# Weight Feedback Computes the Jacobian Transpose Locally in Modern Deep Networks

Junlong Shen[1,2] and Xingyu Li[1,2⋆]

[1] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada
[2] Alberta Machine Intelligence Institute (Amii), Edmonton, Canada
junlong6@ualberta.ca, xingyu@ualberta.ca

**Abstract.** Predictive Coding (PC) offers a biologically motivated alternative to backpropagation via local weight updates, yet routing error between layers still relies on an autograd Jacobian-transpose ($J^\top$) product—the last non-local operation in PC. We show that this dependency is largely avoidable. For any layer $f(x)$=Act(Norm($L(x)$)) with frozen normalization statistics, the exact $J^\top$ *factors* into three locally available terms, $J^\top v = L^\top(\mathbf{s} \odot \sigma'(\text{pre-act}) \odot v)$, where $\sigma'$ is the activation derivative and $\mathbf{s}$=$\gamma/\sigma_{\text{run}}$ is the normalization gain. Prior weight-feedback methods omitted both corrections; restoring them closes the transport gap for this layer class. Note that locality here holds *up to* three assumptions, which we state upfront—weight symmetry ($L^\top$ mirrors the forward operator, as assumed by all PC), a soft spectral-norm control that is not synapse-local, and a nearest-neighbour approximation for MaxPool. Substituting the identity into PC yields WF-Act-PC, which removes the autograd backward pass from error transport. On CIFAR-10/100 (50 epochs, 5 seeds), WF-Act-PC is the only PC method whose accuracy *improves* with depth, surpassing iPC—the strongest classical PC baseline—by 2.7–22.3 pp on CIFAR-10. With both methods tuned per architecture, it matches or exceeds a comparably-tuned backpropagation baseline on the deeper CIFAR-10 architectures (VGG-9: 93.57% vs. 92.43%; ResNet-18: 92.76% vs. 91.54%) and on the harder Tiny-ImageNet benchmark, while trailing tuned BP on the deeper CIFAR-100 VGG cells. Our WF-Act-PC implementation is publicly available at `https://github.com/jlshen025/pcax`.

**Keywords:** Credit assignment · Bio-plausible learning · Predictive coding · Feedback alignment · Jacobian transpose · Weight symmetry

## 1 Introduction

Backpropagation [21] computes weight gradients by propagating error signals through the Jacobian transpose $J^\top$ of each layer. Two properties of this procedure

⋆ Corresponding author.

lack established neural correlates [5]: feedback synaptic weights must mirror feedforward weights exactly (*weight symmetry*), and error signals must propagate globally before any weight updates (*update locking*). Predictive Coding (PC) [6, 20] resolves update locking through local weight updates at each layer [22, 25], making it among the most promising bio-plausible learning frameworks. Yet computing $J^\top$ during error transport—the operation that routes credit between layers—still requires autograd vector-Jacobian products (VJPs) [16]: the last non-local operation in PC. Feedback Alignment (FA) [15] takes the opposite approach, replacing $J^\top$ with a fixed random matrix $B$, but fails on deep convolutional networks [3, 13]. Computing exact error signals without a global backward pass—the $J^\top$ transport problem—has accordingly been regarded as an inherent limitation of weight-feedback architectures.

We revisit this view. For any layer of the form $f(x) = \mathrm{Act}(\mathrm{Norm}(L(x)))$—the standard building block of modern deep networks—weight feedback augmented with two locally computable corrections recovers the *exact* $J^\top$ (Fig. 1):

$$J^\top \cdot v \;=\; L^\top\big(\,\mathbf{s} \odot \sigma'(\text{pre-act}) \odot v\,\big), \tag{1}$$

where $L^\top$ is the transpose of the linear operator (the symmetric synapse), $\sigma'(\text{pre-act})$ is the pointwise activation derivative (analogous to an STDP eligibility trace), and $\mathbf{s} = \gamma/\sigma_{\text{run}}$ is the per-channel normalization gain (analogous to homeostatic divisive normalization [4]). The result is elementary—three applications of the chain rule (Observation 1); what matters is the *locality* of the factors, not the derivation. The key point is that frozen normalization—running normalization layers in eval mode during the inner loop, a natural choice for inference-loop stability—collapses $J_{\text{Norm}}$ to a fixed diagonal, reducing the full $J^\top$ to purely local factors.

**Scope and assumptions.** To clarify, the proposed method makes precise what "local" does and does not mean here. WF-Act-PC computes $J^\top$ from local factors *up to* three assumptions: (i) *weight symmetry*—$L^\top$ is taken to mirror the feedforward operator $L$, the same assumption PC (and BP) already make; (ii) a *soft spectral-norm control* applied to weights for stability, which uses power iteration and is therefore not synapse-local (though it runs on the slow learning timescale, and disabling it costs only 0.85 pp; Sect. 5.5); and (iii) a *nearest-neighbour approximation for MaxPool*, whose exact $J^\top$ would require cached argmax indices (the approximation costs ≈1.25 pp; Sect. 5.5). Within these bounds, Eq. (1) is exact for the non-pooling, frozen-normalization portion of the network.

The identity was overlooked for two historical reasons. Classical FA [15] predated BatchNorm [10]: FA analysed $f = \sigma(Wx)$, where $W^\top v$ is genuinely only an approximation—missing $\sigma'$. When normalization became standard, the FA community moved to learned feedback matrices [1, 12] and target propagation [14], treating the transport gap as inherent. PC, meanwhile, commonly freezes batch statistics during inner-loop inference (running BatchNorm in eval mode) but viewed this as an engineering workaround; the implication—that $J_{\text{Norm}}$ collapses to a fixed diagonal, rendering the remaining correction purely

local—went unnoticed. The two communities each held one piece: FA knew $W^\top$ was approximate but lacked frozen normalization; PC had frozen normalization but never examined its Jacobian structure.

Substituting Eq. (1) into the PC error-transport step yields WF-Act-PC, which removes the global backward pass from error transport—local up to the weight-symmetry and soft spectral-norm caveats above—with no auxiliary feedback networks and no target propagation. On CIFAR-10 (cross-entropy, 50 epochs, 5 seeds), WF-Act-PC surpasses iPC [22]—the strongest classical PC baseline—on every architecture by 2.7–22.3 pp, and, with both methods tuned per architecture, reaches BP-level accuracy on deep architectures (VGG-9: 93.57% vs. 92.43%; ResNet-18: 92.76% vs. 91.54%). On CIFAR-100, WF-Act-PC scales from 65.64% (VGG-5) to 71.07% (ResNet-18), exceeding tuned BP by +0.85 pp on ResNet-18, while iPC collapses (56.07 → 29.45%) and DPC-CN degrades (66.85 → 60.59%); on the deepest CIFAR-100 VGG cells, however, tuned BP catches up (a tie on VGG-7 and a 0.84 pp BP lead on VGG-9). On Tiny-ImageNet (200 classes), WF-Act-PC again improves with depth (46.0 → 61.1%, VGG-5→ResNet-18) and exceeds tuned BP at both depths.

