CONTROLLING SYMMETRY DRIFT IN FORWARD-ONLY CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Edge devices need to adapt in the field despite limited access to backpropagation, making node perturbation an appealing alternative based on local activity, neuron-level perturbations, and scalar loss feedback. Yet persistent perturbation noise can drive parameter growth over long data streams and undermine a learner's ability to adapt to later tasks. We show that, to leading order in a frozen-state analysis, non-gradient perturbation noise produces a mean-zero diffusion along loss-invariant rescaling symmetries, whereas gradient updates have no first-order component along these directions. We introduce gauge-damped node perturbation (GD-NP), which periodically corrects symmetry-charge displacement through local reciprocal rescaling of each hidden neuron's incoming and outgoing weights. For positively homogeneous activations, each positive rescaling preserves the current function exactly, making the correction prediction-unbiased; the unclipped update contracts the targeted charge for an isolated layer pair, while the guarantee depends on the architecture. Across continual-learning streams, GD-NP substantially controls charge displacement on long ReLU runs and can improve late-task accuracy; tanh experiments and simulated quantized writes further show that the benefit depends on the activation and write protocol.

1 INTRODUCTION

Edge devices must learn under tight memory and compute constraints. Analog crossbars may lack efficient access to transposed weights; microcontrollers may not be able to store activations or compute high-precision gradients (Oripov et al., 2025). Yet these devices are expected to adapt in the field, on data they may never store. Node perturbation (NP) offers a forward-only alternative (Fiete & Seung, 2006; Hiratani et al., 2022; Oripov et al., 2025). It perturbs a neuron's pre-activation and combines the resulting scalar loss feedback with local presynaptic activity to update its synapses. The update avoids backpropagation, but its long-horizon behavior is less benign.

Under model mismatch, NP can exhibit weight diffusion and eventual loss instability (Hiratani et al., 2022). This is a serious problem for continual learning: a model must remain able to learn the next task, not just fit the current one. Loss of plasticity also occurs in gradient-trained networks (Dohare et al., 2024), and weight growth can reduce effective learning rates (Lyle et al., 2024).

Weight decay and fixed norm constraints can limit growth, but they generally change the represented function. Variance reduction is not a complete answer either: averaging reduces the charge variance, yet isotropic perturbations retain a component along symmetry directions, and antithetic sampling retains the same leading charge-noise term. We formalize this limit for scalar-reweighted estimators whose weights depend only on linearized loss feedback. This motivates a more specific question: *can we control symmetry-driven parameter growth without changing the network's current predictions?*

Our analysis starts from the rescaling symmetries of positively homogeneous networks. Dividing a hidden neuron's incoming weights by a positive constant and multiplying its outgoing weights by the same constant preserves the function. The difference between the corresponding squared norms, $b_{\ell i} = \|\mathbf{W}_\ell[i,:]\|^2 - \|\mathbf{W}_{\ell+1}[:,i]\|^2$, is conserved by gradient flow (Arora et al., 2018; Du et al., 2018; Kunin et al., 2021), but need not be conserved by a perturbation update. We derive the leading-order diffusion rate of this symmetry charge: its mean increment vanishes in the frozen-state analysis, while its second moment scales with the learning rate and number of independent perturbation draws. The linear-network result has a corresponding diagonal form for homogeneous activations.

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.

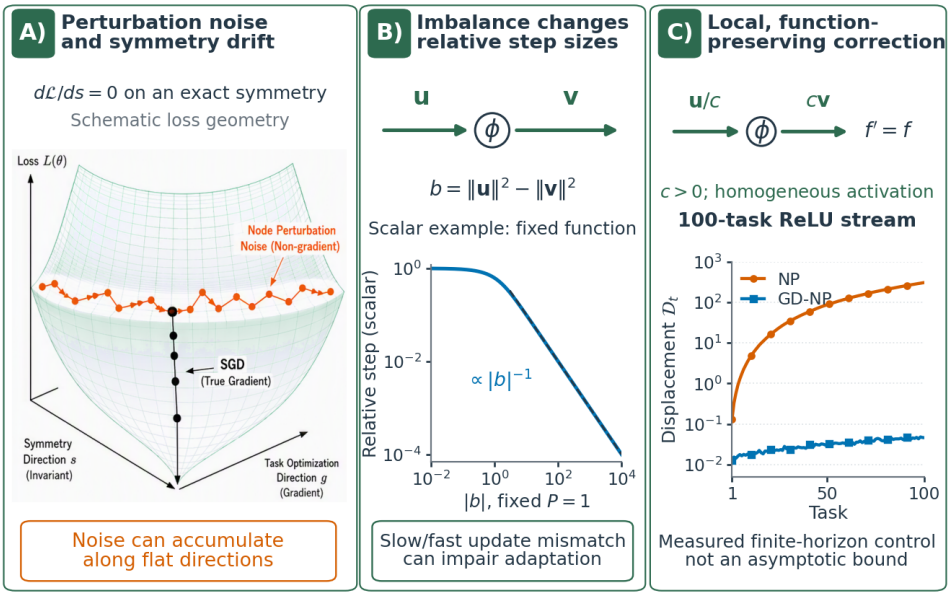

Figure 1: Symmetry drift, its effect on learning, and a local correction: A) Schematic loss geometry: a true gradient has no first-order component along a loss-invariant symmetry, whereas a perturbation estimate can move along it. SGD is a gradient reference, not a forward-only competitor. **B)** Here $b = \|\mathbf{u}\|^2 - \|\mathbf{v}\|^2$. The curve is an analytic scalar example at fixed $P = \|\mathbf{u}\|^2 \|\mathbf{v}\|^2 = 1$, showing the smaller relative gradient step divided by their geometric mean. Its large-imbalance asymptote is $1/|b|$, not a universal plasticity threshold. **C)** Reciprocal scaling, with c chosen by Eq. equation 5, preserves the function of a compatible homogeneous unit in exact arithmetic. The plot uses the actual 100-task ReLU results: mean displacement over ten seeds for NP and budget-free GD-NP, as in Figure 3. It demonstrates finite-horizon control, not proven asymptotic boundedness. Green highlights the schematic structure; the measured NP and GD-NP curves are orange and blue, respectively.

We then introduce gauge-damped node perturbation (GD-NP), a local correction that uses each hidden neuron's incoming and outgoing norms to choose a reciprocal rescaling. For positively homogeneous activations, this operation is an exact symmetry of the network function; the unclipped pairwise update contracts the targeted charge. This guarantee is architecture-specific and does not extend to non-homogeneous activations such as tanh. A fixed-function intervention tests whether changing only the parameterization can impair subsequent learning with a fixed optimizer. We separately evaluate long-horizon task adaptation, charge displacement, and sensitivity to simulated quantized writes.

Our contributions are: ❶ We derive a closed-form, leading-order diffusion law for symmetry charge, including its dependence on the learning rate and perturbation averaging, and test its predictions for linear and homogeneous networks. ❷ We prove that scalar-reweighted estimators based only on linearized loss feedback cannot eliminate charge noise from independent isotropic perturbations unless the update is trivial. ❸ We establish exact identities linking symmetry displacement to weight-norm excess and relative update scales, and use function-preserving orbit displacements to show that parameterization alone can impair fixed-optimizer relearning. ❹ We propose GD-NP, a local reciprocal-rescaling correction that preserves the function and, in its unclipped isolated-pair form, contracts the targeted charge for compatible homogeneous units; we evaluate its scope across continual-learning streams and simulated quantized-write protocols.

2 Related Work

Perturbation learning: Williams (1992) studies stochastic gradient estimators. Weight perturbation acts on parameters (Cauwenberghs, 1992); node perturbation instead perturbs activity (Fiete & Seung, 2006; Hiratani et al., 2022). Hiratani et al. (2022) identify mismatch-induced weight diffusion and the bias introduced by their normalization remedy. Our contribution is not the discovery of that instability, but its charge-level analysis and a correction that preserves a homogeneous network's current function. Recent methods explore decorrelation (Dalm et al., 2026), temporal noise structure

(García Fernández et al., 2024), orthogonal weight modifications (Ma & Yu, 2026), and natural perturbations (Sawada et al., 2025). Our scalar-feedback result does not exclude all such methods, particularly those that change the sampling geometry.

Symmetry and plasticity: Balancedness and symmetry-induced conservation laws are established tools for analyzing gradient dynamics (Arora et al., 2018; Du et al., 2018; Kunin et al., 2021). Parameter symmetry shapes SGD's noise equilibrium (Ziyin et al., 2024) and its entropic dynamics (Ziyin et al., 2025); prior work already connects symmetry to plasticity (Ziyin, 2024; Velarde et al., 2026). Path-SGD changes optimization geometry using rescaling invariance (Neyshabur et al., 2015). We instead study a local post-update correction for a forward-only estimator. Resetting, regenerative regularization, and clipping address broader plasticity issues (Dohare et al., 2024; Kumar et al., 2025; Elsayed et al., 2024). Warm-start generalization is a related but distinct concern (Ash & Adams, 2020). Gauge damping addresses a particular source of parameter growth and does not replace these interventions.

3 Setup and the Source of Symmetry Drift

Learning protocol: We study bias-free feedforward networks on a sequence of supervised tasks, carrying weights forward without resets or replay. Evaluation is on the current task after a fixed training budget. This measures continued adaptation, not retention of earlier tasks. Let $\mathbf{a}_\ell = \phi(\mathbf{W}_\ell \mathbf{a}_{\ell-1})$, with $\mathbf{a}_0 = \mathbf{x}$ and a linear readout. With N examples and G independent perturbation draws per example, NP uses

$$\widehat{\mathbf{g}}_\ell = \frac{1}{GN\sigma^2} \sum_{n=1}^{N} \sum_{j=1}^{G} \underbrace{\Delta\mathcal{L}_{nj}}_{\text{scalar feedback}} \underbrace{\boldsymbol{\xi}_{\ell,nj} \mathbf{a}_{\ell-1,n}^\top}_{\text{local synaptic factors}}, \qquad \mathbf{W}_\ell \leftarrow \mathbf{W}_\ell - \eta \widehat{\mathbf{g}}_\ell. \tag{1}$$

Here $\boldsymbol{\xi}_{\ell,nj} \sim \mathcal{N}(0, \sigma^2 I)$ is injected before the activation, and $\Delta\mathcal{L}_{nj}$ is the perturbed loss minus the clean loss for that example. The presynaptic activity in the update is from the clean pass. One-sided NP therefore needs **one clean and G perturbed forward evaluations**, as well as local eligibility information; it is not a single-forward-pass method. Under suitable differentiability conditions, its expectation approaches the true gradient as $\sigma \to 0$.

The symmetry and its charge: For $\phi(cz) = c\phi(z)$ and $c > 0$, the transformation

$$\mathbf{W}_\ell[i,:] \leftarrow \underbrace{\mathbf{W}_\ell[i,:]/c_i}_{\text{incoming rescaling}}, \qquad \mathbf{W}_{\ell+1}[:,i] \leftarrow \underbrace{c_i \mathbf{W}_{\ell+1}[:,i]}_{\text{outgoing rescaling}} \tag{2}$$

preserves the function. Define $b_{\ell i} = \|\mathbf{W}_\ell[i,:]\|^2 - \|\mathbf{W}_{\ell+1}[:,i]\|^2$. We call this a *charge* because it is conserved by gradient flow, not because it is constant along a symmetry orbit. Our displacement metric is $\mathcal{D}_t = \sum_{\ell,i} (b_{\ell i}(t) - b_{\ell i}(0))^2$. For a two-layer *linear* network the larger matrix symmetry has charge $B = \mathbf{W}_1\mathbf{W}_1^\top - \mathbf{W}_2^\top\mathbf{W}_2$. Only its diagonal has this conservation property for ReLU.

