LEARNING TO BACKPROPAGATE VIA COUPLED DIRECT FEEDBACK DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Direct feedback alignment (DFA) has been proposed as a local-learning alternative to backpropagation (BP), projecting errors directly to each layer via fixed, randomly initialized feedback connections, thereby overcoming the backward locking imposed by BP's sequential backward pass. However, DFA exhibits limited performance compared to BP, primarily because its feedback connections remain fixed and cannot adapt to the evolving forward connections, thus reducing credit-assignment accuracy in deeper networks. In this work, we present direct feedback propagation (DFP), which addresses this limitation by introducing local losses on the feedback connections that induce learning dynamics whose time-varying equilibrium coincides with the chain of transposed forward weights. We theoretically and empirically demonstrate that, under a separation of timescales between fast feedback dynamics and slower forward dynamics, the feedback connections track this equilibrium throughout training. This yields exact BP gradients in the linear regime and accurate approximations in the nonlinear one, resulting in improved gradient alignment compared to DFA. Furthermore, unlike recent feedback-learning rules, DFP's feedback update is independent of the forward pass and can run fully in parallel with it. We show that DFP achieves the best performance among direct feedback methods across all evaluated settings, surpassing the best method by margins ranging from 4.9% absolute accuracy on ResNet-20 to more than 9.8% on VGG-9 and VGG-11 networks, while achieving update locality and enabling training speedups compared to BP.

1 INTRODUCTION

Backpropagation (BP) of the error (Linnainmaa, 1976; Rumelhart et al., 1986; Werbos, 1988) is the current standard for training deep neural networks, enabling impressive results on several domains, ranging from natural language processing (Hochreiter & Schmidhuber, 1997; Vaswani et al., 2017) to computer vision (Cireşan et al., 2013; Simonyan & Zisserman, 2015; He et al., 2016). However, BP depends on a global error signal that has to be sequentially propagated backward throughout the network's layers, as illustrated in Figure 1(A). This has raised concerns both about its direct implementation in the brain (Whittington & Bogacz, 2019; Lillicrap & Santoro, 2019; Ellenberger et al., 2025), and about its efficiency for hardware implementations (Mostafa et al., 2018; Frenkel et al., 2023). In particular, the backward pass of BP relies on the exact transpose of the forward weight matrices to propagate error signals, a requirement known as the *weight transport* problem. In addition, its strictly sequential nature causes *backward locking*, forcing each layer to wait for the gradient signals of all subsequent layers before updating. These problems are often cited as a primary source of biological implausibility for BP (Lillicrap et al., 2014; Nøkland, 2016; Akrout et al., 2019; Ororbia, 2023). Furthermore, its locking mechanism prevents effective layer-wise parallelism and results in a computational bottleneck in deep architectures (Jaderberg et al., 2017; Ororbia, 2023; Frenkel et al., 2023).

Direct feedback alignment (DFA) has been proposed as an alternative algorithm to train deep neural networks, directly addressing both the weight transport and backward locking issues (Nøkland, 2016). Rather than sequentially propagating gradients, DFA employs fixed, random feedback matrices to directly project the output error to all hidden layers, as shown in Figure 1(B). This shift in the error propagation mechanism results in a local learning rule for the forward weights. By relying on the direct error projection, each layer updates independently, using only neural activity information

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.

from adjacent neurons and the error signal delivered by its feedback connection. However, DFA scales poorly to deeper architectures and more complex datasets, lagging far behind BP in classification performance (Webster et al., 2021; Caillon et al., 2026). This limitation is mainly due to its fixed feedback connections, which cannot adapt to better guide the updates of the evolving forward weights during learning.

In this work, we introduce direct feedback propagation (DFP), using the forward weights to locally learn feedback connections while preserving backward-locking-free forward weight updates. This is illustrated in Figure 1(C), where, similarly to DFA, DFP's feedback matrices deliver gradient proxies to the forward weights in parallel. Layer-wise losses drive each feedback matrix to match the product of the transposed forward weights and the feedback matrix of the next layer. This induces coupled learning dynamics where feedback connections evolve on a faster timescale and track a time-varying equilibrium corresponding to the slowly evolving chain of transposed forward weights. In the linear regime, that is, when the activation functions reduce to the identity, this equilibrium yields the exact gradient mapping of BP. In the nonlinear regime, when nonlinear activation functions are applied, it approximates that mapping up to the local activation Jacobians. Notably, DFP does not strictly reintroduce weight transport, as the forward weights are never directly involved in propagating the error signal, which is always delivered through the learned feedback matrices. Each forward weight enters only the local update of its corresponding feedback matrix, preserving locality in both forward and feedback weight updates. The result is a direct feedback algorithm that scales to deeper networks and more complex datasets significantly better than DFA. Our contributions are summarized as follows:

- We introduce DFP, a direct feedback algorithm coupling feedback and forward weight updates, resulting in feedback learning dynamics whose equilibrium corresponds to the chain of transposed forward weights. We prove that, under a separation of timescales, the feedback connections track this equilibrium, recovering exact BP gradients in the linear regime and approximating them in the nonlinear one (Section 3, Appendix A.1).
- We demonstrate that DFP achieves the best performance among direct feedback methods across all evaluated architectures and datasets, narrowing the gap with BP while achieving local and parallelizable learning (Section 4).
- We benchmark a multi-GPU implementation of DFP against an optimized data-parallel BP baseline, showing that DFP trains faster in almost all evaluated configurations, with wall-clock speedups of up to $2.4\times$ on four H100 GPUs (Section 4).

2 Background and Related Works

2.1 Backpropagation

Consider a feedforward network $f_\Theta : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ of depth L, parameterized by weight matrices $\Theta_\ell \in \mathbb{R}^{d_{\ell+1}\times d_\ell}$, $\ell \in \{0, \dots, L-1\}$, with an element-wise activation ϕ and a linear output layer, that is, with ϕ corresponding to the identity function at the last layer, as illustrated in Figure 1(A). Each layer computes the pre-activation $\hat{x}_\ell = \Theta_{\ell-1} x_{\ell-1}$ and the activity $x_\ell = \phi(\hat{x}_\ell)$, so that the forward pass is given by

$$x_L = \Theta_{L-1}\,\phi\Big(\Theta_{L-2}\,\phi\big(\cdots\,\phi(\Theta_1\,\phi(\Theta_0 x_0))\cdots\big)\Big), \tag{1}$$

with bias terms kept implicit for notational simplicity. Given a target $y \in \mathbb{R}^{d_L}$, we consider the squared-error loss

$$\mathcal{L} = \frac{1}{2}\|x_L - y\|_2^2. \tag{2}$$

Backpropagation computes the loss gradients with respect to the pre-activations $\delta_\ell = \partial\mathcal{L}/\partial\hat{x}_\ell$ through the following recursion:

$$\delta_\ell = \begin{cases} x_L - y & \text{if } \ell = L, \\ \phi'(\hat{x}_\ell) \odot \Theta_\ell^\top \delta_{\ell+1} & \text{otherwise} \end{cases} \quad\Longleftrightarrow\quad \delta_\ell = \left(\prod_{i=\ell}^{L-1} \phi'(\hat{x}_i) \odot \Theta_i^\top\right) \delta_L, \tag{3}$$

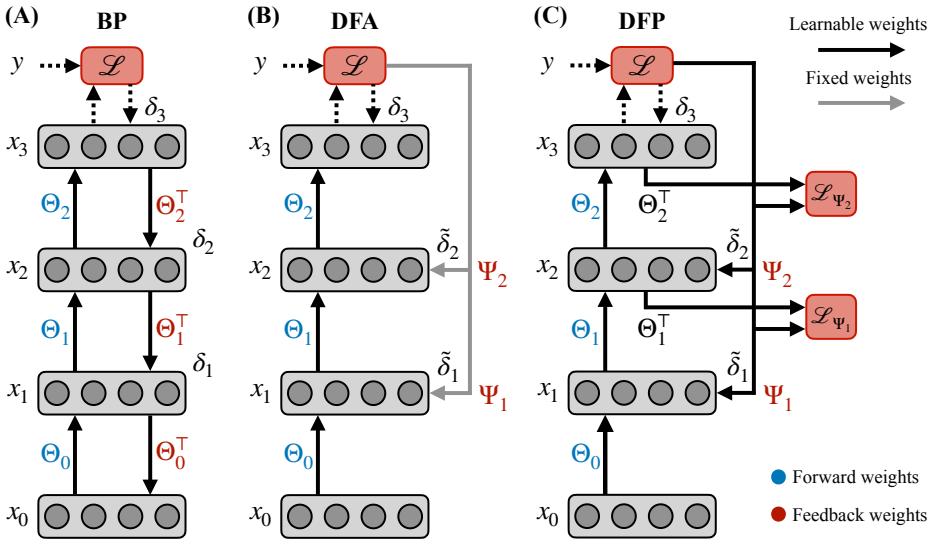

Figure 1: Schematic representations of different neural credit assignment algorithms. Red squares at the top of each schematic represent the global loss. (A) The iterative, sequential feedback of back-propagation (BP). (B) Direct feedback alignment (DFA), which addresses both the backward locking and weight transport issues inherent to BP through direct feedback connections. (C) The proposed direct feedback propagation (DFP) method, which reintroduces weight transposes and leverages them to learn the feedback matrices via the local losses $\mathcal{L}_{\Psi_\ell}$, represented by the red squares to the right of the hidden layers. The resulting learned pathways converge to linear-exact and nonlinear-approximate BP feedback pathways.

where $\odot$ denotes the element-wise product between the diagonal of the activation Jacobian and the transposed weight matrix[1]. The parameters are then updated by gradient descent,

$$\Delta\Theta_\ell = -\eta_\Theta \left(\delta_{\ell+1} x_\ell^\top\right), \tag{4}$$

with forward learning rate η_Θ.