**Contributions.**

1. **Theoretical.** We observe that, under frozen normalization, the exact Jacobian transpose of any $\mathrm{Act}(\mathrm{Norm}(L(\cdot)))$ layer factors into locally available terms—$\sigma'$ and $\mathbf{s}$ alongside $L^\top$ (Observation 1). This *locality* resolves the $J^\top$ transport problem for the layer class underlying modern deep networks.
2. **Algorithmic.** We derive WF-Act-PC, a PC algorithm for deep convolutional networks in which error transport is computed from local factors—local up to weight symmetry and a soft, slow-timescale spectral-norm control, with no autograd VJP in the inner loop.
3. **Empirical.** WF-Act-PC surpasses iPC, the strongest classical PC baseline, by 2.7–22.3 pp and is the only PC method whose accuracy improves with depth, on CIFAR-10/100 and Tiny-ImageNet. With both methods tuned per architecture, it reaches BP-level accuracy on deep architectures (VGG-7/9, ResNet-18), narrowing the accuracy gap between PC and backpropagation on deep convolutional networks.

## 2 Related Work

*Predictive Coding.* PC [6, 20] posits that the brain minimizes hierarchical prediction errors. Whittington and Bogacz [25] showed that PC weight updates approximate backpropagation gradients at convergence. Incremental PC (iPC) [22] improves convergence by updating weights simultaneously with states and is the strongest classical PC variant. The PCX benchmark [18] provides standardized evaluation: iPC reaches 85.51% on VGG-5/CIFAR-10 but degrades to 70.44% on ResNet-18—a depth-scaling failure shared by all PC methods tested.

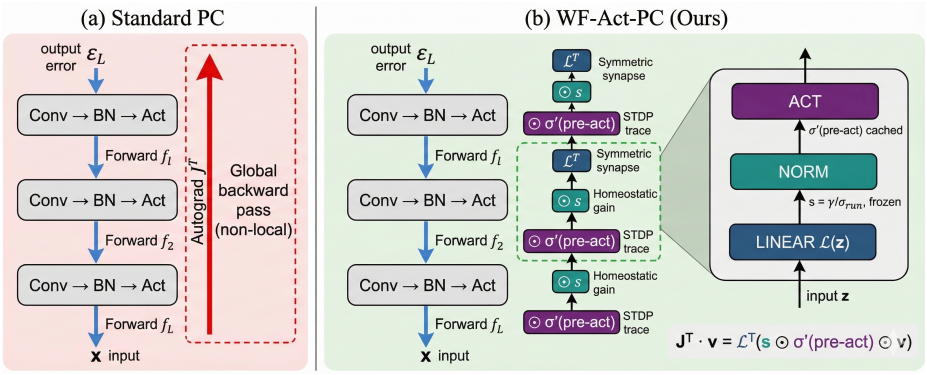



**Fig. 1: WF-Act-PC overview.** Each layer $f_l = \mathrm{Act}(\mathrm{Norm}(L(\cdot)))$ maintains three local factors: the symmetric synapse $L^\top$, the activation derivative $\sigma'$ (STDP trace), and the normalization gain $\mathbf{s} = \gamma/\sigma_{\mathrm{run}}$ (homeostatic gain). Error transport uses $J^\top v = L^\top(\mathbf{s} \odot \sigma' \odot v)$; no autograd backward pass is needed. The crossed-out arrow indicates the eliminated global $J^\top$ computation.

*Depth scalability in PC.* Several recent works address depth scaling from different angles. Qi et al. [19] diagnose *energy imbalance*—prediction error concentrating near the output—and propose exponentially decayed weighting (DPC-CN), achieving 89.46% on VGG-5 but degrading to 86.55% on VGG-9. Goemaere et al. [7] identify exponential signal decay in state-based PC and introduce error-based PC (ePC), a reparametrization that eliminates signal decay and matches backpropagation on deep architectures. Innocenti et al. [9] scale PC to 100+ layers via Depth-$\mu$P parameterization ($\mu$PC), but only on *fully connected* residual networks; the authors note that CIFAR-10 performance is "far from SOTA because of the fully connected (as opposed to convolutional) architectures used," and no convolutional results are reported. All three approaches improve depth robustness but *retain autograd $J^\top$ computation*—the non-local operation that our observation addresses—and none demonstrate competitive accuracy on standard convolutional benchmarks.

*Feedback alignment and the weight transport problem.* FA [15] replaces $J^\top$ with a fixed random matrix $B$; Direct FA [17] projects output errors directly to each layer. Both fail on deep CNNs [3,13]. Observation 1 offers a sharper diagnosis: for $\mathrm{Act}(\mathrm{Norm}(L(\cdot)))$ layers, $L^\top v$ is missing exactly $\sigma'$ and $\mathbf{s}$; restoring them makes weight feedback exact (given the symmetric $L^\top$ that FA was designed to avoid). FA underperformed here not because weight feedback is fundamentally limited, but because two local corrections were absent.

*Other bio-plausible methods.* Target propagation [14] learns separate inverse mappings; learned FA [1,12] trains feedback matrices toward $W^\top$. Both introduce auxiliary parameters and training complexity. WF-Act-PC avoids these: given

symmetric $W^\top$, two local corrections suffice for exact transport on this layer class.

## 3 Preliminaries

Consider a feedforward network with mappings $\{f_l\}_{l=1}^{L}$, parameters $\{W_l\}$, input $x$, and target $y$. PC defines layer-wise prediction errors $\boldsymbol{\varepsilon}_l = \mu_l - f_l(\mu_{l-1})$ and the variational free energy

$$E = \frac{1}{2}\sum_{l=1}^{L-1} \|\boldsymbol{\varepsilon}_l\|^2 + \mathcal{L}\big(f_L(\mu_{L-1}),\, y\big), \tag{2}$$

where $\mathcal{L}$ is the output loss (cross-entropy in all our experiments). Training alternates between *inference*—minimizing $E$ w.r.t. latent states $\{\mu_l\}$ (or equivalently, errors $\{\boldsymbol{\varepsilon}_l\}$)—and *learning*—updating weights $\{W_l\}$ at the equilibrated states.

*Error transport and the $J^\top$ problem.* Under a top-to-bottom (Gauss-Seidel) update schedule [16], the energy gradient w.r.t. $\boldsymbol{\varepsilon}_l$ takes the form

$$\frac{\partial E}{\partial \boldsymbol{\varepsilon}_l} = \boldsymbol{\varepsilon}_l - J_{l+1}^\top \cdot \boldsymbol{\varepsilon}_{l+1}, \tag{3}$$

where $J_{l+1}$ is the Jacobian of $f_{l+1}$ evaluated at the current state. Weight updates are local—$\partial E/\partial W_l$ depends only on activities at layers $l$ and $l-1$. The *sole non-local computation* is the $J_{l+1}^\top \cdot \boldsymbol{\varepsilon}_{l+1}$ term: for a typical Conv-BN-Act block, evaluating this product requires an autograd backward pass through the entire block.