3.1 A Local Diffusion Law

Fix a two-layer linear network $f(\mathbf{x}) = \mathbf{W}_2\mathbf{W}_1\mathbf{x}$, with hidden width H and loss $\frac{1}{2}\|f(\mathbf{x}) - \mathbf{y}\|^2$. Write $\mathbf{a} = \mathbf{W}_1\mathbf{x}$, $\mathbf{e} = f(\mathbf{x}) - \mathbf{y}$ and $q(\mathbf{x}) = \|\mathbf{e}\|^2 + \left\|\mathbf{W}_2^\top \mathbf{e}\right\|^2$. Keeping the first-order loss response to the perturbation and the first-order charge response to the update gives

$$D_B = \underbrace{\frac{2\eta^2}{GN}}_{\text{rate scaling}} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[q(\mathbf{x}) \left\{ (H + \|\mathbf{W}_2\|_F^2) \|\mathbf{a}\|^2 + \|\mathbf{a}\|^2 + \|\mathbf{W}_2\mathbf{a}\|^2 \right\} \right]. \tag{3}$$

D_B is the second moment of this **linearized, frozen-state** charge increment. Its mean is zero. The factor 2 follows from the Gaussian fourth-moment identity, and the expectation must remain joint because the residual and activation depend on the same example (Appendix A).

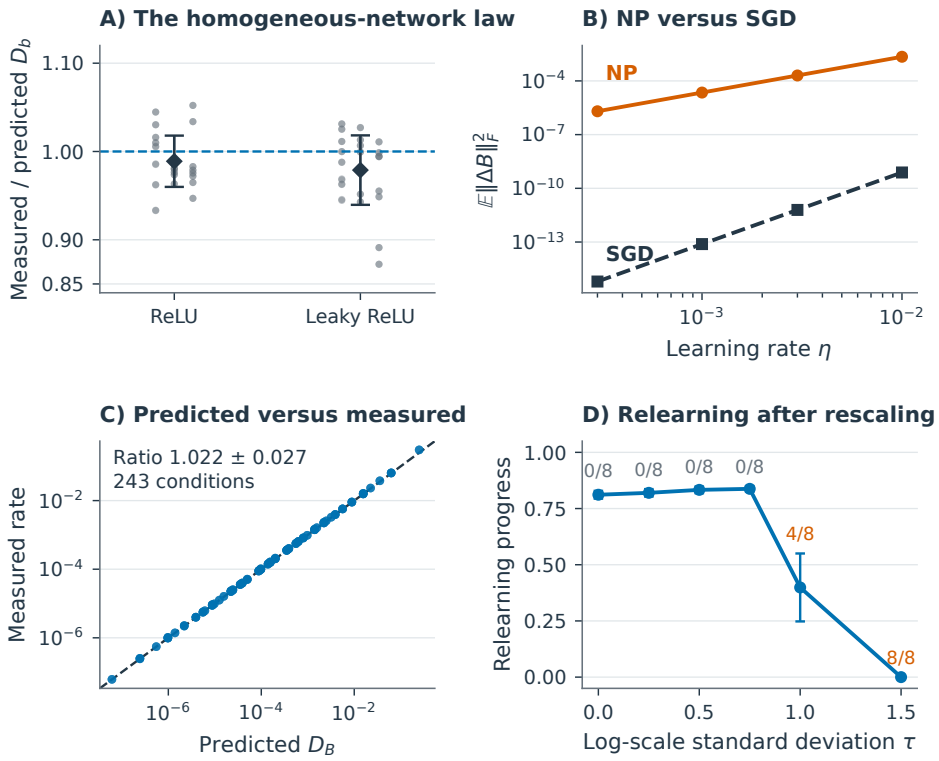

Figure 2: Local noise predictions and their consequence for learning: A) Measured/predicted diagonal rates for ReLU and leaky ReLU: 48 conditions each, with states horizontally offset for visibility. Diamonds and bars show the mean and standard deviation across conditions, not seed-level confidence intervals. **B)** At a fixed linear-network state, NP's charge second moment scales as η^2; SGD's starts at η^4. **C)** Frozen-state measurements versus Eq. equation 3, with an identity line and all 243 conditions. **D)** Relearning after a random function-preserving rescaling $\log c_i \sim \mathcal{N}(0, \tau^2)$. Points show mean progress and standard errors over eight seeds; labels give the number of failed runs. A failure is assigned zero progress. The optimizer is held fixed. Panels B–D use experiments E2, E1, and E16, respectively.

This law predicts quadratic dependence on η, inverse dependence on GN, and **no leading-order dependence on** σ. In a controlled single-step experiment, measured/predicted is 1.022 ± 0.027 (standard deviation across 243 conditions), with log-scale correlation 0.99997 (Figure 2). These conditions vary learning rate, batch size, draw count, perturbation scale, and target scale at a fixed width. They are not independent training seeds or a width-scaling study.

The distinction between a local rate and a global stability claim matters. An actual charge increment also contains terms quadratic in the weight update, so its mean need not vanish at finite η. If the state and noise statistics remain approximately constant, the leading-order model gives $\mathbb{E}\,\|B_t - B_0\|_F^2 \approx tD_B$. It does not establish linear growth throughout arbitrary nonlinear training.

For a homogeneous activation, write $z = \mathbf{W}_1\mathbf{x}$, $\mathbf{a} = \phi(z)$, $m = \phi'(z)$, and $r = \nabla_f \mathcal{L}$. The corresponding diagonal rate is

$$D_b = \frac{4\eta^2}{GN}\mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\left(\|r\|^2 + \left\|m \odot (\mathbf{W}_2^\top r)\right\|^2\right)\left(\|z\|^2 + \sum_i a_i^2 \left\|\mathbf{W}_2[:,i]\right\|^2\right)\right]. \tag{4}$$

The cancellation now uses $z \odot \phi'(z) = \phi(z)$. Figure 2A checks this rate in 96 finite-perturbation conditions across ReLU and leaky ReLU, using three fixed random states per activation. Thus the nonlinear check uses its own diagonal law, not the linear full-matrix formula (derivation and complete protocol in Appendix B).

Algorithm 1 One GD-NP update: homogeneous core

Require: Weights $\mathbf{W}$, initial charges b_0, batch, step t; $\eta, \sigma, G, \alpha, k$; optional decay $\lambda \geq 0$

Estimate all $\widehat{\mathbf{g}}_\ell$ by Eq. equation 1 on the same clean state
$\mathbf{W}_\ell \leftarrow \mathbf{W}_\ell - \eta(\widehat{\mathbf{g}}_\ell + \lambda \mathbf{W}_\ell)$ for every layer
if $t \bmod k = 0$ **then**
  for adjacent layer pairs, in order **do**
    Compute current $n_{\text{in}}, n_{\text{out}}$ for each neuron
    $T \leftarrow (1-\alpha)(n_{\text{in}} - n_{\text{out}}) + \alpha b_0$
    Compute u by Eq. equation 5, flooring norms and u at 10^{-12}
    $c \leftarrow \exp\{\text{clip}(\frac{1}{2}\log u, -0.1, 0.1)\}$
    Apply the **reciprocal rescaling** in Eq. equation 2
  end for
end if

3.2 Why gradient updates and perturbation estimates differ

For every example, a true gradient is orthogonal to the tangent of a function-preserving symmetry. Consequently, its *first-order charge increment vanishes*. SGD still changes the charge through finite-step $O(\eta^2)$ terms; it is not immune to symmetry-related dynamics. NP can instead have a **nonzero first-order charge variance**, even though its leading-order mean update is a gradient. This gives an η^{-2} ratio of NP to SGD charge second moments in the frozen-state experiment, with measured slope -2.00.

What variance reduction can achieve: Averaging independent draws reduces the leading charge variance by $1/G$. Antithetic pairing cancels even-order terms in the loss response but retains the same leading term as one-sided NP. More generally, scalar weights that depend only on linearized loss feedback cannot cancel the charge noise of independent isotropic draws unless the update is trivial (Lemma 1). This is a statement about charge variance, not an invariant ratio of gauge to functional update energy. Correlated directions, anisotropic sampling, and explicit projections fall outside this result.

4 Gauge-damped node perturbation

Rather than modifying the perturbation estimator, GD-NP periodically **corrects the accumulated charge**. For one hidden neuron, let n_{in} and n_{out} be its incoming and outgoing squared norms, $b = n_{\text{in}} - n_{\text{out}}$, and b_0 its initial charge. With damping fraction $\alpha \in (0, 1]$, choose the target $T = b + \alpha(b_0 - b)$. The rescaling in Eq. equation 2 attains this target when

$$\underbrace{c = \sqrt{u}}_{\text{finite rescaling}}, \qquad u = \frac{\sqrt{T^2 + 4n_{\text{in}}n_{\text{out}}} - T}{2n_{\text{out}}}. \tag{5}$$

For strictly positive norms this is the unique positive root of $n_{\text{out}}u^2 + Tu - n_{\text{in}} = 0$. We apply the correction every k NP steps, visiting adjacent layer pairs in order. Algorithm 1 gives the homogeneous core update.

Proposition 1 (Function preservation and local contraction)**:** *For **positively homogeneous activations**, a finite positive rescaling in Eq. equation 2 **preserves the network function**. For an isolated adjacent pair with positive norms, the unclipped update in Eq. equation 5 satisfies* $b_{\text{new}} - b_0 = (1-\alpha)(b - b_0)$.

The first property remains true for any clipped positive scale; the second does not. In the experimental implementation, $|\log c| \leq 0.1$ limits each correction. Neighboring pairs also share weights, so correcting one pair can change another pair's charge. Thus an individual correction is exact as a symmetry operation, but a single sweep is not an exact simultaneous projection of all charges. Degenerate norms and numerical evaluation are discussed in Appendix D, with the expanded update in Algorithm A2.

What the guarantee does not say: **Function preservation applies to the correction**, not to the full learning trajectory, the finite-σ estimator, or the outcome of future tasks. Rescaling changes the effective step sizes and noise levels seen by later updates. It therefore does not imply identical convergence, an unbiased final solution, or unconditional stability of training. A local mean-reverting approximation explains bounded charge fluctuations under stationary noise, but does not supply a global theorem for a nonstationary deep network (Appendix E).

Relation to weight decay: Uniform shrinkage also changes balancedness: $\mathbf{W} \leftarrow (1-\eta\lambda)\mathbf{W}$ gives $b \leftarrow (1-\eta\lambda)^2 b$. The distinction is not that weight decay cannot control the charge. It is that gauge damping selects a function-preserving direction, whereas weight decay generally changes the function. The two can be combined, but their combination need not improve accuracy in every setting.

5 Why parameterization can matter for plasticity

For an isolated neuron with positive incoming and outgoing norms, define $P = n_{\text{in}} n_{\text{out}}$ and $S = n_{\text{in}} + n_{\text{out}}$. Along its rescaling orbit, P is constant and

$$S = \sqrt{b^2 + 4P}, \qquad \underbrace{S - 2\sqrt{P}}_{\text{norm excess}} = \sqrt{b^2 + 4P} - 2\sqrt{P}. \tag{6}$$

The second expression is the squared-norm excess above the balanced representative of that orbit. It increases with $|b|$ even when the function does not change. This is an exact local identity, not a decomposition of all plasticity loss.

For small $|b|$, the excess in Eq. equation 6 is approximately $b^2/(4\sqrt{P})$.

The same rescaling changes gradient-based relative step sizes. If $r_{\text{in}} = \eta \, \|\mathbf{g}_{\text{in}}\| \, / \, \|\mathbf{W}_{\text{in}}\|$ and similarly for r_{out}, then

$$r_{\text{in}}(c) = c^2 r_{\text{in}}(1), \qquad r_{\text{out}}(c) = c^{-2} r_{\text{out}}(1). \tag{7}$$

In the scalar fixed-product case, the smaller relative step scales as $1/|b|$ at large imbalance; for general blocks, the exact ratio also depends on gradient norms (Appendix F).

A fixed-function intervention: We pretrain a ReLU network, rescale its hidden neurons with $\log c_i \sim \mathcal{N}(0, \tau^2)$, and train it on a new task using the same NP learning rate and noise scale. The mean relative change in probe outputs is approximately 2.3×10^{-7}, yet four of eight runs fail at $\tau = 1$ and all eight fail at $\tau = 1.5$ (Figure 2D; Appendix H). This isolates parameterization as a cause of failure for the fixed optimizer. It does not establish irreversible loss of representational capacity or exclude recovery by retuning the optimizer.