2.2 Direct Feedback Alignment

DFA introduces fixed random feedback matrices, denoted by $\Psi_\ell \in \mathbb{R}^{d_\ell \times d_L}$, directly connecting the output layer to every hidden layer, as illustrated in Figure 1(B). Instead of iteratively propagating the gradient backward from the output of the network, as done by BP (see Equation 3), DFA directly computes the pre-activation neural activity gradient using a BP gradient surrogate

$$\tilde{\delta}_\ell = \phi'(\hat{x}_\ell) \odot (\Psi_\ell \delta_L), \tag{5}$$

with $\Psi_L = I$, resulting in the following update rule for the forward weights:

$$\Delta\Theta_\ell = -\eta_\Theta \left(\tilde{\delta}_{\ell+1} x_\ell^\top\right). \tag{6}$$

This update rule is local, as all the information required for the weight update comes from layers that are now directly connected, thanks to the feedback weights Ψ_ℓ.

2.3 Related Work

Beyond DFA's fixed random feedback, other methods learn the feedback connections locally. Direct Kolen–Pollack (DKP) (Webster et al., 2021) proposes to learn the feedback matrices with an update rule inspired by the works of Kolen & Pollack (1994) and Akrout et al. (2019), relying on the neural

[1]Since the activation functions are applied element-wise, their Jacobians are diagonal and can be represented by the vector $\phi'(\hat{x}_\ell) \in \mathbb{R}^{d_\ell}$ of their diagonal entries. We use the Hadamard product with a matrix to specify element-wise multiplication with each of the matrix's columns.

activity and the output error. Interestingly, in the linear case, this approach drives the feedback connections toward the chain of transposed forward weights, as in DFP's feedback dynamics, but with the recursion distorted by the Moore–Penrose pseudoinverse (Casnici et al., 2026). A more recent approach, namely Gradient-Aligned Projected Error (GrAPE) (Caillon et al., 2026), computes a rank-1 estimate of the Jacobian of the network output with respect to each layer's activity via forward-mode Jacobian-vector products (JVPs), and learns each feedback matrix by minimizing a local cosine-similarity loss with the corresponding estimated Jacobian. Local alignment support (LAS) modules (Yang et al., 2024) instead learn the feedback matrices as the transposes of single-layer local networks trained to reproduce the network output from each layer's activity. In contrast to DFP, all three updates depend on the neural activities, and thus on the network's forward pass. A separate line of work is taken by direct random target projection (DRTP) (Frenkel et al., 2021), which keeps the feedback connections fixed, as in DFA, but projects the target vector rather than the output gradient. This yields a more efficient but less scalable learning rule that applies only to supervised classification, where the target serves as a proxy for the sign of the error. Complementary to these methods, sparse connections (Crafton et al., 2019), binary matrices (Han & Yoo, 2019), and low-rank factorizations (Roy et al., 2026) have been proposed to improve memory and computational efficiency of the feedback pathway itself, forming directions orthogonal to ours.

3 METHODOLOGY

Considering Equation 3 in the linear case, that is, with ϕ being the identity function, we define the linear BP mapping from the output gradient δ_L to the gradient δ_ℓ of an arbitrary layer ℓ as

$$\Psi_\ell^\star = \prod_{i=\ell}^{L-1} \Theta_i^\top \quad \Longleftrightarrow \quad \Psi_\ell^\star = \Theta_\ell^\top \Psi_{\ell+1}^\star, \tag{7}$$

where at the last layer we have $\Psi_L^\star = I$, with I being the identity matrix. Since a composition of linear maps can be expressed with a single matrix, each feedback connection Ψ_ℓ can, in principle, represent the entire chain $\Psi_\ell^\star$. We exploit this property to address DFA's primary limitation, letting the otherwise fixed feedback connections track the evolving forward weights and thereby match the feedback pathway of linear BP exactly. To this end, we introduce layer-wise loss functions $\mathcal{L}_{\Psi_\ell}$ on the feedback matrices, as illustrated in Figure 1(C), defined as

$$\mathcal{L}_{\Psi_\ell} = \frac{1}{2} \|\Psi_\ell - \Theta_\ell^\top \Psi_{\ell+1}\|_F^2, \tag{8}$$

where the feedback matrix at the output is fixed to the identity matrix $\Psi_L = I$. Importantly, the squared Frobenius norm loss $\mathcal{L}_{\Psi_\ell}$ considers only its own Ψ_ℓ as the free variable, so that no gradient propagates to the other layers. By letting the feedback matrices' dynamics evolve to minimize their layer-wise loss $\mathcal{L}_{\Psi_\ell}$, we obtain DFP's feedback weight update rule:

$$\Delta \Psi_\ell = -\eta_\Psi \nabla_{\Psi_\ell} \mathcal{L}_{\Psi_\ell}, \tag{9}$$

where η_Ψ corresponds to the feedback learning rate, and the gradient $\nabla_{\Psi_\ell} \mathcal{L}_{\Psi_\ell}$ is equal to

$$\frac{\partial \mathcal{L}_{\Psi_\ell}}{\partial \Psi_\ell} = \Psi_\ell - \Theta_\ell^\top \Psi_{\ell+1}. \tag{10}$$

This gradient involves only the weights Θ_ℓ, Ψ_ℓ, and $\Psi_{\ell+1}$, and no activity or error information, making the feedback update independent of the forward pass and executable in parallel with it. As Equation 8 is strictly convex in Ψ_ℓ, setting its gradient equal to zero yields the unique minimizer

$$\nabla_{\Psi_\ell} \mathcal{L}_{\Psi_\ell} = 0 \quad \Longleftrightarrow \quad \Psi_\ell = \Theta_\ell^\top \Psi_{\ell+1}. \tag{11}$$

Therefore, minimizing Equation 8 at all layers results in a joint stationary state for the feedback matrices coinciding with the linear BP pathway defined in Equation 7:

$$\Psi_\ell = \Psi_\ell^\star. \tag{12}$$

However, Equation 7 defines an arbitrarily long chain of linear mappings, whose singular value spectrum can degenerate with depth, with many values vanishing and a tail growing large (Saxe et al., 2014). As a consequence, the errors transported to each layer may vanish or explode in magnitude.

Normalizing the feedback matrices directly would drive them away from their equilibrium, compromising the linear-exact alignment with the BP pathway. Therefore, since the feedback pathway inherits its conditioning from the forward weights, which evolve following Equation 6, we implicitly regularize the former by driving the latter toward orthogonality through the additional local loss

$$\mathcal{L}_\perp = \sum_{\ell=0}^{L-1} \|\Theta_\ell^\top \Theta_\ell - I\|_F^2. \tag{13}$$

This loss corresponds to a layer-wise squared Frobenius penalty on the deviation from column orthogonality of the forward weights (Massart, 2022). By constraining the singular values of each factor, and therefore those of the entire chain, to remain close to one, this loss implicitly conditions the feedback equilibrium at its source, as also illustrated in Appendix A.2.3.

Since the forward weights Θ_ℓ are also updated during training, the feedback system evolves in a non-autonomous setting. To support later analysis of the dynamics, we take the gradient-flow limit of the discrete updates, in which the learning rates act as infinitesimal time constants. We therefore consider the joint dynamics yielding the coupled system:

$$\begin{aligned} \dot{\Psi}_\ell &= -\eta_\Psi \left(\Psi_\ell - \Theta_\ell^\top \Psi_{\ell+1}\right), \\ \dot{\Theta}_\ell &= -\eta_\Theta \left(\phi'(\hat{x}_{\ell+1}) \odot (\Psi_{\ell+1}\delta_L)\, x_\ell^\top + \eta_\perp \nabla_{\Theta_\ell}\mathcal{L}_\perp\right), \end{aligned} \tag{14}$$

where $\nabla_{\Theta_\ell}\mathcal{L}_\perp$ is the orthogonality loss gradient, and $\eta_\perp$ is an additional scaling factor to modulate its contribution to the update. This coupled system fully specifies DFP, combining the feedback learning rule, the DFA-style forward update, and the orthogonality regularization.

When considering Θ_ℓ evolving over time, the equilibrium in Equation 12 becomes time-dependent. In this setting, a first condition for convergence is that the system exhibits bounded forward weights and gradients[2], ensuring that the evolution of the forward parameters remains controlled, as arbitrarily large values or updates cannot be tracked by finite feedback matrices. Additionally, a separation of timescales is required, meaning that the feedback matrices have to evolve significantly faster (ideally infinitely faster) than the forward weights, so that the feedback dynamics can track the time-varying equilibrium defined by the moving forward parameters.

