instrument. To set alignment we rotate each layer's delivered credit, at every step and at unchanged norm, towards or away from the credit $W^\top$ would transport, so that its cosine moves by a chosen offset or to a chosen target. The reference is read from W at rotation time, so every rotated run is a diagnostic intervention and none is offered as transport-free learning. Target 1 realises a mean cosine of 1.00000, and the rotation at zero offset reads 0.09 pp against the unrotated run, with a CI that includes zero. At target 1 the chain lands close to its backpropagation accuracy of 92.59%.

E THE ACCURACY VALUE OF A UNIT OF ALIGNMENT

Setting alignment directly does move a weight mirror's accuracy. The rotation instrument of Appendix D turns each layer's delivered credit towards or away from the credit $W^\top$ would deliver, within the plane the two credits span and at unchanged norm. Because it uses $W^\top$, it is a diagnostic and not a learning rule. On the 9-layer chain with the tuned setting, we raised each layer's cosine by 0.0027 in one run of a seed pair and lowered it by as much in the other, about the size of the largest own-weight difference. The raised run trains 1.04 pp better over six seed pairs, CI $[0.54, 1.54]$. Per 0.001 of cosine, about $0.3°$ at these angles, the same rotation gains $15.3\times$ more on ResNet-18 than on the 10-layer residual network, with the chain in between.

The rate does not extrapolate to exact alignment. On the chain we rotated a tuned mirror's credit all the way to exact alignment, closing an angle of $10.12°$. Over six seed pairs the 95% CI of the accuracy change runs from -0.25 to $+0.24$ pp, so the whole remaining angle is worth under a quarter of a point there. A difference in alignment therefore cannot be converted into a difference in accuracy without measuring the conversion in the configuration at hand.

Table 9 gives the rotations. Each row compares seed-paired runs rotated up and down by the same dose. The realised separation of the two cosines, the realised $|\Delta|$ column, sets the denominator of the rate, which is quoted per 0.001 of cosine. The smaller dose on ResNet-18 gives nearly the same rate as the larger one, so over the doses in Table 9 the rate does not depend on the dose. Zero-mean noise costs $22.9\times$ less accuracy per unit of cosine than the rotation on the 9-layer chain.

Table 9: **Setting alignment by intervention moves accuracy at a rate that varies $15.3\times$ across configurations.** Rotations of the delivered credit referenced to $W^\top$, tuned weight mirror, sequential trainer, $n = 6$ seed pairs per row. The rate is the accuracy change per 0.001 of realised cosine separation, $|\Delta|$; dose and $|\Delta|$ are in cosine.

network	epochs	dose	realised $\lvert\Delta\rvert$	Δ top-1 (pp)	95% CI	pp per 0.001
9-layer chain	150	$\pm$0.00270	0.00536	$+1.04$	$[+0.54, +1.54]$	0.1939
10-layer residual	150	$\pm$0.00270	0.00479	$+0.35$	$[+0.18, +0.52]$	0.0735
ResNet-18	15	$\pm$0.00270	0.00542	$+6.11$	$[+5.49, +6.74]$	1.1276
ResNet-18	15	$\pm$0.00065	$\approx$0.00143	$+1.57$	$[+1.28, +1.87]$	1.1018

F MORE MIRRORING AND THE MIRROR GRID

Section 6 rests on runs of ResNet-18 in the sequential trainer, 15 epochs, six seed pairs per contrast. Raising the low-budget setting to sixteen minibatches of probes, the 2048 probes of Akrout et al. (2019), leaves it 16.57 pp below the tuned setting, CI $[15.85, 17.30]$. The grid crosses $\eta_B \in \{0.1, 0.2, 0.4\}$ with $k \in \{1, 2, 4\}$ mirroring minibatches of 128 probes, six seeds per cell (Table 4). At $k = 1$ the larger rates raise the across-seed standard deviation to 19.04 pp at $\eta_B = 0.2$ and 29.03 pp at $\eta_B = 0.4$. A path through the grid from the low-budget to the tuned setting splits their gap in this trainer into three legs, each with a CI that excludes zero. Two minibatches instead of one gain 2.26 pp, the larger rate 15.95 pp, and four minibatches instead of two 0.90 pp.

G ORDERING RULES INSTEAD OF SETTINGS

Table 2 tests ranking rules by alignment on the 9-layer chain trained for 150 epochs, where the tuned mirror trains close to backpropagation. The two mirror settings inside the reported $25°$ range are 10.12 pp apart, and sign symmetry, far outside it, trains between them. Against the same sign-symmetry setting the two mirrors' cosine differences are 0.27587 and 0.27609, while their accuracy differences have opposite signs. With the shortcut restored, sign symmetry lands 5.42 pp below the tuned mirror and 1.26 pp below the low-budget one, after one of the 48 runs in that comparison diverged and was replaced. With that run retained, the second contrast has a CI that includes zero. On ResNet-18 in the iterative trainer, sign symmetry trains 4.21 pp above the low-budget mirror at $n = 18$, with an exact sign-flip p of 0.0084, although only 12 of 18 pairs agree in sign (sign-test $p = 0.2379$). These failures at cosine differences of 0.27609 and 0.31616 exclude any threshold at

or below 0.31616 as a general criterion for ordering rules. A higher threshold stays consistent with everything we measured.

H THE DEPTH-UTILITY CHECK

Whether a rule trained the network at all is the question of the depth-utility diagnostic of Hao et al. (2026), which compares a trained network with a baseline whose hidden blocks stay frozen. On ResNet-18 in the sequential trainer both mirror settings pass it, the low-budget setting by 19.56 pp and the tuned setting by 38.67 pp, over a frozen-blocks baseline of $48.98 \pm 0.62\%$ over three seeds.