6 Experiments

Our primary evidence is a matched-configuration, 100-task Permuted MNIST study with ten seeds. We also examine shorter benchmark streams, convolutional and spiking architectures, and simulated quantized writes. We report **current-task accuracy and charge displacement separately**. For tanh, the balancedness statistic is an *algebraic diagnostic*, not a conserved symmetry charge. Protocols, selected hyperparameters, seed counts, and supplementary results appear in Appendix G.

Primary protocol: The long-horizon study uses a 784–128–128–10 MLP, 1,000 updates per task, batch size 64, $G = 1$, and $\sigma = 10^{-3}$. All arms within an activation share the learning rate: 10^{-3} for ReLU and 0.003 for tanh. Damping uses $\alpha = 0.05$, $k = 10$, and the per-event scale limit in Algorithm 1; decay uses $\lambda = 0.01$. The cumulative budget is either disabled or limits squared norms to their initial values times $[e^{-1}, e]$. A damping proposal is accepted when its relative output change on the current batch is at most 0.02; otherwise the rate is quartered, up to four trials, then the proposal is rejected. This check is redundant in exact homogeneous arithmetic but is retained in the implementation. Seeds 0 through 9 are paired across arms.

Table 1: All long-horizon configurations: Permuted MNIST, 100 tasks, ten seeds. Accuracy is in percent; late accuracy is mean ± s.e.m. over the last ten tasks. Early accuracy averages the first ten tasks. $\mathcal{D}$ is final balancedness displacement. R is the mean per-seed ratio $\mathcal{D}_{100}/\mathcal{D}_{51}$ used in this experiment, not an asymptotic boundedness test. Bold marks the largest late accuracy and smallest displacement within each activation, not statistical significance.

Rule	Budget	Early acc.	Late acc.	$\mathcal{D}$	R
ReLU					
NP	n/a	77.01	74.65 ± 0.12	3.0×10^2	3.39
NP + decay	n/a	77.23	78.40 ± 0.12	3.4×10^2	1.88
GD-NP	on	77.29	76.59 ± 0.17	1.4×10^2	4.62
GD-NP	off	77.32	77.43 ± 0.10	4.6×10^{-2}	1.42
GD-NP + decay	on	77.06	$\mathbf{79.02 \pm 0.07}$	1.5×10^2	2.37
GD-NP + decay	off	77.06	76.55 ± 0.06	$\mathbf{3.6 \times 10^{-2}}$	1.03
tanh					
NP	n/a	78.61	67.83 ± 0.17	2.0×10^4	3.68
NP + decay	n/a	79.55	78.65 ± 0.08	7.6×10^2	1.09
GD-NP	on	79.16	68.71 ± 0.13	1.5×10^4	7.75
GD-NP	off	79.18	33.61 ± 0.29	1.3×10^4	788.60
GD-NP + decay	on	79.65	$\mathbf{80.41 \pm 0.11}$	6.4×10^1	1.91
GD-NP + decay	off	79.64	79.74 ± 0.13	$\mathbf{1.7 \times 10^{-1}}$	1.02

6.1 Long-horizon accuracy and charge control

Table 1 reports every arm of the long-horizon experiment: NP, NP with weight decay, and gauge damping with or without decay and a cumulative scale budget. This budget limits incoming and outgoing squared norms relative to initialization. It differs from the per-event $|\log c| \leq 0.1$ limit, which remains active in all damping arms. Figure 3 shows the budget-free trajectories.

On ReLU, gauge damping without the cumulative budget reduces displacement by $6.65 \times 10^3\times$ relative to NP and raises late accuracy from $74.65 \pm 0.12\%$ to $77.43 \pm 0.10\%$. Weight decay attains $78.40 \pm 0.12\%$ with much larger displacement. Combining damping and decay gives the smallest displacement, but lower accuracy than either intervention alone. This is an important distinction: **charge control is not an accuracy objective**.

The tanh experiment exposes the boundary of the argument. Gauge damping alone reaches only $33.61 \pm 0.29\%$ late accuracy, below NP's $67.83 \pm 0.17\%$, despite using a batchwise output-change check. With decay, the budget-free combination reaches $79.74 \pm 0.13\%$ and reduces displacement by $1.18 \times 10^5\times$. The budget-on combination has higher accuracy, $80.41 \pm 0.11\%$, but greater displacement. These are different trade-offs, not evidence that the budget is universally harmful or that damping alone is safe on non-homogeneous networks.

These comparisons separate three controls that can otherwise look similar when summarized only by parameter norms. Weight decay shrinks all weights and can stabilize training, but it also changes the function and the scale of every layer's updates. Gauge damping instead moves along an exact function-preserving direction for ReLU, but it does not directly constrain all norms or optimize accuracy. The cumulative scale budget imposes bounds relative to initialization; it is neither a symmetry operation nor a charge-restoring rule. Consistent with these distinctions, the ReLU budget-on arms have substantially more residual displacement than their matched budget-free counterparts, while their accuracies do not rank in the same order as their displacement. A scale cap can limit the range of representatives available to the local correction, but these results do not isolate that interaction as the sole cause of the observed difference.

The activation changes the interpretation further. With ReLU, the charge tracks an exact rescaling symmetry, so correction can remove this component of parameter drift without directly changing current predictions. Tanh does not satisfy positive homogeneity: the same reciprocal rescaling generally changes activations and is not an exact motion on a function-invariant orbit. Its balancedness statistic is consequently an algebraic diagnostic, not a measure of symmetry drift. The batchwise output-change check rejects or shortens some proposals, but it cannot make the operation an exact symmetry or ensure that the trajectory remains close on unseen inputs. The poor performance of tanh

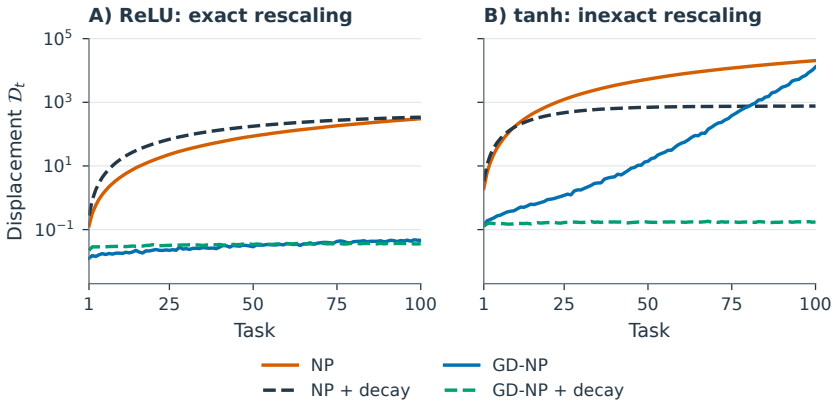

Figure 3: Charge control and its architectural limit: Mean displacement over ten seeds for the four configurations without a cumulative scale budget in E26. **A)** ReLU admits exact rescaling. **B)** Tanh does not, and damping alone eventually loses charge control. Curves share the same vertical scale. Per-task uncertainty was not retained in the aggregate result file, so no uncertainty bands are drawn. Table 1 includes the budget-on arms and the accuracy trade-off.

damping alone, and the different outcomes with decay and the cumulative budget, are therefore not contradictions of the ReLU guarantee. They mark its architectural boundary. Across both activations, charge displacement and current-task accuracy answer different questions; neither should be used as a surrogate for the other.

6.2 Composition and the strength of the baselines

At a matched learning rate on 20 tanh Permuted MNIST tasks, weight decay achieves $81.98 \pm 0.06\%$ mean accuracy and gauge damping achieves $79.87 \pm 0.11\%$. The combination gives $81.94 \pm 0.05\%$ with substantially lower balancedness displacement (Table A11). The data support complementary control mechanisms, not a claim that damping dominates weight decay. The norm split in that table is only an algebraic diagnostic for tanh; it cannot identify function-invariant growth.

The broader comparison uses method-specific validation searches and ten held-out seeds. Damping reduces displacement in all five reported compositions on Permuted MNIST. The conditioned sampler remains the most difficult case, with displacement 6483 before damping and 515 afterwards (Table A10). This comparison is descriptive: search budgets differ, and the NP and conditioned-NP damped variants were tuned separately from their parents. The low-rank and conditioned samplers are implementation-level controls, not faithful reproductions of the published LOCO or natural-perturbation algorithms. On random binary CIFAR-10 tasks, the conditioned damped variant attains $68.39 \pm 0.41\%$ mean accuracy, but its margin over the conditioned baseline is small relative to the reported uncertainty.

7 Hardware implications: simulated writes and the cost of a correction

The hardware study is a simulation, not a device demonstration. Table 2 reports all six controls in the quantized-write comparison. In the 8-bit tanh experiment, the gain over NP is $+0.42$, $+3.73$, and $+0.87$ percentage points when projection to the representable weights occurs after every update, every ten updates, or every hundred updates, respectively (Table A12). The ten-update condition gives paired $d_z = 12.07$ and Holm-adjusted $p = 0.0098$. For periods above one, the simulator performs intervening forward passes using full-precision working weights. These results therefore describe periodic quantization of a working copy, not inference restricted to a quantized array between writes.

One possible deployment places norm reductions and scale calculation in a peripheral digital controller alongside the crossbar array. At a scheduled correction interval, the controller computes a positive reciprocal scale from the stored initial charge and coordinates writes to both participating

Table 2: Accuracy and displacement under quantized writes: Tanh MLP, 20 tasks, ten seeds; accuracy is mean ± s.e.m. in percent. Each task receives 1,000 updates with batch size 64, $\eta = 0.003$, $G = 1$, and $\sigma = 0.001$. Periods above one use full-precision working weights between projections. Fixed configurations use $\alpha = 0.05$, $k = 10$, and trust tolerance 0.02 for damping, $\lambda = 0.001$ for decay, clipping at 1.5 times each layer's initial maximum magnitude, and rank 16 for low-rank NP. The tanh displacement is an algebraic diagnostic. Paired tests appear in Table A12.

Rule	Period 1	Period 10	Period 100	$\mathcal{D}$ (period 10)
NP	57.86 ± 0.40	70.15 ± 0.22	77.52 ± 0.09	1.1×10^3
NP + decay	58.10 ± 0.49	70.43 ± 0.21	77.67 ± 0.05	9.2×10^2
NP + clipping	57.86 ± 0.40	70.15 ± 0.19	77.50 ± 0.09	1.1×10^3
Low-rank NP	58.19 ± 0.44	70.24 ± 0.26	77.24 ± 0.10	1.1×10^3
GD-NP	58.28 ± 0.44	73.89 ± 0.19	78.39 ± 0.08	1.5×10^0
GD-NP + decay	58.37 ± 0.51	74.02 ± 0.19	78.58 ± 0.08	1.4×10^0

weight blocks. This requires no backward pass, but does demand access across adjacent arrays and sufficient temporary state. The gauge-correction schedule is distinct from the device's quantized-write projection schedule, though their alignment creates systems trade-offs. More frequent projections reduce the time between writes to the representable state but require additional quantization operations, and the measured gains interact non-monotonically with this projection period.

In a physical array, conductance-dependent and asymmetric writes can violate the product-preserving relation and prevent the device from realizing a function-preserving rescaling. While a simulated differential-pair control offers some mitigation, it is not a general guarantee against write asymmetry (Appendix J). Furthermore, the correction adds overhead: at $k = 10$, bare rescaling adds approximately 0.09% to the leading matrix-arithmetic count, rising to 6.8% with the output-change check, while pairwise writes add 11.4%. These arithmetic counts abstract away controller latency, verify reads, data movement, and our measured 36.1% software wall-clock overhead. Dedicated hardware-in-the-loop validation thus remains necessary to establish practical energy and endurance advantages (Table A4).