*Error-based formulation (from ePC).* Following Goemaere et al. [7], we optimize directly over errors $\boldsymbol{\varepsilon}_l$ rather than states $\mu_l$, with states recovered as $\mu_l = f_l(\mu_{l-1}) + \boldsymbol{\varepsilon}_l$. This reparametrization eliminates exponential signal decay during inference: errors at all layers receive gradient signals from the first iteration, rather than waiting for signals to propagate layer by layer. The energy and gradient structure (Eqs. 2–3) remain identical; only the optimization variables change. WF-Act-PC adopts two components from ePC—error-based variables and a top-to-bottom sequential (Gauss-Seidel) sweep—and pairs them with an RMSProp inner-loop optimizer. Observation 1 replaces the remaining autograd $J^\top$ computation with a closed-form local formula, making the inner loop free of autograd VJPs. The two advances address distinct bottlenecks: ePC resolves signal decay; our observation eliminates non-local transport.

## 4 Method

### 4.1 Local Factorization of the Jacobian Transpose

Weight feedback is conventionally treated as an approximation of $J^\top$. We make a simple observation: for modern normalized layers, the exact $J^\top$—the same

quantity backpropagation computes—factors into terms that are each locally available, once two corrections are included. We record this as Observation 1.

**Observation 1 (Local factorization of the Jacobian transpose).** *Let* $f(x) = \mathrm{Act}(\mathrm{Norm}(L(x)))$ *where:*

- $L$ *is any linear operator (convolution, fully-connected, depthwise convolution, . . . ) with adjoint* $L^\top$*;*
- Norm *is any affine normalization with frozen per-channel scale* $\mathbf{s}$ *and shift* $\mathbf{b}$*, so that* $\mathrm{Norm}(z) = \mathbf{s} \odot z + \mathbf{b}$ *with* $\mathbf{s}$ *fixed during error propagation (BatchNorm [10]:* $s_c = \gamma_c/\sigma_c^{\mathrm{run}}$*; LayerNorm [2]:* $s = \gamma/\sigma_{\mathrm{LN}}$*; GroupNorm [26]:* $s = \gamma_g/\sigma_G$*;* no norm*:* $\mathbf{s} = \mathbf{1}$*);*
- Act *is any pointwise activation with derivative* $\sigma'$*.*

*Then for any vector* $v$ *in the output space of* $f$*:*

$$J^\top \cdot v \;=\; L^\top\Big(\mathbf{s} \odot \sigma'\big(\mathrm{Norm}(L(x))\big) \odot v\Big). \tag{4}$$

*Proof.* By the chain rule, $J = J_{\mathrm{Act}} \cdot J_{\mathrm{Norm}} \cdot J_L$, so $J^\top = J_L^\top \cdot J_{\mathrm{Norm}}^\top \cdot J_{\mathrm{Act}}^\top$. We evaluate each factor.

**Step 1** (activation). Since Act is pointwise, $J_{\mathrm{Act}} = \mathrm{diag}(\sigma'(\text{pre-act}))$, giving $J_{\mathrm{Act}}^\top \cdot v = \sigma'(\text{pre-act}) \odot v$.

**Step 2** (frozen normalization). Because $\mathrm{Norm}(z) = \mathbf{s} \odot z + \mathbf{b}$ with $\mathbf{s}$ frozen, $J_{\mathrm{Norm}} = \mathrm{diag}(\mathbf{s})$, giving $J_{\mathrm{Norm}}^\top \cdot u = \mathbf{s} \odot u$.

**Step 3** (linear operator). By definition of the adjoint, $J_L^\top = L^\top$.

**Composition.**

$$J^\top \cdot v = L^\top\big(J_{\mathrm{Norm}}^\top(J_{\mathrm{Act}}^\top \cdot v)\big) = L^\top\big(\mathbf{s} \odot \sigma'(\text{pre-act}) \odot v\big). \square$$

The derivation is elementary; what had gone unrecognised in prior weight-transport research is that both corrections are *locally available* (see Sect. 1 for the historical analysis). A dimension-tracking proof for the Conv+BN+Act case and instantiations for other normalization schemes appear in the supplementary material.

*Remark 1 (Frozen normalization is a feature, not a constraint).* Three independent motivations converge on frozen statistics. *Mathematically*, frozen $\mathbf{s}$ is required for $J_{\mathrm{Norm}}$ to collapse to a fixed diagonal (Step 2). *Algorithmically*, running BatchNorm in eval mode during the inner loop prevents corruption of running statistics, a natural choice for inference-loop stability. *Biologically*, homeostatic normalization gains are slow variables, stable across a single perception episode [4]. The convergence of all three motivations on the same condition is not coincidental: the same frozen statistics make PC biologically sensible and weight feedback exact.

*Remark 2 (Special cases).* Without normalization ($\mathbf{s} = \mathbf{1}$), Eq. (4) reduces to $J^\top v = L^\top(\sigma'(L(x)) \odot v)$—the textbook backprop formula; classical FA further drops $\sigma'$. For MaxPool, our implementation uses nearest-neighbour upsampling

(repeating each value over the 2×2 pool window), a local approximation that avoids caching argmax positions and is the one place Eq. (4) does not hold exactly. The autograd-vs-local ablation (Table 2) confirms this approximation incurs small accuracy loss.

*Remark 3 (Residual skip connections).* For $f(x) = f_{\text{plain}}(x) + x$, we have $J^\top v = J_{\text{plain}}^\top v + v$; Observation 1 covers $J_{\text{plain}}^\top$ exactly. The identity bypass ensures $\sigma_{\max}(J) \geq 1$, so spectral norm clipping alone cannot guarantee contraction. Top-layer gradient clipping ($\|v_{\text{top}}\| \leq c$) bounds errors before the Gauss-Seidel loop, enabling stable training with skip scale = 1.0.

**Corollary 1 (Local computability).** *Every factor in Eq. (4) is locally computable:*

| ***Factor*** | ***Computation*** | ***Neural mechanism*** |
|---|---|---|
| $L^\top$ | *Adjoint of feedforward operator* | *Symmetric synapse* |
| $\sigma'$(pre-act) | *Activation derivative at pre-act* | *STDP eligibility trace* |
| $\mathbf{s} = \gamma/\sigma_{\text{run}}$ | *Per-channel norm. gain* | *Homeostatic gain control* |

*No global backward pass or separate feedback weight matrices are required. The symmetric synapse $L^\top$, however, is the same $W^\top$ already assumed by the PC framework: WF-Act-PC makes this pre-existing assumption locally computable but does* not *remove it—a limitation we return to in Sect. 6.*

### 4.2 The WF-Act-PC Algorithm

Substituting Observation 1 into the PC error-transport step (3) yields a local energy gradient for each layer $l$:

$$\frac{\partial E}{\partial \boldsymbol{\varepsilon}_l} = \boldsymbol{\varepsilon}_l - L_{l+1}^\top\big(\, \mathbf{s}_{l+1} \odot \sigma'(\text{pre-act}_{l+1}) \odot \boldsymbol{\varepsilon}_{l+1} \,\big). \tag{5}$$

All quantities are locally available: $\boldsymbol{\varepsilon}_l$ and $\boldsymbol{\varepsilon}_{l+1}$ are adjacent-layer errors; $\mathbf{s}_{l+1}$ is the normalization gain ($\gamma_{l+1}/\sigma_{l+1}^{\text{run}}$ for BatchNorm, $\mathbf{1}$ without normalization); $\sigma'(\text{pre-act}_{l+1})$ is cached from the forward pass; and $L_{l+1}^\top$ is transposed convolution for convolutional layers or $W^\top$ for dense layers. We minimize $E$ w.r.t. $\{\boldsymbol{\varepsilon}_l\}$ via RMSProp gradient descent on Eq. (5), with a top-to-bottom Gauss-Seidel sweep exploiting the triangular dependency structure. The full procedure is given in Algorithm 1.