Theorem 1 (Convergence of the feedback dynamics)**.** *Let $\{\Theta_\ell(t)\}_{\ell=0}^{L-1}$ and $\{\Psi_\ell(t)\}_{\ell=1}^{L-1}$ denote the forward and feedback weights of a fully-connected network of finite depth L trained via DFP, or equivalently of a convolutional network under the block-Toeplitz representation (d'Ascoli et al., 2019), evolving according to Equation 14. Assume there exist finite constants A_ℓ, B_ℓ such that, uniformly in $t \geq 0$,*

$$\begin{aligned} \|\Theta_\ell(t)\|_2 &\leq A_\ell, \quad \forall \ell \in \{0, \dots, L-1\}, \\ \left\|\tilde{\nabla}_{\Theta_\ell}\mathcal{L}^\top \Psi^\star_{\ell+1}\right\|_2 &\leq B_\ell, \quad \forall \ell \in \{1, \dots, L-1\}, \end{aligned} \tag{15}$$

where $\tilde{\nabla}_{\Theta_\ell}\mathcal{L}$ denotes the forward weight update of Equation 14, and $\Psi_L = I$. Then, in the limits of infinite training time $T \to \infty$ and infinite timescale separation $\varphi = \eta_\Theta/\eta_\Psi \to 0$, the feedback matrices converge to the chain of transposed forward weights $\Psi^\star_\ell$ defined in Equation 7:

$$\lim_{\varphi \to 0} \lim_{T \to \infty} \left\|\Psi_\ell(T) - \Psi^\star_\ell(T)\right\|_2 = 0, \quad \forall \ell \in \{1, \dots, L-1\}, \tag{16}$$

and the feedback pathway thus attains exact asymptotic alignment with the BP mapping in the linear regime.

Appendix A.1 provides a complete convergence analysis. We first show that the coupled dynamics constitute a singular perturbation problem, to which Tikhonov's theorem applies. We then develop the proof constructively, where the argument is first established explicitly for the final three layers and subsequently generalized to an arbitrary layer ℓ. The proof demonstrates that, in the limit of infinitely fast feedback dynamics and infinite time, the tracking error converges to zero, regardless of the bounded drift in the forward weights.

The pseudocode of DFP is shown in Algorithm 1. The feedback parameters update stage (1) can be parallelized across all layers and run independently of the subsequent two phases. The forward pass

[2]Throughout the main text and the appendix, when referring to forward weight updates in the context of direct feedback, we use simply *gradient* as shorthand for the feedback-projected gradient proxy.

Algorithm 1 Direct Feedback Propagation

for each $(x, y) \in \mathcal{D}$ **do**
  1) Feedback parameters update *(parallel)*
  for $\ell = L - 1$ to $\ell = 1$ **do**
    $\Psi_\ell \leftarrow \Psi_\ell - \eta_\Psi \left(\Psi_\ell - \Theta_\ell^\top \Psi_{\ell+1}\right)$
  end for
  2) Forward pass *(sequential)*
  for $\ell = 1$ to $\ell = L - 1$ **do**
    $x_\ell \leftarrow \phi(\Theta_{\ell-1} x_{\ell-1})$
  end for
  $x_L \leftarrow \Theta_{L-1} x_{L-1}$
  3) Forward parameters update *(parallel)*
  $\delta_L \leftarrow x_L - y$
  $\Theta_{L-1} \leftarrow \Theta_{L-1} - \eta_\Theta \left(\delta_L x_{L-1}^\top + \eta_\perp \nabla_{\Theta_{L-1}} \mathcal{L}_\perp\right)$
  for $\ell = L - 2$ to $\ell = 0$ **do**
    $\Theta_\ell \leftarrow \Theta_\ell - \eta_\Theta \left(\phi'(\Theta_\ell x_\ell) \odot (\Psi_{\ell+1} \delta_L)\, x_\ell^\top + \eta_\perp \nabla_{\Theta_\ell} \mathcal{L}_\perp\right)$
  end for
end for

stage (2) is unchanged from the standard formulation, sequentially propagating the input to produce the network's prediction. Lastly, the forward parameters update stage (3) can also be executed in parallel for all layers, enabling simultaneous gradient delivery throughout the network.

4 RESULTS

In this section, we start by analyzing DFP's gradient alignment with BP. We then evaluate DFP on classification experiments of increasing complexity, comparing it against the baselines BP and DFA, the current state-of-the-art (GrAPE), the theoretically closest method (DKP), whose feedback dynamics converge to a fixed point similar to DFP's, and DRTP, which represents the fixed, most lightweight end of the direct feedback spectrum. Lastly, we empirically demonstrate the parallelization advantages of DFP over BP.

4.1 GRADIENT ALIGNMENT

While Theorem 1 assumes constant upper bounds on quantities such as projected gradient magnitudes, the latter typically decay as training proceeds. Therefore, in practice, feedback dynamics effectively faster than the forward ones and a finite number of update steps suffice to keep the tracking error small. This is empirically demonstrated in Figure 2, which illustrates the cosine similarity and magnitude difference between the BP gradients and those of DFP (blue) and DFA (red), in the linear (dark lines) and nonlinear (light lines) regimes, across four hidden layers of a fully-connected network with learning rates $\eta_\Psi = 3 \cdot 10^{-4}$ and $\eta_\Theta = 1 \cdot 10^{-5}$. In the linear regime, DFP reaches perfect alignment with BP within a finite number of steps across all layers. When a ReLU activation is introduced, the BP gradient mapping additionally depends on the activation Jacobians. Consequently, DFP can only approximate BP gradients in the nonlinear setting, as shown by a cosine similarity below one in Figure 2, while still achieving superior alignment relative to DFA, whose alignment must instead emerge from the forward weights adapting toward the fixed feedback connections (Refinetti et al., 2021). Lastly, DFP's gradient magnitude difference from BP decreases significantly faster and to lower values than that of DFA, showing that DFP not only aligns better with the direction of the true gradient, but also more closely matches its scale.