8 Limitations

The exact function-preservation result assumes a **compatible homogeneous architecture and exact arithmetic**. Tanh, finite-precision writes, normalization, fixed spiking thresholds, and additional parameter constraints require separate analysis. In particular, normalization changes the symmetry structure; it does not imply that all rescaling symmetries or plasticity problems disappear. The rate law is local, and the long-horizon results cover finite streams. A small endpoint growth ratio is evidence of stabilization over that interval, not a proof of asymptotic boundedness. The benchmarks use small models and current-task evaluation. The random class-pair streams are not standard class-incremental retention benchmarks, and the CIFAR-10 study uses grayscale flattened inputs. Width scaling also degrades the base estimator, a limitation that gauge damping does not resolve (Appendix K). Larger architectures, matched-compute comparisons with faithful recent baselines, and hardware-in-the-loop measurements remain necessary to establish a broader advantage. Furthermore, extending this framework beyond sequential feedforward topologies requires deriving the corresponding conserved charges for recurrent or attention-based models.

Conclusion

Perturbation estimates can change balancedness at first order even when their mean approximates a gradient. This creates a source of parameter growth that is absent from the first-order charge dynamics of SGD. Gauge damping provides a local way to control that growth while preserving the current function of a homogeneous network. The experiments show large reductions in charge displacement and, in some settings, better sustained adaptation. They also show why those outcomes must be distinguished: the symmetry must match the architecture, the write mechanism must be able to realize the correction, and low charge displacement alone is not a guarantee of useful learning. Ultimately, recognizing and controlling the structural drift inherent in non-gradient estimators is a necessary step toward robust, lifelong adaptation at the edge.

AI USE STATEMENT

In this work, we use generative AI tools to assist with editing the text and improve the presentation. We also used these tools to run code inspection and verification. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the code and results in the supplementary zip file. The supplementary material specifies the evaluation setting and links each table and figure to experiment records. Numerical tables and figures are created directly from those records. The accompanying code includes algebraic consistency tests, a separate finite-perturbation check of the diagonal rate, and checks of numerical assets and cross-references. These checks are distinct from the training experiments. The code separates forward-only training from autograd-based reference measurements.

REFERENCES

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 244–253. PMLR, 2018. URL https://proceedings.mlr.press/v80/arora18a.html.

Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3884–3894, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/288cd2567953f06e460a33951f55daaf-Paper.pdf.

Gert Cauwenberghs. A fast stochastic error-descent algorithm for supervised learning and optimization. In *Advances in Neural Information Processing Systems*, volume 5, pp. 244–251, 1992. URL https://papers.nips.cc/paper_files/paper/1992/hash/c06d06da9666a219db15cf575aff2824-Abstract.html.

Sander Dalm, Marcel van Gerven, and Nasir Ahmad. Node perturbation can effectively train multilayer neural networks. *Neuromorphic Computing and Engineering*, 6(3):034020, 2026. doi: 10.1088/2634-4386/ae9be4.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026): 768–774, 2024. doi: 10.1038/s41586-024-07711-7.

Simon S. Du, Wei Hu, and Jason D. Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/fe131d7f5a6b38b23cc967316c13dae2-Abstract.html.

Mohamed Elsayed, Qingfeng Lan, Clare Lyle, and A. Rupam Mahmood. Weight clipping for deep continual and reinforcement learning. In *Reinforcement Learning Conference (RLC)*, 2024. URL https://arxiv.org/abs/2407.01704.

Ila R. Fiete and H. Sebastian Seung. Gradient learning in spiking neural networks by dynamic perturbation of conductances. *Physical Review Letters*, 97(4):048104, 2006. doi: 10.1103/PhysRevLett.97.048104.

Jesús García Fernández, Nasir Ahmad, and Marcel van Gerven. Ornstein–Uhlenbeck adaptation as a mechanism for learning in brains and machines. *Entropy*, 26(12):1125, 2024. doi: 10.3390/e26121125.

Naoki Hiratani, Yash Mehta, Timothy P. Lillicrap, and Peter E. Latham. On the stability and scalability of node perturbation learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 31929–31941, 2022. doi: 10.52202/068431-2314. URL https://papers.nips.cc/paper_files/paper/2022/hash/cf38eb1549024cce4b3d2c1bb87a6c27-Abstract-Conference.html.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization. In *Proceedings of the 3rd Conference on Lifelong Learning Agents (CoLLAs)*, volume 274 of *Proceedings of Machine Learning Research*, pp. 410–430. PMLR, 2025. URL https://proceedings.mlr.press/v274/kumar25a.html.

Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel L. K. Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics. In *9th International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2012.04728.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, James Martens, Hado van Hasselt, Razvan Pascanu, and Will Dabney. Normalization and effective learning rates in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 106440–106473, 2024. doi: 10.52202/079017-3378. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c04d37be05ba74419d2d5705972a9d64-Paper-Conference.pdf.

Guoqing Ma and Shan Yu. Orthogonal weight modification enhances learning scalability and convergence efficiency without gradient backpropagation. In *ICASSP 2026 – IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4976–4980. IEEE, 2026. doi: 10.1109/ICASSP55912.2026.11464514.

Behnam Neyshabur, Ruslan Salakhutdinov, and Nathan Srebro. Path-SGD: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2422–2430, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/eaa32c96f620053cf442ad32258076b9-Abstract.html.

Bakhrom G. Oripov, Andrew Dienstfrey, Adam N. McCaughan, and Sonia M. Buckley. Scaling of hardware-compatible perturbative training algorithms. *APL Machine Learning*, 3(2):026107, 2025. doi: 10.1063/5.0258271.

Hiroshi Sawada, Kazuo Aoyama, and Yuya Hikima. Natural perturbations for black-box training of neural networks by zeroth-order optimization. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53063–53079. PMLR, 2025. URL https://proceedings.mlr.press/v267/sawada25a.html.

Osvaldo M. Velarde, Lucas C. Parra, Alireza Hashemi, and Hernan A. Makse. Emergence of fibrations, compression, and symmetry breaking in artificial neural networks. *CoRR*, abs/2609.01768, 2026. doi: 10.48550/arXiv.2609.01768. URL https://arxiv.org/abs/2609.01768. arXiv preprint.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696.

Liu Ziyin. Symmetry induces structure and constraint of learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62847–62866. PMLR, 2024. URL https://proceedings.mlr.press/v235/ziyin24a.html.

Liu Ziyin, Mingze Wang, Hongchao Li, and Lei Wu. Parameter symmetry and noise equilibrium of stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 37, pp. 93874–93906, 2024. doi: 10.52202/079017-2977. URL https://papers.nips.cc/paper_files/paper/2024/hash/aa933b5abc1be30baece1d230ec575a7-Abstract-Conference.html.

Liu Ziyin, Yizhou Xu, and Isaac Chuang. Neural thermodynamics: Entropic forces in deep and universal representation learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 56057–56089, 2025. doi: 10.52202/085713-1877. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/51173cf34c5faac9796a47dc2fdd3a71-Abstract-Conference.html.

A LINEAR-NETWORK DIFFUSION CALCULATION

The calculation below is conditional on fixed weights and an example $(\mathbf{x}, \mathbf{y})$. The input distribution and weights need not be isotropic. The perturbations must be independent isotropic Gaussians, and the loss response is linearized before its moments are computed.

First-order charge increment: Let $\mathbf{a} = \mathbf{W}_1\mathbf{x}$, $\mathbf{e} = \mathbf{W}_2\mathbf{a} - \mathbf{y}$, $\boldsymbol{\xi} = (\boldsymbol{\xi}_1^\top, \boldsymbol{\xi}_2^\top)^\top$. Define

$$h = \begin{bmatrix} \mathbf{W}_2^\top \mathbf{e} \\ \mathbf{e} \end{bmatrix}, \qquad Q = [I, -\mathbf{W}_2^\top], \qquad z = h^\top \boldsymbol{\xi}, \qquad v = Q\boldsymbol{\xi}.$$

The first-order weight changes are $\delta\mathbf{W}_1 = -\eta z \boldsymbol{\xi}_1 \mathbf{x}^\top/\sigma^2$ and $\delta\mathbf{W}_2 = -\eta z \boldsymbol{\xi}_2 \mathbf{a}^\top/\sigma^2$. The corresponding linear part of the charge change is

$$\delta B^{(1)} = -\frac{\eta z}{\sigma^2}(v\mathbf{a}^\top + \mathbf{a}v^\top). \tag{8}$$

Since $Qh = 0$, the jointly Gaussian variables z and v are independent. In particular, $\mathbb{E}[\delta B^{(1)} \mid \mathbf{x}, \mathbf{y}] = 0$. Here Q is a linear map on *node perturbations*; it is not an orthogonal projector in parameter space.

Second moment: Using $\left\|v\mathbf{a}^\top + \mathbf{a}v^\top\right\|_F^2 = 2\left\|v\right\|^2\left\|\mathbf{a}\right\|^2 + 2(v^\top\mathbf{a})^2$ and the Gaussian moments

$$\mathbb{E}[z^2\left\|v\right\|^2] = \sigma^4\{\left\|h\right\|^2 \operatorname{tr}(QQ^\top) + 2\left\|Qh\right\|^2\}, \tag{9}$$

$$\mathbb{E}[z^2(v^\top\mathbf{a})^2] = \sigma^4\{\left\|h\right\|^2 \mathbf{a}^\top QQ^\top\mathbf{a} + 2(\mathbf{a}^\top Qh)^2\}, \tag{10}$$

the Qh terms vanish. Since $QQ^\top = I + \mathbf{W}_2^\top\mathbf{W}_2$, this gives

$$\mathbb{E}[\left\|\delta B^{(1)}\right\|_F^2 \mid \mathbf{x}, \mathbf{y}] = 2\eta^2\left\|h\right\|^2\left[(H + \left\|\mathbf{W}_2\right\|_F^2)\left\|\mathbf{a}\right\|^2 + \left\|\mathbf{a}\right\|^2 + \left\|\mathbf{W}_2\mathbf{a}\right\|^2\right]. \tag{11}$$

Each example's first-order charge increment has conditional mean zero. Independent examples and independent draws therefore divide this second moment by NG, yielding Eq. equation 3. This conclusion assumes per-example loss differences and independent per-example perturbations as in Eq. equation 1. A single loss difference shared across an entire batch is a different estimator and requires its own calculation.

Joint versus factorized expectations: The residual factor $\left\|h\right\|^2$ and the activation-dependent bracket share the same example, so their product must be averaged jointly. Experiment E1 reports mean measured/predicted 1.022 for the joint expression and 1.156 for its separately averaged approximation. Both use the same empirical input distribution. We use the saved empirical-marginal prediction for this comparison, not the additional isotropic-input approximation recorded in the original experiment. The discrepancy in the latter is a property of this experiment, not a universal percentage or a theorem about positive correlation. Figure A4 shows the residuals as well as the overall fit.

Finite-step terms and SGD: For any weight increment, the exact charge change is

$$\begin{aligned}\Delta B = {} & \Delta\mathbf{W}_1\mathbf{W}_1^\top + \mathbf{W}_1\Delta\mathbf{W}_1^\top - \Delta\mathbf{W}_2^\top\mathbf{W}_2 - \mathbf{W}_2^\top\Delta\mathbf{W}_2 \\ & + \underbrace{\Delta\mathbf{W}_1\Delta\mathbf{W}_1^\top - \Delta\mathbf{W}_2^\top\Delta\mathbf{W}_2}_{\text{quadratic finite-step terms}}.\end{aligned} \tag{12}$$

For a per-example SGD step, $\Delta\mathbf{W}_1 = -\eta\mathbf{W}_2^\top\mathbf{e}\mathbf{x}^\top$ and $\Delta\mathbf{W}_2 = -\eta\mathbf{e}\mathbf{a}^\top$. The first line cancels, but the second generally does not. Thus SGD's charge increment is $O(\eta^2)$ and its squared increment is $O(\eta^4)$ at a fixed state. For NP these orders are $O(\eta)$ and $O(\eta^2)$, respectively. Neither statement implies that a finite-step trajectory is exactly driftless.