We monitor convergence via the *convergence gap*

$$\Delta = E_{\text{local}} - \tfrac{1}{2}\textstyle\sum_l \|\boldsymbol{\varepsilon}_l\|^2 \;\geq\; 0. \tag{6}$$

When $\Delta \approx 0$, the inner loop has converged and weight updates are exact local gradients; when $\Delta \gg 0$, updates are noisy. This provides an online diagnostic of training health without additional cost.

**Algorithm 1** WF-Act-PC — one training step

**Require:** Batch $(x, y)$; model $\{f_l, W_l\}$; iterations $T$; inference step $\eta_\varepsilon$; RMSProp decay $\alpha_{\text{rms}}$; weight step $\eta_w$; SNC threshold $\tau$; error clip $c$

**Forward pass:** compute $h_l = f_l(h_{l-1})$, $h_0 = x$; cache pre-act$_l$ and $\mathbf{s}_l$ for each $l$ ($\mathbf{s}_l = \mathbf{1}$ if no normalization)
$\boldsymbol{\varepsilon}_l \leftarrow \mathbf{0}$, $r_l \leftarrow \mathbf{1}$ for all $l$ ▷ Errors and RMSProp states
$\boldsymbol{\varepsilon}_L \leftarrow \tilde{y} - \text{softmax}(h_L)$; $\boldsymbol{\varepsilon}_L \leftarrow \boldsymbol{\varepsilon}_L \cdot \min\big(1,\ c\,/\,\|\boldsymbol{\varepsilon}_L\|\big)$ ▷ Output error + clip
**for** $t = 1$ to $T$ **do** ▷ Gauss-Seidel top → bottom
  **for** $l = L{-}1$ **downto** 1 **do**
    $g_l \leftarrow \boldsymbol{\varepsilon}_l - L_{l+1}^\top\big(\mathbf{s}_{l+1} \odot \sigma'(\text{pre-act}_{l+1}) \odot \boldsymbol{\varepsilon}_{l+1}\big)$ ▷ Local gradient (5)
    $r_l \leftarrow (1{-}\alpha_{\text{rms}})\, r_l + \alpha_{\text{rms}}\, g_l^2$
    $\boldsymbol{\varepsilon}_l \leftarrow \boldsymbol{\varepsilon}_l - \eta_\varepsilon\, g_l \,/\, \sqrt{r_l + 10^{-8}}$ ▷ RMSProp update
  **end for**
**end for**
Compute $E_{\text{local}}$; update $W_l \leftarrow W_l - \eta_w \cdot \text{Adam}(\nabla_{W_l} E_{\text{local}})$ ▷ Local weight update
**for** each layer $l$ with $\sigma_{\max}(W_l) > \tau$ **do** ▷ Soft SNC
  $W_l \leftarrow W_l - (\sigma_{\max}(W_l) - \tau)\, \boldsymbol{u}_l \boldsymbol{v}_l^\top$
**end for**

### 4.3 Soft Spectral Norm Clipping

Stable training requires controlling the spectral norm of each layer's weight matrix. After each weight update, if $\sigma_{\max}(W_l) > \tau$, we apply a soft rank-1 correction:

$$W_l \;\leftarrow\; W_l - \big(\sigma_{\max}(W_l) - \tau\big)\, \boldsymbol{u}_l\, \boldsymbol{v}_l^\top, \tag{7}$$

where $(\boldsymbol{u}_l, \boldsymbol{v}_l)$ are the leading singular vectors of $W_l$ (estimated by $n_{\text{steps}} = 20$ power iterations). Unlike hard clipping ($W \leftarrow W \cdot \tau/\sigma_{\max}$), this modifies only the leading singular direction, avoiding catastrophic weight rescaling during transient spikes.

Power iteration is *not* synapse-local—together with weight symmetry, one of the two assumptions under which WF-Act-PC is "local". It operates on weights (not error signals) and is decoupled from the per-sample inference loop—analogous to homeostatic synaptic scaling [24] on the slow learning timescale. Disabling SNC reduces accuracy by only 0.85 pp (Table 2), so WF-Act-PC's core advantage—local $J^\top$ transport—does not depend on it.

Independent of SNC, local error transport has engineering benefits: no global backward pass, layer-parallel weight updates, $O(1)$ rather than $O(L)$ activation memory per layer, and a natural mapping to neuromorphic hardware.

### 4.4 Convergence

**Proposition 1.** *With frozen normalization (and, without BatchNorm, this holds directly), the PC energy $E$ is quadratic in $\{\boldsymbol{\varepsilon}_l\}$. For fixed states $\{h_l\}$, the top-to-bottom Gauss-Seidel sweep exploits the triangular dependency structure: each $\boldsymbol{\varepsilon}_l$ update depends only on the already-updated $\boldsymbol{\varepsilon}_{l+1}$. When $\sigma_{\max}(J_{l+1}) < 1$ for all $l$, a single sweep converges to the global minimiser of $E$ w.r.t. $\{\boldsymbol{\varepsilon}_l\}$.*

In practice, the spectral condition is not enforced directly. Soft SNC maintains $\sigma_{\max}(W_l) \leq \tau$, an empirically sufficient proxy: with $\tau = 3.0$ and $T$=20 RMSProp iterations, the convergence gap $\Delta$ reaches near-zero on all architectures (Sect. 5.5). For residual architectures, gradient clipping of $v_{\text{top}}$ provides additional stabilization (Remark 3).

## 5 Experiments

### 5.1 Setup

*Datasets.* CIFAR-10 and CIFAR-100 [11]: 50,000 training / 10,000 test images across 10 and 100 classes. Both use RandomCrop(32, padding=4) and RandomHorizontalFlip(0.5), matching the PCX benchmark protocol [18]. We additionally evaluate on Tiny-ImageNet (200 classes, 64×64) to test depth scaling beyond CIFAR (Sect. 5.3).