4.2 CLASSIFICATION EXPERIMENTS

To ensure reproducibility of our experiments, the training details, architecture specifications, and best hyperparameters of DFP are extensively reported in Appendix A.2. All implementations are based on the PyTorch framework and will be made available on GitHub.[3]

[3]Link omitted to preserve the double-blind review process.

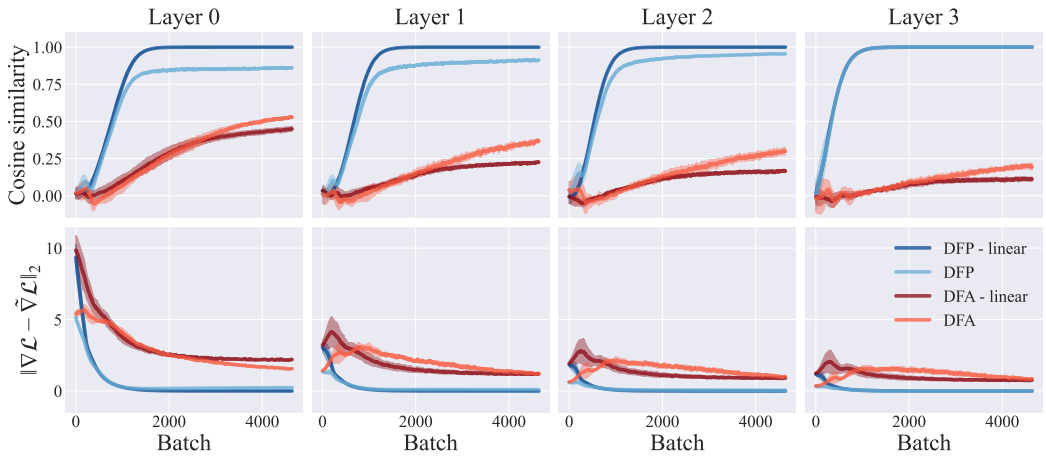

Figure 2: Gradient analysis for DFA (red) and DFP (blue), in linear (dark) and nonlinear (light) settings, averaged over 5 seeds for a 5-layer MLP trained on MNIST, with the orthogonality loss disabled to avoid biasing the comparison. The top row shows the gradient cosine similarity with BP, where DFP reaches perfect alignment in the linear regime and stays consistently above DFA in the nonlinear one. The bottom row shows the batch-averaged L2 norm of the difference between the gradient delivered by each method and the one computed by BP. DFP's gradient difference decreases faster than DFA's in both regimes, approaching zero in every layer.

Table 1: Test accuracy in % (mean ± standard deviation) averaged over 5 seeds across multiple datasets. The best result among direct feedback algorithms is in bold.

% Accuracy		MNIST		Fashion-MNIST		CIFAR-10	
Method	Local update	MLP-4	VGG-7	MLP-4	VGG-7	MLP-4	VGG-7
BP	×	$98.75^{\pm 0.08}$	$99.59^{\pm 0.05}$	$90.71^{\pm 0.11}$	$93.61^{\pm 0.20}$	$57.66^{\pm 0.38}$	$87.17^{\pm 0.25}$
DFA	✓	$98.31^{\pm 0.08}$	$99.02^{\pm 0.09}$	$89.49^{\pm 0.19}$	$91.43^{\pm 0.20}$	$46.39^{\pm 1.25}$	$74.17^{\pm 0.37}$
DRTP	✓	$95.67^{\pm 0.03}$	$98.99^{\pm 0.07}$	$86.81^{\pm 0.14}$	$90.34^{\pm 0.16}$	$37.22^{\pm 0.29}$	$56.40^{\pm 0.46}$
DKP	✓	$98.27^{\pm 0.14}$	$99.06^{\pm 0.07}$	$89.86^{\pm 0.16}$	$91.94^{\pm 0.27}$	$51.80^{\pm 1.07}$	$78.01^{\pm 0.43}$
GrAPE	✓	$98.28^{\pm 0.05}$	$99.37^{\pm 0.04}$	$89.96^{\pm 0.12}$	$92.68^{\pm 0.19}$	$55.82^{\pm 0.13}$	$79.04^{\pm 0.25}$
DFP	✓	$\mathbf{98.67^{\pm 0.08}}$	$\mathbf{99.49^{\pm 0.04}}$	$\mathbf{90.44^{\pm 0.11}}$	$\mathbf{93.07^{\pm 0.11}}$	$\mathbf{56.25^{\pm 0.39}}$	$\mathbf{83.66^{\pm 0.22}}$