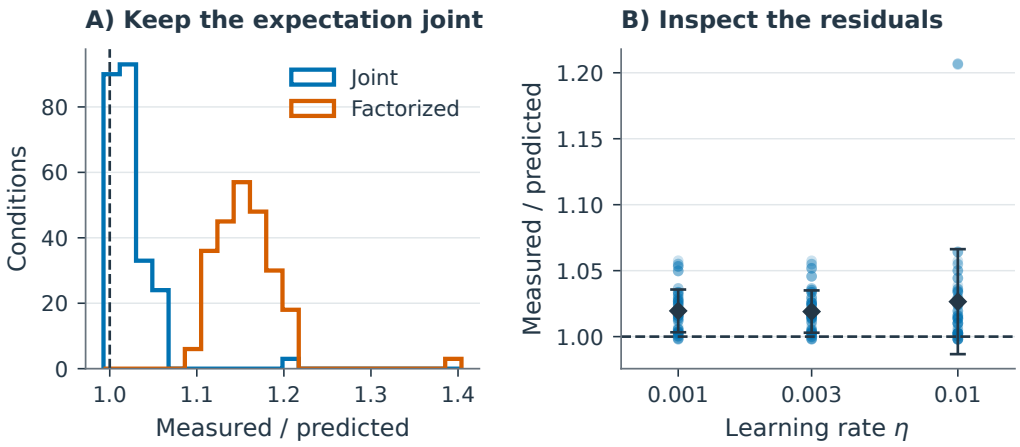

Figure A4: Residual checks for the local linear-network law: A) Distributions of measured/predicted rates for the joint expression and the separately averaged approximation. **B)** Individual joint-expression ratios at each learning rate, with means and standard deviations across conditions. The horizontal reference is 1. These are the 243 E1 conditions, not seed-level confidence intervals. The ratio plot exposes deviations that a wide-range log–log fit can hide.

B THE DIAGONAL RATE FOR HOMOGENEOUS ACTIVATIONS

The full-matrix law should not be used as an exact prediction for a ReLU network. A direct calculation gives the appropriate diagonal counterpart. For $f(\mathbf{x}) = \mathbf{W}_2\phi(\mathbf{W}_1\mathbf{x})$, let $z_1 = \mathbf{W}_1\mathbf{x}$, $\mathbf{a} = \phi(z_1)$, $m = \phi'(z_1)$, and $r = \nabla_f \mathcal{L}$. Assume differentiability at the clean pre-activations and use the linearized perturbation response

$$\Delta\mathcal{L}^{(1)} = h^\top \boldsymbol{\xi}, \qquad h = \begin{bmatrix} m \odot (\mathbf{W}_2^\top r) \\ r \end{bmatrix}, \qquad K = \begin{bmatrix} \mathrm{diag}(z_1) & -\mathrm{diag}(\mathbf{a})\mathbf{W}_2^\top \end{bmatrix}. \tag{13}$$

Then $\delta b^{(1)} = -2\eta(h^\top\boldsymbol{\xi})K\boldsymbol{\xi}/\sigma^2$. Positive homogeneity implies $z_1 \odot m = \mathbf{a}$, so $Kh = 0$. The same Gaussian calculation now gives

$$\mathrm{Cov}(\delta b^{(1)} \mid \mathbf{x}, \mathbf{y}) = 4\eta^2 \|h\|^2 KK^\top, \tag{14}$$

and hence

$$D_b = \frac{4\eta^2}{GN}\mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\left(\|r\|^2 + \left\|m \odot (\mathbf{W}_2^\top r)\right\|^2\right)\left(\|z_1\|^2 + \sum_i a_i^2 \|\mathbf{W}_2[:,i]\|^2\right)\right]. \tag{15}$$

This is a leading-order, frozen-state formula, not a claim about finite perturbations crossing activation boundaries. For ReLU the calculation applies away from its kinks. The consistency tests check $Kh = 0$, the covariance formula, and its failure for a non-homogeneous activation using exact Gaussian quadrature.

A finite-perturbation check: We separately compare Eq. equation 15 with actual one-step NP updates at three randomly initialized 6–8–3 networks per activation. Each state uses 256 Gaussian inputs and targets from a width-12 tanh teacher. The grid varies $\eta, \sigma \in \{10^{-4}, 10^{-3}\}$ and $G, N \in \{1, 4\}$, giving 48 conditions each for ReLU and leaky ReLU (negative slope 0.1). Each condition uses 8,192 repetitions in float64. Weights remain fixed between repetitions, and predictions use the exact empirical input distribution without fitted parameters. We reuse random numbers across η and σ at fixed state, G, and N, so these conditions are not independent seeds. Table A3 includes the range of observed ratios, not only their mean. The vectorized update is checked against the original NP implementation. This is a local moment check, not a training or plasticity benchmark.

Table A3: Finite-perturbation check of the diagonal law: Measured/predicted ratios for all 96 conditions. The standard deviation is across conditions, not a confidence interval. Per-condition Monte Carlo standard errors and all observations are included with the code.

Activation	Conditions	Mean ratio (s.d.)	Range
ReLU	48	0.989 ± 0.029	$[0.933, 1.052]$
Leaky ReLU	48	0.979 ± 0.039	$[0.872, 1.031]$

C What scalar loss feedback can remove

Let $D_{\mathbf{x}}$ map a vector of node perturbations to the concatenated parameter direction $(\boldsymbol{\xi}_\ell \mathbf{a}_{\ell-1}^\top)_\ell$ at a fixed clean example. Let J_b be the Jacobian of all diagonal charges with respect to those parameters, and set $\mathcal{H} = J_b D_{\mathbf{x}}$. If h is the concatenated pre-activation gradient, then $D_{\mathbf{x}} h = \nabla_{\mathbf{W}} \mathcal{L}$. Conservation of the charge under gradient flow therefore implies $\mathcal{H} h = 0$.

Lemma 1 (Residual charge noise under scalar feedback)**:** *Fix the state and example. Let $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_G$ be independent $\mathcal{N}(0, \sigma^2 I)$ vectors and $z_j = h^\top \boldsymbol{\xi}_j$. Consider an update $-\eta \sum_j c_j D_{\mathbf{x}} \boldsymbol{\xi}_j$, where each c_j depends only on $(z_1, \dots, z_G)$ and the fixed clean state, has finite second moment, and does not otherwise observe the perturbations. Its first-order charge increment δb satisfies*

$$\mathbb{E}[\delta b] = 0, \qquad \mathbb{E}\,\|\delta b\|^2 = \eta^2 \sigma^2 \,\|\mathcal{H}\|_F^2 \underbrace{\sum_j \mathbb{E}[c_j^2]}_{\text{scalar-weight energy}} . \tag{16}$$

If $\mathcal{H} \neq 0$, this variance is positive whenever at least one coefficient is nonzero with positive probability.

Proof. For each draw, z_j and $\mathcal{H}\boldsymbol{\xi}_j$ are jointly Gaussian with cross-covariance $\sigma^2 \mathcal{H} h = 0$, so they are independent. Independence of the draws makes the entire collection of charge directions independent of the collection of feedback scalars. Conditioning on all z_j, the charge directions remain independent, zero-mean Gaussians with covariance $\sigma^2 \mathcal{H}\mathcal{H}^\top$. Expanding the squared norm of their weighted sum and then averaging proves both statements. □

The lemma concerns *linearized* feedback. Higher-order losses, perturbation-dependent coefficients, correlated draws, or a modified sampling covariance require separate treatment. An antithetic pair can be treated as one effective draw at leading order because $(\mathcal{L}(\boldsymbol{\xi}) - \mathcal{L}(-\boldsymbol{\xi}))/2 = h^\top \boldsymbol{\xi}$ at that order.

Why this is not a gauge-fraction invariance theorem: Multiplying a single realized vector by a nonzero scalar preserves its angle. This does not preserve a ratio of expected energies or the angle of a weighted sum. For example, let X, Y be independent standard Gaussians and let the first coordinate be the functional direction. Both $X(X, Y)$ and $(X^3/3)(X, Y)$ have mean $(1, 0)$, but their gauge-to-functional second-moment ratios are $1/3$ and $1/7$. Thus even unbiased scalar reweightings can change that ratio.

D Rescaling, implementation, and cost

Proof of Proposition 1: Positive homogeneity gives $\phi(\mathbf{W}_\ell[i,:]\mathbf{a}_{\ell-1}/c_i) = \phi(\mathbf{W}_\ell[i,:]\mathbf{a}_{\ell-1})/c_i$. Multiplication of the corresponding outgoing column by c_i cancels this factor at the next pre-activation. For the charge, the transformed norms are $(n_{\text{in}}/u, u n_{\text{out}})$, so imposing $n_{\text{in}}/u - u n_{\text{out}} = T$ yields Eq. equation 5. Positivity of both norms makes the positive root unique. Substitution gives the stated contraction.

Numerical evaluation and degeneracy: For positive norms, the root can be evaluated without subtracting nearly equal positive numbers:

$$u = \begin{cases} \dfrac{2n_{\text{in}}}{\sqrt{T^2 + 4n_{\text{in}}n_{\text{out}}} + T}, & T \geq 0, \\ \dfrac{\sqrt{T^2 + 4n_{\text{in}}n_{\text{out}}} - T}{2n_{\text{out}}}, & T < 0. \end{cases} \tag{17}$$

Algorithm A2 Gauge-damped NP for a homogeneous network

Require: Initial charges b_0; batch of N examples; draws G; learning rate η; noise scale σ; damping α; period k; log-scale limit $s_{\max}$

Compute clean per-example losses and local presynaptic activities
$\widehat{\mathbf{g}}_\ell \leftarrow 0$ for every layer
for $j = 1, \dots, G$ **do**
    Draw independent node noise for every example and layer
    Compute $\Delta\mathcal{L}_{nj}$ from a perturbed forward pass
    $\widehat{\mathbf{g}}_\ell \leftarrow \widehat{\mathbf{g}}_\ell + \sum_n \Delta\mathcal{L}_{nj}\boldsymbol{\xi}_{\ell,nj}\mathbf{a}_{\ell-1,n}^\top/(NG\sigma^2)$
end for
$\mathbf{W}_\ell \leftarrow \mathbf{W}_\ell - \eta\widehat{\mathbf{g}}_\ell$
if $t \bmod k = 0$ **then**
    for adjacent layer pairs, in order **do**
        Compute current incoming and outgoing squared norms per neuron
        $T_i \leftarrow (1-\alpha)(n_{\mathrm{in},i} - n_{\mathrm{out},i}) + \alpha b_{0,i}$
        Obtain a positive u_i from Eq. equation 5, with the numerical safeguards described above
        $c_i \leftarrow \exp\{\mathrm{clip}(\frac{1}{2}\log u_i, -s_{\max}, s_{\max})\}$
        Apply Eq. equation 2 to the pair
    end for
end if

The experimental implementation uses the direct root, floors squared norms and u at 10^{-12}, and clips $|\log c| \le 0.1$. These choices keep scales positive, but can prevent exact target attainment. If a norm is zero, an arbitrary target charge need not be reachable by a finite rescaling. The contraction proposition explicitly excludes that case.

The sign in Algorithm 1 is the gradient-estimation convention: the loss difference enters the estimate positively and the parameter update subtracts it. Optional decay adds $\lambda\mathbf{W}_\ell$ to the estimate before this subtraction.

Cumulative budgets and tanh: The cumulative budget confines incoming and outgoing squared norms to their initial values multiplied by factors in $[e^{-B}, e^{B}]$. If the admissible scale interval is empty, the implementation leaves that neuron's scale unchanged. The tanh safeguard evaluates the relative output change on the current batch, accepting it if it is at most 0.02. It tries up to four rates, reducing each rejected rate by a factor of four, and restores the saved parameters after a rejection. This check requires a baseline forward pass and one per trial. It bounds neither the change on unseen inputs nor the cumulative change over training. E26 demonstrates that it does not prevent long-horizon tanh degradation.