*Architectures.* VGG-5/7/9 [23] (channel configs [128, 256, 512, 512], [128, 128, 256, 256, 512, 512], [128, 128, 256, 256, 512, 512, 512, 512]) and standard ResNet-18 [8], following PCX benchmark configurations. All four architectures use Conv+GELU blocks *without* BatchNorm ($\mathbf{s} = \mathbf{1}$ in Observation 1): the VGGs use Conv+GELU+MaxPool, and ResNet-18 adds residual connections. The main experiments therefore exercise the $\sigma'$ correction—and, for ResNet-18, the residual handling of Remark 3. We validate the remaining normalization-gain correction ($\mathbf{s} = \gamma/\sigma_{\text{run}} \neq \mathbf{1}$) separately by adding BatchNorm to VGG-5 (Sect. 5.5, Table 3).

*Hyperparameters.* All CIFAR-10 experiments share: Soft SNC ($\tau = 3.0$, $n_{\text{steps}} = 20$), warmup-cosine LR schedule (2 epoch warmup, decay to $10^{-6}$), AdamW ($\beta_1$=0.9, $\beta_2$=0.999, weight decay $10^{-4}$), RMSProp inner loop ($\alpha_{\text{rms}}$=0.2, $\eta_\varepsilon$=0.1), $T$=20, label smoothing 0.05, Cutout (patch size 8), gradient clipping (norm $\leq 1.0$), top-layer error clipping $\|v_{\text{top}}\| \leq 5.0$, batch size 128, and learning rate $\eta_w = 7{\times}10^{-4}$ with *no per-model tuning.*

*Baselines.* Prior PC and BP numbers are from the PCX benchmark [18] (50 epochs, 5 seeds, with augmentation): the benchmark's two backpropagation references, BP-CE and BP-SE (standard backpropagation with cross-entropy and squared-error loss, respectively), trained under its own recipe; iPC [22], the strongest PC-family method in the benchmark; DPC-CN [19], which adds exponential energy weighting; and ePC [7], which reparametrises PC in error coordinates. For a controlled comparison, we additionally report "BP (tuned)": backpropagation trained with the *same* augmentation recipe as WF-Act-PC (Cutout, label smoothing, cosine warmup-decay) and tuned per architecture (learning rate searched over $\{3, 7\}{\times}10^{-4}, \{1, 3\}{\times}10^{-3}$). Because WF-Act-PC and BP have different optimal learning rates, tuning each method at its own best LR is the fair protocol; the full grid is in Suppl. Mat. Sect. H. Any gap between the benchmark's BP references and BP (tuned) reflects differences in loss and training recipe, not the learning algorithm.

**Table 1:** Test accuracy (%) at 50 epochs (CE loss unless noted; BP-SE uses squared-error loss). BP-CE, BP-SE, and prior PC baselines: mean ± std over 5 seeds from [7, 18, 19]. WF-Act-PC and BP (tuned): mean ± std over 5 seeds. *Bio?*: "Yes" = $J^\top$ transport is locally computable via Observation 1 (no autograd VJP), up to weight symmetry and soft SNC; "Partial" = local weight updates but $J^\top$ requires autograd; "No" = global backward pass. **Bold**: best per column among methods using the controlled augmentation recipe; BP-CE and BP-SE are the benchmark's own BP references (cross-entropy and squared-error loss) and are excluded from bolding.

| | | CIFAR-10 | | | | CIFAR-100 | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Bio?** | **VGG-5** | **VGG-7** | **VGG-9** | **ResNet-18** | **VGG-5** | **VGG-7** | **VGG-9** | **ResNet-18** |
| BP-CE [18] (CE) | No | $88.11_{\pm 0.13}$ | $88.60_{\pm 0.10}$ | $89.18_{\pm 0.08}$ | $92.83_{\pm 0.18}$ | $60.82_{\pm 0.10}$ | $59.96_{\pm 0.10}$ | $60.63_{\pm 0.28}$ | $72.32_{\pm 0.26}$ |
| BP-SE [18] (SE) | No | $89.43_{\pm 0.12}$ | $89.91_{\pm 0.10}$ | $90.02_{\pm 0.18}$ | $93.21_{\pm 0.07}$ | $66.28_{\pm 0.23}$ | $65.36_{\pm 0.15}$ | $65.51_{\pm 0.23}$ | $71.89_{\pm 0.16}$ |
| BP (tuned) | No | $88.03_{\pm 0.18}$ | $91.39_{\pm 0.05}$ | $92.43_{\pm 0.29}$ | $91.54_{\pm 0.36}$ | $64.09_{\pm 0.14}$ | $\mathbf{69.00}_{\pm 0.30}$ | $\mathbf{71.90}_{\pm 0.32}$ | $70.22_{\pm 0.27}$ |
| DPC-CN [19] (CE) | Partial | $\mathbf{89.46}_{\pm 0.14}$ | $89.11_{\pm 0.26}$ | $86.55_{\pm 0.81}$ | — | $\mathbf{66.85}_{\pm 0.07}$ | $63.89_{\pm 0.22}$ | $60.59_{\pm 0.47}$ | — |
| iPC [18, 22] (CE) | Partial | $85.51_{\pm 0.12}$ | $80.15_{\pm 0.21}$ | $79.02_{\pm 0.21}$ | $70.44_{\pm 0.81}$ | $56.07_{\pm 0.16}$ | $43.99_{\pm 0.30}$ | $44.76_{\pm 0.40}$ | $29.45_{\pm 1.36}$ |
| ePC [7] (CE) | Partial | $88.27_{\pm 0.18}$ | $88.84_{\pm 0.31}$ | $86.81_{\pm 0.09}$ | $91.73_{\pm 0.21}$ | $63.39_{\pm 0.25}$ | $58.62_{\pm 0.20}$ | $60.65_{\pm 0.25}$ | $69.47_{\pm 0.32}$ |
| **WF-Act-PC** (ours) (CE) | **Yes** | $88.25_{\pm 0.31}$ | $\mathbf{92.15}_{\pm 0.72}$ | $\mathbf{93.57}_{\pm 0.48}$ | $\mathbf{92.76}_{\pm 0.85}$ | $65.64_{\pm 0.57}$ | $68.75_{\pm 0.64}$ | $70.32_{\pm 0.53}$ | $\mathbf{71.07}_{\pm 0.71}$ |

Both WF-Act-PC and BP (tuned) are tuned per architecture (each at its own best LR; VGG-5/CIFAR-10 BP not separately retuned—matched-recipe value shown). CIFAR-10: AdamW ($\eta_w$=$7\times10^{-4}$), Soft SNC ($\tau$=3.0), $T$=20 RMSProp, Cutout(8), 50 epochs, 5 seeds, same hyperparameters across all architectures. CIFAR-100: CutMix/Mixup, $\eta_w$=$2\times10^{-3}$ (VGG-5/7) or $1\times10^{-3}$ (VGG-9, ResNet-18), label smoothing 0.1; see supplementary. The WF-Act-PC VGG-7/9/CIFAR-100 entries above use the main CIFAR-100 recipe ($\eta_w$=$2\times10^{-3}/1\times10^{-3}$); under symmetric per-method LR tuning (matching BP's protocol, $\eta_w$=$7\times10^{-4}$) they reach $68.76_{\pm 0.31}$ / $71.06_{\pm 0.18}$, i.e. within −0.24 / −0.84 pp of tuned BP (Suppl. Mat. Sect. H).