Table 1 compares the results of BP, DFA, DRTP, DKP, GrAPE, and DFP for a 4-layer MLP and a VGG-7-like network, averaged across 5 independent seeds. On MNIST and Fashion-MNIST, all methods perform similarly, except DRTP, which falls slightly behind the others in both the MLP and VGG experiments. On CIFAR-10, DFA, DRTP, and DKP start to show an increased performance gap compared to BP, falling behind by more than 5% absolute accuracy on the MLP. In contrast, DFP and GrAPE show greater resilience, falling behind BP by less than 2% in absolute accuracy. When moving to the VGG-7 network, the gap with BP increases further, with DFA and DRTP underperforming BP by the largest margins, while DKP and GrAPE lag behind by 9 and 8 percentage points, respectively. DFP exhibits the strongest resilience, with a gap of only 3.5 percentage points compared to BP, and outperforms the other direct feedback algorithms in all evaluated scenarios.

In Table 2, we then evaluate DFP against its baseline (DFA), the other best-performing direct feedback algorithm (GrAPE), and BP, on more challenging scenarios spanning VGG-9, VGG-11, and ResNet-20, on CIFAR-100 and Tiny ImageNet. The increased complexity severely affects DFA, highlighting its scaling limitation. On the VGG networks, while GrAPE partially recovers this gap, DFP remains the strongest direct feedback method, lagging only 4 and 7 percentage points behind BP on CIFAR-100 and reaching similar performance on Tiny ImageNet. On ResNet-20, where DFP's approximation of the nonlinear BP chain starts to compound, it still outperforms DFA and GrAPE by more than 16.5 and 4.5 percentage points, respectively, on both datasets. Overall, via its coupled forward–feedback weight dynamics, DFP learns to backpropagate a more informative error signal, consistently achieving the highest classification accuracy among direct feedback methods.

Table 2: Test accuracy in % (mean ± standard deviation) averaged over 5 seeds across multiple datasets. The best result among direct feedback algorithms is in bold.

% Accuracy	CIFAR-100			Tiny ImageNet		
Method	VGG-9	VGG-11	ResNet-20	VGG-9	VGG-11	ResNet-20
BP	$64.62^{\pm 0.14}$	$66.93^{\pm 0.17}$	$68.04^{\pm 0.12}$	$47.88^{\pm 0.25}$	$49.26^{\pm 0.73}$	$52.94^{\pm 0.31}$
DFA	$41.41^{\pm 0.88}$	$38.78^{\pm 0.75}$	$18.48^{\pm 0.69}$	$27.51^{\pm 0.42}$	$26.05^{\pm 0.13}$	$11.41^{\pm 0.55}$
GrAPE	$49.36^{\pm 0.22}$	$47.87^{\pm 0.30}$	$35.85^{\pm 0.57}$	$37.65^{\pm 0.31}$	$35.23^{\pm 0.44}$	$22.54^{\pm 1.11}$
DFP	$\mathbf{60.92^{\pm 0.28}}$	$\mathbf{59.63^{\pm 0.63}}$	$\mathbf{40.78^{\pm 0.71}}$	$\mathbf{47.50^{\pm 0.25}}$	$\mathbf{48.71^{\pm 0.31}}$	$\mathbf{27.92^{\pm 0.94}}$

Table 3: Top-1 test accuracy (%) and training time per epoch (seconds) for a VGG-9 network trained on CIFAR-100, at varying update intervals for $\mathcal{L}_\perp$ (mean ± standard deviation over 10 runs).

$\mathcal{L}_\perp$ update interval	1	3	5	10	20	50	∞
$\eta_\perp$	0.08	0.32	0.97	1.35	2.52	2.91	0
Top-1 accuracy	$60.67^{\pm 0.31}$	$60.60^{\pm 0.25}$	$60.84^{\pm 0.23}$	$60.93^{\pm 0.43}$	$60.85^{\pm 0.43}$	$60.45^{\pm 0.27}$	$49.22^{\pm 0.37}$
Time per epoch	$10.91^{\pm 0.04}$	$8.08^{\pm 0.01}$	$7.77^{\pm 0.01}$	$7.50^{\pm 0.01}$	$7.31^{\pm 0.02}$	$7.30^{\pm 0.06}$	$7.10^{\pm 0.01}$

4.3 Parallelization experiments

Since the orthogonality loss is the primary computational bottleneck of DFP due to its cubic complexity, we first show in Table 3 that increasing the interval at which we apply this orthogonality loss does not degrade performance. These results concern a VGG-9 network trained on CIFAR-100, under the same training setting as in Table 2. The update interval can be extended even to one update every 20 batches without degrading performance, when compensated by an increased $\eta_\perp$. In the case of an ∞ interval, that is, when no orthogonality loss is applied, the network feedback pathway is no longer well-conditioned, resulting in poorer performance. Notably, despite the lack of normalization in the feedback pathway, DFP performance is still on par with GrAPE in the same setting, which applies normalization in its feedback matrices. Appendix A.2.3 shows how the orthogonality loss conditions the singular-value spectrum of the feedback pathway, and that a more uniform spectrum results in projected gradients with more balanced magnitudes across layers.