Locality and hardware access: The correction uses incoming and outgoing norms plus an initial target charge per hidden neuron. It does not require a gradient or a global gauge-space projection. Computing the norms still requires reductions over weights, and realizing the rescaling requires modifying both adjacent arrays. A hardware implementation would need suitable peripheral arithmetic and access to these reductions. Their availability and cost are architectural assumptions, not consequences of the symmetry theorem.

Overview figure: Figure 1 separates three kinds of evidence. Panel A) is a qualitative schematic, not a measured trajectory. Panel B) uses a scalar linear unit at fixed positive gain, with $P = n_{\mathrm{in}}n_{\mathrm{out}} = 1$ and nonzero gradient. In this case the smaller relative step, normalized by the geometric mean of the two relative steps, is exactly $2/(\sqrt{b^2+4} + |b|)$; the dashed curve is its large-$|b|$ asymptote $1/|b|$. This special scalar identity does not remove the gradient-dependent offset in Eq. equation 19 for general weight blocks. Panel C) uses the same ReLU NP and budget-free GD-NP mean displacement curves from E26 as Figure 3, without smoothing or fabricated uncertainty bands.

Let P_{w} be the parameter count and let $Q_{\mathrm{w}} = \sum_{\ell=1}^{L-1}(|\mathbf{W}_\ell| + |\mathbf{W}_{\ell+1}|)$ count visits to weights over a pairwise sweep, where $|\mathbf{W}_\ell|$ denotes the number of entries. Here $P_{\mathrm{w}} = 118{,}016$ and $Q_{\mathrm{w}} = 134{,}400$; the interior matrix is visited twice. A batch forward pass costs approximately $2NP_{\mathrm{w}}$ arithmetic

Table A4: Cost accounting for a 784–128–128–10 **MLP:** Batch size 64, $G = 1$. Arithmetic percentages use the leading matrix-operation model described below; "with check" adds the two forward passes for a first-trial acceptance. Writes assume either one fused correction per weight or a separate correction for each adjacent pair. These are accounting models, not measured hardware costs. The CPU timing below is one measurement per method.

Period	Rescale (%)	With check (%)	Fused writes (%)	Pairwise writes (%)
1	0.89	67.56	100.0	113.9
5	0.18	13.51	20.0	22.8
10	0.09	6.76	10.0	11.4
50	0.02	1.35	2.0	2.3

operations, counting multiplication and addition separately. Two forward passes and the batch outer-product estimate therefore give approximately $6NP_{\mathrm{w}}$ for one-sided NP with $G = 1$. Norm reductions and rescaling add approximately $3Q_{\mathrm{w}}/k$, excluding lower-order per-neuron operations. The batchwise output-change check adds *two* forward passes per damping event on a first-trial acceptance, and more with backtracking. Table A4 includes this cost instead of equating a bare rescaling with the full checked implementation.

Target charges alone require 256 persistent scalars for this architecture; a cumulative budget additionally needs reference norms. The implementation also retains initial weights and temporary copies for the trust check. These state requirements should not be confused with a peak-memory measurement. The measured times are 2.02 ms per NP step and 2.75 ms per GD-NP step, a 36.1% increase. This timing uses tanh, $B = 1$, and output-change tolerance 0.02, with 50 warm-up steps followed by 1,000 timed updates. It is not a timing of the budget-free homogeneous core. These timings should not be substituted by the much smaller nominal arithmetic percentage.

E Local mean reversion is not global training stability

For one isolated charge with fixed $S = n_{\mathrm{in}} + n_{\mathrm{out}}$, an additive restoring field $-\lambda\nabla_{\mathbf{W}}\{\frac{1}{2}(b - b_0)^2\}$ induces

$$\frac{\mathrm{d}}{\mathrm{d}t}(b - b_0) = -4\lambda S(b - b_0). \tag{18}$$

Approximating the forcing by white noise with variance rate D gives an Ornstein–Uhlenbeck model with stationary variance $D/(8\lambda S)$. For a full matrix charge, the corresponding drift is $-2\lambda(\beta S + S\beta)$, where $\beta = B - B_0$ and $S = \mathbf{W}_1\mathbf{W}_1^\top + \mathbf{W}_2^\top\mathbf{W}_2$. With fixed positive eigenvalues s_i of S, its mode rates are $2\lambda(s_i + s_j)$. The additive-damping study E5 has a stationary-variance slope -1.019 versus λ, close to the local prediction -1. This is a test of that approximation, not of the global behavior of Algorithm 1.

For the ideal finite correction, suppose that the charge receives an independent, zero-mean increment ϵ_m between damping events, with $\mathbb{E}\epsilon_m^2 = v$, and is then contracted exactly by $a = 1 - \alpha$. The post-correction model $\beta_{m+1} = a(\beta_m + \epsilon_m)$ has stationary variance $a^2 v/(1 - a^2)$ for $0 < \alpha < 1$, and zero immediately after each correction when $\alpha = 1$. State-dependent noise, changing norms, clipped corrections, and interactions between layer pairs are excluded from this model. A finite trajectory or endpoint ratio alone cannot establish these assumptions.

F Norm and relative-step identities

Equation equation 6 follows from $(n_{\mathrm{in}} + n_{\mathrm{out}})^2 = (n_{\mathrm{in}} - n_{\mathrm{out}})^2 + 4n_{\mathrm{in}}n_{\mathrm{out}}$. For $P > 0$ and small $|b|$ the excess is $b^2/(4\sqrt{P}) + O(b^4/P^{3/2})$. The charge b and its displacement $b - b_0$ are different quantities: the minimum norm occurs at $b = 0$, not necessarily at the initial charge. Restoring b_0 therefore controls displacement without necessarily minimizing the norm.

For Eq. equation 7, the gradient transforms inversely to each weight block under the reparameterization. Writing $v = \frac{1}{2}\log\{r_{\mathrm{in}}(1)/r_{\mathrm{out}}(1)\}$ gives the exact expression

$$\frac{\min(r_{\mathrm{in}}(c), r_{\mathrm{out}}(c))}{\sqrt{r_{\mathrm{in}}(c)r_{\mathrm{out}}(c)}} = \exp(-|v + 2\log c|). \tag{19}$$

The offset v depends on the gradients. It cannot in general be replaced by a weight-only imbalance coordinate. At either extreme of a nondegenerate isolated orbit, $|b| \to \infty$ and the smaller relative step is proportional to $1/|b|$, with a constant depending on the reference weights and gradients. The uniform-layer rescaling study measures slope -0.990, consistent with this asymptotic behavior.

The weight-only index $\chi = \sqrt{\min(n_{\text{in}}, n_{\text{out}})/\max(n_{\text{in}}, n_{\text{out}})}$ is a useful imbalance diagnostic. It is not, without additional assumptions, an exact learning-rate penalty or a calibrated predictor of failure. For deeper networks, neighboring rescalings share weights. Summing isolated-pair excesses is not an exact global norm decomposition. We do not use it as one.

G Experimental protocols

Data and evaluation: Permuted MNIST uses the first $20{,}000$ training examples, normalized with training-set statistics. The first task uses the identity permutation; subsequent tasks use independent pixel permutations determined by the stream seed. Reported evaluation typically uses the first $2{,}000$ test examples. Training samples batches with replacement. Models are bias-free MLPs with two hidden layers of width 128, unless otherwise stated. All current-task classification results use cross-entropy.

The two class-pair streams sample two distinct classes at each task, relabel them to 0 and 1, and reuse a shared two-class head. Pairs can recur. We call these *random binary MNIST* and *random binary CIFAR-10*, rather than class-incremental MNIST or standard Split-CIFAR-10. CIFAR-10 images are converted to grayscale using weights $(0.299, 0.587, 0.114)$, normalized, and flattened. No old-task retention metric is reported.

Long-horizon study (E26): The 100 tasks each receive $1{,}000$ updates with batch size 64, $G = 1$, and $\sigma = 10^{-3}$. The learning rate is 10^{-3} for ReLU and 3×10^{-3} for tanh, shared by all six arms within each activation. Damping uses $\alpha = 0.05$, $k = 10$, a per-event log-scale limit of 0.1, and output-change tolerance 0.02. Decay uses $\lambda = 10^{-2}$. The cumulative budget is either $B = 1$ or disabled. Seeds are $0, \ldots, 9$. None of the twelve arms has a reported nonfinite-weight divergence; low accuracy in the tanh arm is therefore not the same event as this divergence flag. The endpoint ratio uses the 51st and 100th tasks because the midpoint is indexed from zero. We preserve that definition.

Single-step rate study (E1): A 20–12–8 linear student is fitted to a width-40 nonlinear tanh teacher on $4{,}096$ Gaussian examples. The student is pretrained with true gradients for $4{,}000$ steps. The grid is the Cartesian product

$$\eta \in \{10^{-3}, 3 \times 10^{-3}, 10^{-2}\}, \qquad G, N \in \{1, 4, 16\},$$
$$\sigma \in \{10^{-5}, 10^{-4}, 10^{-3}\}, \qquad \text{target scale} \in \{0.25, 1, 2\}.$$

Each condition estimates the actual one-step charge moment from $8{,}000$ repetitions with weights restored between repetitions. The calculation uses float64. The teacher's nonlinearity, rather than its width alone, provides model mismatch for this linear student.

Method-specific tuning (E6): Each stream has 20 tasks, 600 updates per task, batch size 64, tanh activations, $G = 1$, and $\sigma = 10^{-3}$. Mean current-task accuracy on stream seed 1000 selects the hyperparameters; seeds $0, \ldots, 9$ are held out for reporting. The grids are *not* equal-budget searches. All methods consider $\eta \in \{0.003, 0.01, 0.03\}$. Table A5 gives the additional dimensions and configuration counts. Table A7 records every selected setting for the ten independently tuned methods on all three streams. ANP, low-rank NP, and decay compositions inherit their parent settings. Their added damping uses fixed $\alpha = 0.05$, $k = 10$, and no cumulative budget or output-change check. These settings differ from the independently tuned damping variants. GD-NP and conditioned GD-NP are independently tuned entries, so their comparison to their parents does not isolate a fixed-hyperparameter damping effect.

What the sampler controls implement: The low-rank control samples a fresh random orthonormal activity-space basis per layer and draws Gaussian noise in that subspace, rescaled to match the isotropic expected squared norm. The conditioned control shapes a diagonal noise covariance from

Table A5: Validation search spaces: Counts include the three learning rates and one validation seed. A larger search is a potential source of advantage, not a guarantee of equal tuning.

Method	Additional choices	Count
NP; ANP; conditioned NP	None	3 each
NP + decay	$\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}\}$	9
NP + L2-init	$\lambda_0 \in \{10^{-4}, 10^{-3}, 10^{-2}\}$	9
Shrink-and-perturb	Shrink $\in \{0.99, 0.999\}$; period $\in \{100, 500\}$	12
Low-rank NP	Rank $\in \{8, 32\}$	6
GD-NP; conditioned GD-NP	$\alpha \in \{0.05, 0.3\}$; $k \in \{10, 50\}$; $B \in \{0, 1\}$; trust tolerance 0.02	24 each
Continual backprop control	Replacement rate $\in \{10^{-5}, 10^{-4}, 10^{-3}\}$	9

running node-signal moments and whitens presynaptic inputs using a regularized second-moment matrix. These are inspired by related perturbation methods, but the code does not reproduce the full published algorithms of Ma & Yu (2026) or Sawada et al. (2025). They should not be interpreted as benchmark results for those papers. The continual-backprop implementation is an autograd-based reference control, not a forward-only competitor.

Uncertainty and tests: Unless stated otherwise, error bars are standard errors across seeds. The E1 rate ratio instead reports variation across conditions. Reported paired tests use the same stream seeds; this does not imply identical random-number consumption across different algorithms. For the write-protocol study, Holm correction is applied to the five method-versus-NP comparisons *within each protocol*. It is not a joint correction across all protocols and experiments. Confidence intervals for paired accuracy differences are 95% bootstrap intervals. Large paired effect sizes can reflect low variation in a paired difference and should not be read as a comparable gain in percentage points.