### 5.2 Main Results: CIFAR-10

Table 1 presents the main results. Three patterns emerge. *First*, WF-Act-PC surpasses iPC—the strongest classical PC method—on every architecture, with the advantage growing monotonically with depth: the gap widens from +2.7 pp (VGG-5) to +22.3 pp (ResNet-18) on CIFAR-10. Where iPC degrades from 85.51% to 70.44%, WF-Act-PC *improves* from 88.25% to 93.57%. *Second*, with both methods tuned per architecture, WF-Act-PC meets or exceeds BP (tuned) on the deeper CIFAR-10 architectures (+0.8, +1.1, +1.2 pp on VGG-7/9/ResNet-18); this holds under the controlled same-recipe comparison—though both of the benchmark's own BP references exceed WF-Act-PC on ResNet-18 (BP-CE 92.83%, BP-SE 93.21% vs. 92.76%), reflecting the benchmark's stronger per-architecture recipe (Sect. 5.1). On VGG-5 it exceeds same-recipe BP and the benchmark's BP-CE (88.25% vs. 88.03/88.11%) but remains below BP-SE (89.43%) and DPC-CN (89.46%), indicating that the gap to a fully-tuned baseline has not closed on shallow models. To our knowledge, this is the first PC method to reach BP-level accuracy on deep convolutional architectures under a controlled comparison. *Third*, ePC [7]—which shares the error reparametrisation but uses autograd $J^\top$—degrades on deeper VGGs (88.27 → 86.81%, VGG-5→9), while

WF-Act-PC improves; local transport, not the training recipe alone, drives the depth-scaling advantage.

### 5.3 CIFAR-100, Tiny-ImageNet, and Wall-Clock Time

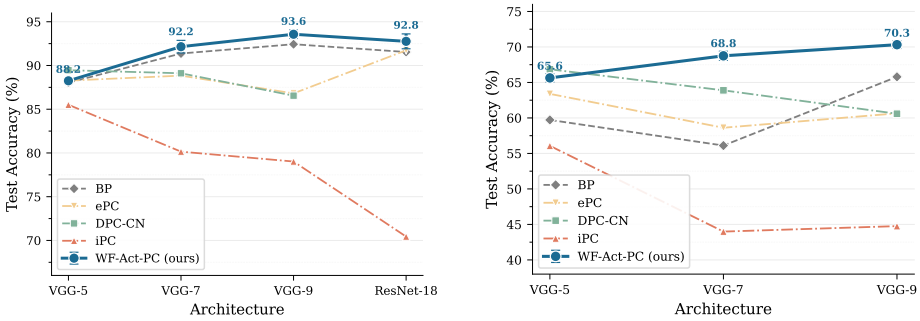
*CIFAR-100.* CIFAR-100 presents a harder test of inner-loop stability: with 100 output classes, error norms are substantially larger than on CIFAR-10. WF-Act-PC maintains monotonic depth scaling (Table 1, right; Figure 2b): 65.64% (VGG-5), 68.75% (VGG-7), 70.32% (VGG-9), 71.07% (ResNet-18). Prior PC methods either collapse with depth (iPC: 56.07 → 29.45%, VGG-5→ResNet-18) or degrade on deeper VGGs (DPC-CN: 66.85 → 60.59%; ePC: 63.39 → 60.65%, VGG-5→9). WF-Act-PC surpasses tuned BP on VGG-5 and ResNet-18 (by +1.6 and +0.9 pp). On the two deepest CIFAR-100 VGG cells, however, tuned BP catches up: under symmetric per-method LR tuning the methods tie on VGG-7 (68.76 vs. 69.00, −0.24 pp) and tuned BP leads on VGG-9 (71.06 vs. 71.90, −0.84 pp; small but statistically significant). The depth-scaling property of WF-Act-PC itself is unaffected. CIFAR-100 uses CutMix/Mixup with learning rate $2\times10^{-3}$ (VGG-5/7) or $1\times10^{-3}$ (VGG-9, ResNet-18); see supplementary.

*Scaling beyond CIFAR.* To test depth scaling on a harder benchmark, we evaluate on Tiny-ImageNet (200 classes, 64×64) with the unchanged WF-Act-PC recipe. VGG-5 reaches 46.0% and ResNet-18 reaches 61.1% top-1 (vs. per-architecture-tuned BP at 45.0% and 59.2%), a +15.1 pp gain with depth and +1.0 / +1.9 pp over tuned BP at each depth. These Tiny-ImageNet runs are *single-seed* (following our Tiny-ImageNet protocol) and should be read as a scaling data point rather than a precise benchmark; the full per-architecture BP LR grid is in Suppl. Mat. Sect. H.

*Wall-clock time.* The $T$=20 inner loop incurs ≈3× overhead relative to single-pass BP on a single A100 GPU: 22 s vs. 7 s per epoch on VGG-5, 57 s vs. 13 s on ResNet-18. This cost is inherent to iterative inference; total training for VGG-5 is ≈18 min (50 epochs). The inner loop also affords a clean accuracy–compute trade-off: even $T$=1 (a single Gauss-Seidel sweep, 6.4 s/epoch, near BP speed) reaches 87.64% on VGG-5/CIFAR-10, already above iPC (85.51%; cf. Table 2). Layer-parallel neuromorphic hardware would amortise the inner loop across cores; a detailed timing breakdown appears in Suppl. Mat. Sect. G.

### 5.4 Depth Scaling Analysis

Figure 2 visualises the central empirical finding: WF-Act-PC is the only PC method whose accuracy *increases* with depth on both datasets. Positive depth scaling is the expected consequence of correct $J^\top$ transport: deeper networks have greater representational capacity, and exact error routing lets WF-Act-PC exploit it—mirroring the behaviour of backpropagation.




**(a)** CIFAR-10 **(b)** CIFAR-100

**Fig. 2:** Depth scaling on CIFAR-10 and CIFAR-100. WF-Act-PC (solid blue) is the only PC method whose accuracy *increases monotonically* with depth on both datasets. Error bars: ±1 std over 5 seeds.

### 5.5 Ablation Study

Table 2 ablates the key design choices on VGG-5/CIFAR-10, using the same hyperparameters as Table 1 (mean ± std, 5 seeds, 50 epochs).

*Transport mechanism.* Random $B$ (FA-style) collapses to 27.42%, consistent with the known failure of FA on deep CNNs [3, 13]: structured feedback is necessary. $W^\top$ alone (without $\sigma'$) yields 81.62%; adding the $\sigma'$ correction recovers 88.25% (+6.6 pp), establishing $\sigma'$ as the critical factor in Observation 1. Importantly, both ablations run the *full* WF-Act-PC recipe (SNC, top-clip, Cutout, label smoothing), so neither the stabilisation stack nor the network topology alone explains the result: the $\mathbf{s}\odot\sigma'$ correction is essential, not incidental. Replacing local transport with autograd $J^\top$ gives 89.01%—a gap of only 0.8 pp, within seed variance. This narrow gap validates that the local formula faithfully implements $J^\top$; the value of local transport is *bio-plausibility*—eliminating the autograd dependency—rather than additional accuracy.