Table A6: Sources and replication counts for the figures and tables: E1 and E2 are controlled mechanism studies, not ten-seed benchmark comparisons. The new small theory checks are kept separate from all training results.

Study	Result file stem	Replication
Rate; SGD comparison	`e1, e2`	Monte Carlo at fixed states
Additive mean reversion	`e5`	Controlled trajectory study
Tuned benchmark streams	`e6_pmnist, cimnist, cifar`	10 seeds
Spiking extension	`e13_spiking`	10 seeds
Write protocol	`e14_write_model`	10 seeds
Fixed-function relearning	`e16_drift_plasticity`	8 seeds
Smooth streams	`e19_smooth_stream`	10 seeds
Width and convolutions	`e20_scaling_conv`	10 seeds
Asymmetry control	`e21_asymmetry_fix`	10 seeds
Norm diagnostic	`e24_decomposition`	10 seeds
Cost accounting	`e25_overhead`	One timing per method
Long horizon	`e26_budget_study`	10 seeds

Table A6 lists the result sources and replication counts. All numerical assets are generated from the named JSON files. Their hashes and the available training-module fingerprints are recorded in the accompanying manifest. These fingerprints do not by themselves establish complete provenance: the experiment driver, package versions, and runtime environment must also be archived for an independently reproducible submission.

Table A7: Selected E6 hyperparameters: Selection uses validation seed 1000. Values not shown retain the method defaults described in the protocol. $B = 0$ disables the cumulative budget.

Method	η	Additional selected settings
Permuted MNIST		
NP	0.003	None
Antithetic NP	0.003	None
NP + decay	0.003	$\lambda = 0.01$
Shrink-and-perturb	0.003	$k_{\mathrm{SP}} = 100$, $\gamma = 0.99$
NP + L2-init	0.003	$\lambda_0 = 0.01$
Low-rank NP	0.003	$r = 32$
Conditioned NP	0.01	None
GD-NP	0.003	$k = 50$, $B = 0$, $\alpha = 0.3$, $\epsilon_{\mathrm{trust}} = 0.02$
Conditioned GD-NP	0.01	$k = 50$, $B = 0$, $\alpha = 0.05$, $\epsilon_{\mathrm{trust}} = 0.02$
Continual backprop control	0.03	$r_{\mathrm{replace}} = 1 \times 10^{-5}$
Random binary MNIST		
NP	0.01	None
Antithetic NP	0.01	None
NP + decay	0.01	$\lambda = 0.01$
Shrink-and-perturb	0.01	$k_{\mathrm{SP}} = 100$, $\gamma = 0.99$
NP + L2-init	0.01	$\lambda_0 = 0.01$
Low-rank NP	0.01	$r = 8$
Conditioned NP	0.03	None
GD-NP	0.01	$k = 50$, $B = 0$, $\alpha = 0.3$, $\epsilon_{\mathrm{trust}} = 0.02$
Conditioned GD-NP	0.03	$k = 10$, $B = 0$, $\alpha = 0.3$, $\epsilon_{\mathrm{trust}} = 0.02$
Continual backprop control	0.03	$r_{\mathrm{replace}} = 1 \times 10^{-4}$
Random binary CIFAR-10		
NP	0.01	None
Antithetic NP	0.003	None
NP + decay	0.01	$\lambda = 0.001$
Shrink-and-perturb	0.003	$k_{\mathrm{SP}} = 500$, $\gamma = 0.99$
NP + L2-init	0.003	$\lambda_0 = 0.01$
Low-rank NP	0.003	$r = 8$
Conditioned NP	0.03	None
GD-NP	0.003	$k = 50$, $B = 1$, $\alpha = 0.05$, $\epsilon_{\mathrm{trust}} = 0.02$
Conditioned GD-NP	0.03	$k = 10$, $B = 0$, $\alpha = 0.05$, $\epsilon_{\mathrm{trust}} = 0.02$
Continual backprop control	0.03	$r_{\mathrm{replace}} = 1 \times 10^{-5}$

H FIXED-FUNCTION RELEARNING

E16 uses a 20–48–48–8 ReLU student and width-40 ReLU teachers. Each teacher task contains $4{,}096$ training and $2{,}048$ test examples. NP uses $G = 4$, batch size 32, $\sigma = 10^{-3}$, and $\eta = 0.003$. After $1{,}500$ pretraining updates, each hidden layer receives independent per-neuron scales $c_i = \exp(\tau Z_i)$ with standard Gaussian Z_i. Thus τ is the *standard deviation* of log scales, not the absolute value of every log scale. The probe error uses 512 test inputs. The student then receives $1{,}500$ updates on a new teacher task.

Let L_{before} and L_{after} be new-task test losses before and after relearning. Progress is $(L_{\mathrm{before}} - L_{\mathrm{after}})/L_{\mathrm{before}}$. A run is marked failed if the final loss is nonfinite or exceeds ten times its initial loss, and receives progress zero rather than being discarded. Table A8 includes every tested displacement. This intervention changes conditioning, absolute perturbation scales relative to the weights, and effective learning rates simultaneously. It tests sensitivity of the specified optimizer to an equivalent parameterization, not the performance of a rescaling-adaptive optimizer.

Table A8: Relearning at the same initial function: Eight paired seeds per log-scale standard deviation τ. Probe error is relative output error. Progress is mean $\pm$ s.e.m., with the stated zero score for failures.

τ	Probe error	Relearning progress	Failed
0.00	0	0.811 ± 0.017	0/8
0.25	2.3×10^{-7}	0.820 ± 0.017	0/8
0.50	2.3×10^{-7}	0.833 ± 0.015	0/8
0.75	2.3×10^{-7}	0.838 ± 0.012	0/8
1.00	2.3×10^{-7}	0.399 ± 0.151	4/8
1.50	2.3×10^{-7}	0.000 ± 0.000	8/8

I BENCHMARK AND COMPOSITION RESULTS

Table A9 reports all independently tuned controls from E6, including the autograd-based reference. Table A10 gives the five Permuted MNIST compositions. The two use the same E6 outputs, while the longer-per-task E24 study in Table A11 is a separate protocol. Consequently, their mean accuracies are not interchangeable.

Table A9: Independently tuned controls (E6): Mean accuracy, first-five accuracy, and last-five accuracy are in percent; the mean carries s.e.m. over ten seeds. All MLPs use tanh. CBP uses backpropagation.

Method	Mean accuracy	First five	Last five	$\mathcal{D}$
Permuted MNIST				
NP	74.08 ± 0.08	75.14	72.92	3.3×10^2
Antithetic NP	74.11 ± 0.08	75.14	73.06	3.3×10^2
NP + decay	75.51 ± 0.06	75.37	75.52	2.4×10^2
Shrink-and-perturb	75.87 ± 0.13	75.12	76.25	3.1×10^2
NP + L2-init	74.53 ± 0.05	75.09	74.15	1.6×10^2
Low-rank NP	74.00 ± 0.09	74.83	72.86	3.5×10^2
Conditioned NP	74.18 ± 0.05	77.65	71.44	6.5×10^3
GD-NP	75.43 ± 0.08	75.43	75.08	3.5×10^{-1}
Conditioned GD-NP	75.80 ± 0.06	78.03	73.78	5.2×10^2
Continual backprop control	91.00 ± 0.01	90.68	91.11	1.8×10^1
Random binary MNIST				
NP	97.91 ± 0.16	98.28	97.29	9.5×10^2
Antithetic NP	97.92 ± 0.15	98.24	97.35	1.0×10^3
NP + decay	98.18 ± 0.14	98.34	97.78	4.9×10^2
Shrink-and-perturb	98.14 ± 0.14	98.32	97.79	4.0×10^2
NP + L2-init	98.10 ± 0.14	98.24	97.76	1.3×10^2
Low-rank NP	97.86 ± 0.16	98.28	97.13	1.1×10^3
Conditioned NP	98.03 ± 0.13	98.26	97.72	2.6×10^2
GD-NP	97.89 ± 0.20	98.22	97.21	3.2×10^0
Conditioned GD-NP	98.23 ± 0.11	98.36	98.03	3.2×10^{-2}
Continual backprop control	99.21 ± 0.05	99.23	99.12	7.4×10^0
Random binary CIFAR-10				
NP	66.44 ± 0.43	65.57	67.05	1.2×10^4
Antithetic NP	66.72 ± 0.47	66.02	67.39	1.5×10^2
NP + decay	66.47 ± 0.44	65.52	67.22	9.5×10^3
Shrink-and-perturb	66.85 ± 0.42	66.06	67.50	1.3×10^2
NP + L2-init	66.77 ± 0.47	65.97	67.40	7.6×10^1
Low-rank NP	66.75 ± 0.40	66.17	67.43	1.4×10^2
Conditioned NP	68.00 ± 0.44	66.87	68.75	1.7×10^3
GD-NP	66.70 ± 0.48	65.95	67.25	5.0×10^1
Conditioned GD-NP	68.39 ± 0.41	66.85	69.52	2.8×10^0
Continual backprop control	70.89 ± 0.42	69.65	71.44	8.5×10^0

Table A10: Composition on tanh Permuted MNIST (E6): Ten seeds, 20 tasks, 600 updates per task. Accuracy is mean $\pm$ s.e.m. in percent. Displacement is the algebraic balancedness diagnostic. NP and conditioned-NP damped variants are separately tuned; the other three compositions inherit their parent's selected settings.

Base rule	Acc. before	Acc. after	$\mathcal{D}$ before	$\mathcal{D}$ after
NP	74.08 ± 0.08	75.43 ± 0.08	3.3×10^2	3.5×10^{-1}

Table A11: A matched-configuration comparison (E24): Tanh Permuted MNIST, 20 tasks, 1,500 updates per task, ten seeds. The learning rate is 0.003 for all four rules. "Balanced reference" is the norm after numerical diagonal balancing; "excess" is the difference from that reference. For tanh this transformation changes the function, so these columns are not functional and function-invariant components of learning.

Rule	Accuracy	$\Delta \lVert \mathbf{W} \rVert^2$	Δ excess	Δ reference	$\mathcal{D}$
NP	79.78 ± 0.05	578.0	257.2	320.8	1.4×10^3
NP + decay	81.98 ± 0.06	-197.8	-95.1	-102.7	4.9×10^2
GD-NP	79.87 ± 0.11	396.6	-122.0	518.6	1.1×10^0
GD-NP + decay	81.94 ± 0.05	-32.0	22.8	-54.8	2.1×10^{-1}

The numerical balancing diagnostic uses repeated adjacent-pair rescalings. For a homogeneous network these stay on a symmetry orbit; for tanh they do not. Even in a homogeneous network, function preservation of each correction does not imply that the subsequent learning trajectory leaves the balanced reference norm unchanged. The observations in Table A11 should therefore not be used as a proof that decay and damping remove disjoint components.

J SIMULATED WRITE PROTOCOLS

E14 uses tanh MLPs on 20 Permuted MNIST tasks with 1,000 updates per task, batch size 64, learning rate 0.003, $G = 1$, and $\sigma = 10^{-3}$. The nominal range is fixed from initialization and is quantized to 8 bits. For period 1, the device model is attached to the rule. For periods 10 and 100, full-precision working weights are updated and projected periodically. There is no separately frozen quantized inference copy during the intervening steps. Table A12 and Figure A5 report the consequences of those distinct numerical protocols.

Table A12: The projection protocol changes the measured gain: E14, ten seeds. Accuracy and paired gain are in percent and percentage points, respectively. Accuracy uncertainty is s.e.m.; paired intervals are 95% bootstrap intervals. Holm correction covers the five comparisons to NP within each protocol.

Period	NP	GD-NP	Gain [95% CI]	p (Holm)
1	57.86 ± 0.40	58.28 ± 0.44	$+0.42$ $[0.23, 0.61]$	0.0195
10	70.15 ± 0.22	73.89 ± 0.19	$+3.73$ $[3.56, 3.92]$	0.0098
100	77.52 ± 0.09	78.39 ± 0.08	$+0.87$ $[0.71, 1.07]$	0.0098

Asymmetry control (E21): This is a separate ReLU experiment: 20 tasks, 600 updates per task, batch size 32, learning rate 0.001, and a ten-update write period. A snapshot is retained between writes; the accumulated weight difference is then passed through a device model with 8-bit quantization, write-noise coefficient 0.1, device-variation coefficient 0.1, and asymmetry 0, 0.005, or 0.02. Again, the working weights are used for intervening forward passes. The four arms combine NP or GD-NP with or without a simulated differential-pair mitigation (Table A13). This tests composition under a particular device model, not universal immunity to write asymmetry.

The exactness guarantee concerns the requested rescaling. Conductance-dependent hardware need not realize that transformation, so a matched write-path evaluation remains necessary. The reported simulations do not establish a remedy for general conductance nonlinearity, nor do they measure physical energy use or write endurance.

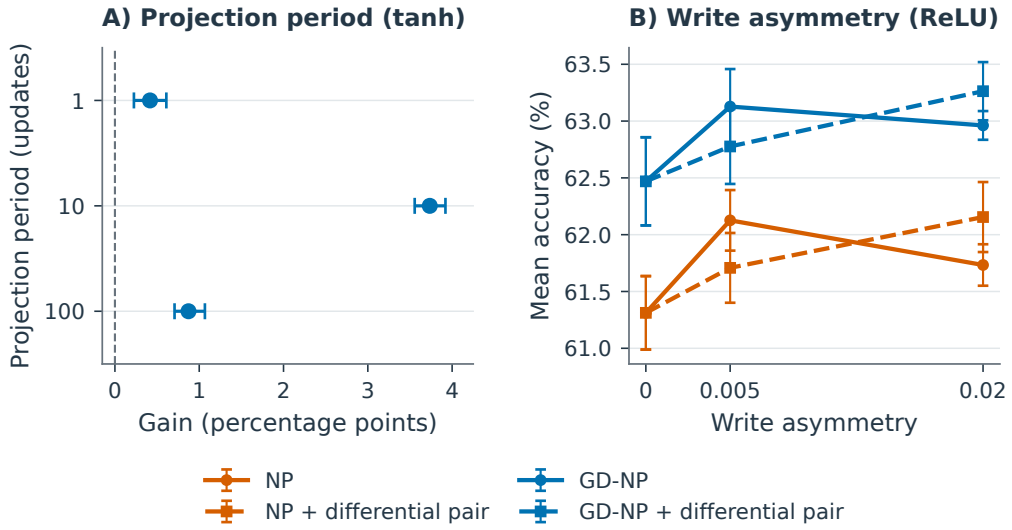

Figure A5: Device-model dependence: A) E14 paired accuracy gain of GD-NP over NP for three projection periods, with 95% bootstrap intervals. **B)** E21 mean accuracy and s.e.m. under write asymmetry. Solid curves omit the differential-pair control; dashed curves include it. E14 uses tanh and E21 uses ReLU, so the panels are different experiments, not points on a common performance curve.

Table A13: All asymmetry-control arms (E21): Ten seeds. Accuracy is mean ± s.e.m. in percent. Degradation is first-half minus last-half accuracy in percentage points, so negative values indicate improvement. Displacement refers to the diagonal ReLU charge.

Asymmetry	Arm	Accuracy	Degradation	$\mathcal{D}$
0.0	NP	61.31 ± 0.32	$+0.71$	4.5×10^{1}
	NP + differential pair	61.31 ± 0.32	$+0.71$	4.5×10^{1}
	GD-NP	62.47 ± 0.39	-0.75	7.2×10^{-1}
	GD-NP + differential pair	62.47 ± 0.39	-0.75	7.2×10^{-1}
0.005	NP	62.13 ± 0.27	$+0.29$	4.7×10^{1}
	NP + differential pair	61.71 ± 0.31	$+0.90$	4.6×10^{1}
	GD-NP	63.13 ± 0.33	-1.00	7.9×10^{-1}
	GD-NP + differential pair	62.78 ± 0.33	-0.58	7.2×10^{-1}
0.02	NP	61.73 ± 0.18	$+2.60$	6.4×10^{1}
	NP + differential pair	62.16 ± 0.31	$+0.87$	4.7×10^{1}
	GD-NP	62.96 ± 0.13	$+1.38$	1.4×10^{0}
	GD-NP + differential pair	63.26 ± 0.26	-0.38	7.5×10^{-1}

K TRANSFER AND ITS LIMITS

Width is an empirical stress test, not a verification of the linear-network law: E20 trains a one-hidden-layer ReLU classifier on MNIST for $2{,}000$ updates at batch size 32 and learning rate 0.001, over ten seeds. The reported diagonal rate is final displacement divided by the number of updates, not a frozen-state conditional variance. The estimator cosine averages 256 perturbation estimates before comparison with the true gradient. Table A14 shows that diagonal displacement grows with width and that even this averaged estimator becomes less aligned. The full-matrix proxy and its comparison with the linear law are not valid evidence that Eq. equation 3 quantitatively predicts ReLU training.

Table A14: Width stress test (E20): ReLU MNIST, ten seeds. Values are mean $\pm$ s.e.m. The charge rate is $\mathcal{D}_T/T$, and the cosine uses an average of 256 estimates. Only the diagonal charge is symmetry-compatible.

Width	Diagonal displacement per step	Estimator cosine
16	$3.6 \times 10^{-5} \pm 5.5 \times 10^{-6}$	0.968 ± 0.004
32	$4.0 \times 10^{-5} \pm 3.9 \times 10^{-6}$	0.937 ± 0.002
64	$7.5 \times 10^{-5} \pm 5.4 \times 10^{-6}$	0.902 ± 0.008
128	$1.2 \times 10^{-4} \pm 5.1 \times 10^{-6}$	0.808 ± 0.005
256	$2.1 \times 10^{-4} \pm 7.2 \times 10^{-6}$	0.747 ± 0.007
512	$3.9 \times 10^{-4} \pm 4.4 \times 10^{-6}$	0.665 ± 0.011
1024	$6.7 \times 10^{-4} \pm 1.3 \times 10^{-5}$	0.604 ± 0.014

Convolutions: For a bias-free homogeneous convolutional network, divide all filters producing channel i by c_i and multiply all next-layer weights reading that channel by c_i. This preserves the function and gives the per-channel charge $b_i = \|\mathbf{W}_\ell[i]\|^2 - \|\mathbf{W}_{\ell+1}[:, i]\|^2$, with norms over all spatial filter entries. Average and max pooling commute with positive scaling. At a convolution-to-dense transition, the outgoing norm must include the entire flattened spatial block for that channel. The two-layer linear rate law does not automatically transfer to convolutions with shared perturbations and spatial dependencies.

The 16–32 channel experiment uses 12 Permuted MNIST tasks, 800 updates per task, batch size 32, and learning rate 0.001. Per-channel damping uses $\alpha = 0.05$ and $k = 10$. It reduces final displacement by $130\times$ without a clear accuracy improvement (Table A15). Its endpoint ratio does not establish boundedness.

Table A15: Convolutional control (E20): Ten seeds. Accuracy is mean $\pm$ s.e.m. in percent. The endpoint ratio is the mean per-seed $\mathcal{D}_{12}/\mathcal{D}_7$, not a ratio of the two mean displacements.

Rule	Accuracy	Displacement	Endpoint ratio
NP	45.87 ± 1.35	3.0×10^3	3.22 ± 0.33
NP + decay	47.61 ± 1.08	2.9×10^3	3.58 ± 0.47
Per-channel GD-NP	46.05 ± 0.89	2.3×10^1	3.77 ± 0.51

No discrete task boundaries: E19 uses a 784–128–128–10 tanh MLP for $30{,}000$ updates, with batch size 64, $\eta = 0.003$, $G = 1$, and $\sigma = 10^{-3}$. MNIST inputs rotate in a fixed random two-dimensional subspace. The angle is $8\pi u$ for the smooth schedule and $8\pi\{u + 0.25\sin(6\pi u)\}$ for the varying-rate schedule, where u is the fraction of updates completed. Both schedules report lower algebraic balancedness displacement with damping (Table A16), without requiring task-switch detection. Because only two of 784 input directions are rotated, this is a mild nonstationarity test, not a substitute for strongly changing streams. Forty equally spaced probes are recorded; the endpoint ratio uses probes 40 and 21.

Spiking networks need a different coordinate: In the implemented ideal leaky integrate-and-fire model, scaling a neuron's incoming weights, adaptive threshold, and membrane trajectory by the same positive constant preserves its threshold crossings. This requires compatible initial states and reset dynamics. The outgoing weights do not receive the ReLU-style reciprocal scale, because

Table A16: Continuously changing inputs (E19): Ten seeds. The two schedules are analyzed separately. Ratios are the endpoint growth diagnostics, not tests of an asymptotic theorem.

Rule	$\mathcal{D}$ (smooth)	R (smooth)	$\mathcal{D}$ (varying)	R (varying)
NP	2.3×10^2	2.82	2.3×10^2	2.80
Antithetic NP	2.4×10^2	2.88	2.4×10^2	3.04
NP + decay	2.0×10^2	2.59	1.9×10^2	2.60
Low-rank NP	2.6×10^2	2.93	2.5×10^2	2.97
GD-NP	3.1×10^{-1}	1.73	3.2×10^{-1}	1.84
GD-NP + decay	2.0×10^{-1}	1.38	2.0×10^{-1}	1.39

emitted spikes are binary. Writing $p_i = \log \|\mathbf{W}_\ell[i,:]\| + \log \theta_i$ and $q_i = \log \|\mathbf{W}_\ell[i,:]\| - \log \theta_i$, the joint scale changes p_i and leaves q_i invariant. p_i is a coordinate along this scaling freedom, not a charge proved conserved by Euclidean gradient flow. The reported log-coordinate metric is $\mathcal{D}_p = \sum_{\ell,i}(p_{\ell i} - p_{\ell i,0})^2$; the ReLU-style diagnostic is the squared balancedness displacement $\mathcal{D}$ defined in the main text.

The exploratory 15-task, ten-seed LIF study reports an accuracy increase with this coordinate-specific correction (Table A17). It uses a 784–100–100–10 network with 20 simulation timesteps, membrane decay 0.9, 600 updates per task, batch size 32, $\eta = 0.003$, $\sigma = 0.1$, and $G = 4$. Adaptive thresholds use homeostatic rate 0.02 and target firing rate 0.15; damping uses $\alpha = 0.05$ every ten updates, and the decay control uses $\lambda = 0.001$. However, the corrected network's ReLU-style displacement $\mathcal{D}$ reaches approximately 10^{13}. The result illustrates the need to choose the correct symmetry; it does not establish acceptable dynamic range for a physical spiking implementation. With fixed thresholds, the stated joint rescaling is unavailable.

Table A17: Exploratory LIF extension (E13): Fifteen tasks, ten seeds. Accuracy is mean $\pm$ s.e.m. in percent. $\mathcal{D}_p$ and $\mathcal{D}$ are squared displacements in different coordinates; neither is a raw charge or a weight norm.

Rule	Threshold	Accuracy	$\mathcal{D}_p$	$\mathcal{D}$
NP	fixed	63.50 ± 0.35	3.7×10^0	7.4×10^1
NP + decay	fixed	63.26 ± 0.44	2.8×10^0	7.2×10^1
ReLU-style damping	fixed	64.44 ± 0.37	1.0×10^0	1.6×10^{-1}
NP	adaptive	64.43 ± 0.21	4.3×10^2	1.4×10^2
NP + decay	adaptive	64.73 ± 0.26	4.5×10^2	1.3×10^2
Spike-coordinate damping	adaptive	71.41 ± 0.24	1.2×10^1	1.1×10^{13}

What remains unresolved: The rescaling intervention is not compared with an optimizer that adapts both its learning rate and perturbation scale to the new parameterization. The broad benchmark suite uses unequal search budgets and simplified sampler controls. The hardware simulations do not measure physical energy, endurance, or inference from a fixed quantized array between writes. These gaps limit conclusions about general superiority, but do not alter the local charge calculation or the function-preservation property of a compatible rescaling.