*Inner loop iterations.* Accuracy improves from $T$=1 (87.64%) through $T$=10 (88.29%) and plateaus at $T$=20 (88.25%). $T$=40 overshoots: RMSProp's adaptive denominator shrinks over many iterations, amplifying updates and destabilising convergence. The effective range is $T$=10–20, fewer than the $T$=30 used by iPC [18, 22]. Even $T$=1—a single Gauss-Seidel sweep at 6.4 s/epoch—achieves 87.64%, surpassing iPC (85.51%) at 3.5× lower cost.

*Sensitivity to $\tau$ and $T$.* Sweeping the SNC threshold $\tau \in \{2, 3, 5, \infty\}$ on VGG-5/CIFAR-10 (single seed) gives 87.9/88.3/87.9/87.4%: accuracy varies by $\leq 0.9$ pp across the range, including $\tau$=$\infty$ (no clipping), so WF-Act-PC is not brittle to this hyperparameter. Likewise, extending the inner loop to $T$=30 holds accuracy (88.2%, single seed); the collapse at $T$=40 (Table 2) is RMSProp overshoot, outside the recommended $T \in [10, 20]$ range. The method is therefore robust over $T \in [5, 30]$ and $\tau \in [2, \infty)$.

**Table 2:** Ablation study on VGG-5/CIFAR-10 (mean ± std over 5 seeds, same hyperparameters as Table 1).

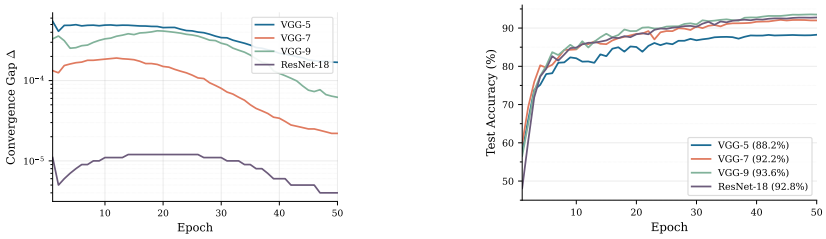
| Group | Variant | Acc (%) | s/epoch |
|---|---|---|---|
| Transport | Random $B$ (FA-style) | $27.42_{\pm 0.83}$ | 22.0 |
| | $W^\top$ only (no $\sigma'$) | $81.62_{\pm 0.47}$ | 20.1 |
| | $W^\top + \sigma'$ (**WF-Act-PC**) | $\mathbf{88.25}_{\pm 0.31}$ | 22.1 |
| | Autograd $J^\top$ (`jax.vjp`) | $89.01_{\pm 0.24}$ | — |
| Inner loop $T$ | $T = 1$ | $87.64_{\pm 0.38}$ | 6.4 |
| | $T = 5$ | $88.07_{\pm 0.29}$ | 10.8 |
| | $T = 10$ | $88.29_{\pm 0.33}$ | 15.5 |
| | $T = 20$ (default) | $\mathbf{88.25}_{\pm 0.31}$ | 22.1 |
| | $T = 40$ | $72.01_{\pm 0.91}$ | 33.2 |
| Stability | SNC disabled | $87.40_{\pm 0.52}$ | 20.6 |

**Table 3:** BN-VGG-5/CIFAR-10 ablation (mean ± std over 5 seeds): effect of each correction factor in Observation 1 when BatchNorm is present ($s \neq 1$).

| Transport variant | Acc (%) |
|---|---|
| $W^\top$ only (no $\sigma'$, no $s$) | $83.74_{\pm 0.14}$ |
| $W^\top \cdot \sigma'$ (no $s$) | $86.03_{\pm 0.22}$ |
| $W^\top \cdot s$ (no $\sigma'$) | $84.48_{\pm 0.20}$ |
| $W^\top \cdot \sigma' \cdot s$ (**full**) | $87.60_{\pm 0.23}$ |
| Autograd $J^\top$ (`jax.vjp`) | $89.82_{\pm 0.27}$ |

*Spectral norm clipping.* Disabling SNC drops accuracy by only 0.85 pp (87.40% vs. 88.25%), still exceeding iPC by +1.9 pp. SNC is a practical stabiliser, not a prerequisite: WF-Act-PC's core advantage—local $J^\top$ transport via $\sigma'$—persists without it. This matters for bio-plausibility, since SNC (power iteration) is the one non-local component beyond the symmetric synapse.

*BatchNorm correction ($s \neq 1$).* The preceding ablation uses no-BN VGGs where $s$=1, exercising only the $\sigma'$ correction. To validate the full Observation 1 including the BN gain $s = \gamma/\sqrt{\mathrm{Var} + \epsilon}$, we add BatchNorm to VGG-5 and selectively ablate each factor (Table 3). Removing $s$ drops accuracy by 1.6 pp (86.03% vs. 87.60%); removing $\sigma'$ costs 3.1 pp; removing both ($W^\top$ only) costs 3.9 pp. Autograd $J^\top$ reaches 89.82%, indicating that the residual 2.2 pp gap to autograd is attributable to the MaxPool nearest-neighbour approximation (the only non-exact component). All gaps are large relative to the seed spread ($\leq 0.3$ pp), so both correction factors in Observation 1 contribute meaningfully when $s \neq 1$.




**(a)** Convergence gap $\Delta$ **(b)** Test accuracy vs. epoch

**Fig. 3:** Training dynamics on CIFAR-10. **(a)** Convergence gap $\Delta$ (Eq. 6) drops to near-zero for all architectures, confirming $T$=20 iterations suffice. **(b)** Deeper models reach higher final accuracy, consistent with positive depth scaling (Table 1).

*MaxPool unpooling.* Our default uses nearest-neighbour unpooling—a local approximation that avoids caching argmax positions (Remark 2). Comparing against exact argmax unpooling (which scatters each error to the cached argmax position) on VGG-5/CIFAR-10 (mean $\pm$ std over 5 seeds) yields $88.32_{\pm 0.35}\%$ vs. $89.58_{\pm 0.21}\%$—a gap of only 1.25 pp, validating the local approximation. Switching to exact unpooling closes the small residual gap between local transport and autograd $J^\top$ (89.01%, Table 2), confirming that MaxPool, not the $W^\top\sigma'$ formula, is the dominant source of the local-vs-autograd gap.

*Inner-loop convergence.* Figure 3a tracks the convergence gap $\Delta$ (Eq. 6) during training with $T$=20. $\Delta$ drops to $\sim 10^{-4}$ within the first few epochs across all architectures, indicating reliable convergence. Figure 3b shows that deeper models consistently reach higher final accuracy.

## 6 Discussion and Conclusion

*Why WF-Act-PC matches or exceeds backpropagation.* On VGG-7/9/ResNet-18 (CIFAR-10), WF-Act-PC outperforms BP (tuned) by +0.8/+1.1/+1.2 pp. We tentatively attribute this to the Gauss-Seidel inner loop acting as an implicit regulariser, iteratively refining activations toward a local energy minimum that may favour generalisation—a denoising effect absent from single-pass BP.[3] That even $T$=1 surpasses iPC (87.64% vs. 85.51%, Table 2) suggests the advantage originates in the local $J^\top$ transport formula rather than the number of inner-loop iterations.

*Relation to ePC.* As discussed in Sect. 3, WF-Act-PC shares the error-based formulation and Gauss-Seidel sweep with ePC [7], paired with an RMSProp inner loop. The autograd-vs-local ablation (Table 2) quantifies the decomposition: the

[3] This is a hypothesis to explain the gap, not a claim established by controlled experiments.

shared training recipe accounts for the accuracy gains over iPC, while Observation 1 eliminates the autograd dependency at negligible accuracy cost (88.25% local vs. 89.01% autograd, within seed variance). The observation's contribution is therefore not additional accuracy but *locality*: WF-Act-PC computes $J^\top$ transport from local factors with no autograd VJP and no performance loss relative to its autograd-dependent counterpart.

*Limitations.* We restate the assumptions flagged in Sect. 1, as none is removed by this work. (i) *Weight symmetry*: $L^\top$ is assumed to mirror the feedforward operator—shared with all PC methods and BP alike; combining WF-Act-PC with learned approximate feedback [1] remains open. (ii) *Soft SNC*: power iteration is not synapse-local, though it operates on the slow learning timescale and disabling it costs only 0.85 pp (Table 2). (iii) *MaxPool*: the nearest-neighbour unpool is a local approximation (≈1.25 pp vs. exact argmax), so Eq. (4) is exact only for the non-pooling portion of the network. Beyond these, the $T$=20 inner loop adds ≈3× wall-clock overhead (Sect. 5.3), inherent to iterative inference, and our evaluation follows the PCX benchmark protocol on CIFAR-10/100 with a Tiny-ImageNet scaling check—ImageNet-1K, deeper ResNets, and attention-based architectures via frozen LayerNorm (Suppl. Mat. Sect. B) are natural next steps. Finally, the picture against backpropagation is not uniformly favourable: on VGG-9/CIFAR-100, symmetrically-tuned BP exceeds WF-Act-PC by 0.84 pp (Sect. 5.3), and the benchmark's own BP references (BP-CE/BP-SE)—which use a stronger per-architecture recipe—exceed WF-Act-PC on ResNet-18 (both datasets).

*Conclusion.* We showed that weight feedback combined with two locally computable corrections—$\sigma'$(pre-act) and $\mathbf{s} = \gamma/\sigma_{\text{run}}$—computes the exact Jacobian transpose for Act(Norm($L(\cdot)$)) layers, up to the weight-symmetry, soft-SNC, and MaxPool caveats above. Neither the FA nor the PC community had put these pieces together. Substituting this identity into PC yields WF-Act-PC, which surpasses iPC (by up to +22.3 pp) and reaches BP-level accuracy on deep convolutional architectures under per-architecture tuning—the only PC method whose accuracy improves with depth. Weight feedback did not fail because $W^\top$ was used; it failed because $\sigma'$ and $\gamma/\sigma$ were left out.

**Acknowledgements.** This work was supported in part by the Canada CIFAR AI Chairs Program, the Alberta Machine Intelligence Institute (Amii), and the Natural Sciences and Engineering Research Council of Canada (NSERC).

**Disclosure of Interests.** The authors declare that they have no competing interests.

## References


1. Akrout, M., Wilson, C., Humphreys, P.C., Lillicrap, T.P., Tweed, D.B.: Deep learning without weight transport. In: Advances in Neural Information Processing Systems (NeurIPS) (2019)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Bartunov, S., Santoro, A., Richards, B., Marris, L., Hinton, G.E., Lillicrap, T.: Assessing the scalability of biologically-motivated deep learning algorithms and architectures. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
4. Carandini, M., Heeger, D.J.: Normalization as a canonical neural computation. Nature Reviews Neuroscience **13**(1), 51–62 (2012)
5. Crick, F.: The recent excitement about neural networks. Nature **337**(6203), 129–132 (1989)
6. Friston, K.: A theory of cortical responses. Philosophical Transactions of the Royal Society B **360**(1456), 815–836 (2005)
7. Goemaere, C., Oliviers, G., Bogacz, R., Demeester, T.: ePC: Fast and deep predictive coding in digital simulation. arXiv preprint arXiv:2505.20137 (2025)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
9. Innocenti, F., Achour, E.M., Buckley, C.L.: $\mu$PC: Scaling predictive coding to 100+ layer networks. In: Advances in Neural Information Processing Systems (NeurIPS) (2025)
10. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (ICML) (2015)
11. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
12. Kunin, D., Nayebi, A., Sagastuy-Brena, J., Ganguli, S., Bloom, J.M., Yamins, D.L.K.: Two routes to scalable credit assignment without weight symmetry. In: International Conference on Machine Learning (ICML) (2020)
13. Launay, J., Poli, I., Boniface, F., Krzakala, F.: Direct feedback alignment scales to modern deep learning tasks and architectures. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
14. Lee, D.H., Zhang, S., Fischer, A., Bengio, Y.: Difference target propagation. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD) (2015)
15. Lillicrap, T.P., Cownden, D., Tweed, D.B., Akerman, C.J.: Random synaptic feedback weights support error backpropagation for deep learning. Nature Communications **7**, 13276 (2016)
16. Millidge, B., Tschantz, A., Buckley, C.L.: Predictive coding approximates backprop along arbitrary computation graphs. Neural Computation **34**(6), 1329–1368 (2022)
17. Nøkland, A.: Direct feedback alignment provides learning in deep neural networks. In: Advances in Neural Information Processing Systems (NeurIPS) (2016)
18. Pinchetti, L., Qi, C., Lokshyn, O., Oliviers, G., Emde, C., Tang, M., M'Charrak, A., Frieder, S., Menzat, B., Bogacz, R., Lukasiewicz, T., Salvatori, T.: Benchmarking predictive coding networks — made simple. In: International Conference on Learning Representations (ICLR) (2025)

19. Qi, C., Lukasiewicz, T., Salvatori, T.: Training deep predictive coding networks. In: New Frontiers in Associative Memories Workshop, ICLR (2025)
20. Rao, R.P.N., Ballard, D.H.: Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience **2**(1), 79–87 (1999)
21. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
22. Salvatori, T., Song, Y., Yordanov, Y., Millidge, B., Emde, C., Xu, Z., Sha, L., Bogacz, R., Lukasiewicz, T.: A stable, fast, and fully automatic learning algorithm for predictive coding networks. In: International Conference on Learning Representations (ICLR) (2024)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2015)
24. Turrigiano, G.G., Nelson, S.B.: Homeostatic plasticity in the developing nervous system. Nature Reviews Neuroscience **5**(2), 97–107 (2004)
25. Whittington, J.C.R., Bogacz, R.: An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. Neural Computation **29**(5), 1229–1262 (2017)
26. Wu, Y., He, K.: Group normalization. In: European Conference on Computer Vision (ECCV) (2018)