To demonstrate that parallelizability can lead to concrete wall-clock-time gains, we measure the training speedup over BP of a parallel DFP implementation, for an MLP trained on MNIST at varying network width, depth, and orthogonality-update interval. For this experiment, we used four H100 NVIDIA GPUs to exploit DFP's update locality by offloading its workload to multiple devices, while BP is data-parallelized across the four devices. More details about the multi-GPU implementations of BP and DFP can be found in Appendix A.2.4.

Figure 3 shows that DFP trains faster than BP in almost all evaluated settings. For smaller networks, DFP is slower than BP, as at that scale, the communication cost is dominated by synchronization rather than by data movement or compute, and DFP's training step requires more synchronization operations than BP's single gradient aggregation. As either the depth or the width of the network increases, BP begins to lag behind DFP, which is not constrained by an iterative backward pass and requires less data movement. Indeed, DFP achieves its highest speedup over BP at the largest depths. For wider networks, DFP suffers from the cubic cost of the orthogonality loss, and, despite remaining faster than BP, its speedup begins to deteriorate. This issue can be mitigated by increasing the interval at which the orthogonality loss is applied. Computing it once every 5 batches is already sufficient to flatten the downward trend. Larger intervals, such as 20, even reverse it entirely, with DFP reaching a training speedup of up to $2.4\times$ over BP at the largest depths and widths.

5 Discussion and Future Work

While DFP significantly improves upon previous direct feedback methods, its feedback-weight update relies on the transpose of the forward weights. However, in contrast to BP, it does not strictly reintroduce weight transport, as the forward weights are never involved in propagating the error signal (Nøkland, 2016; Akrout et al., 2019; Ororbia, 2023). From the biological-plausibility perspective

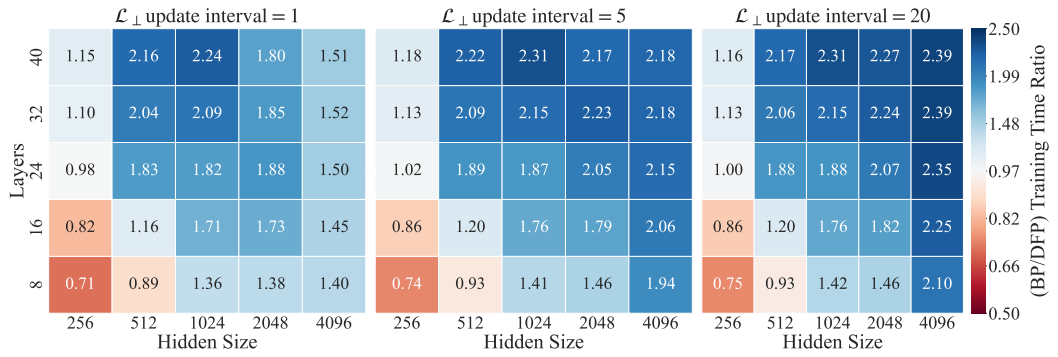

Figure 3: Training time ratio of BP over parallelized DFP (values above 1 indicate DFP is faster) for an MLP trained on MNIST on four H100 NVIDIA GPUs, with batch size 256 per GPU, averaged over 5 epochs. Each matrix corresponds to a different update interval of $\mathcal{L}_\perp$, spanning once every 1, 5 and 20 batches respectively. DFP trains faster than BP in almost all evaluated settings, with gains growing with depth, reaching up to a $2.4\times$ speedup at the largest depths and widths.

that also motivated DFA, the need for transposed weights can nevertheless be addressed by learning a symmetric set of weights, as has already been explored in the literature (Kolen & Pollack, 1994; Akrout et al., 2019). Concerning hardware implementations, this requirement is directly supported by transposable memory structures, which enable bidirectional access to the stored weights (Kim et al., 2018; Song et al., 2023; Yuan et al., 2025).

A possible avenue for future work is to incorporate nonlinearities in the feedback-weight update of DFP, potentially yielding gradients even more aligned with those of BP in the nonlinear regime. However, such a formulation might lose DFP's independence between the feedback-weight update and the forward pass. This will also require solving a stability challenge. Indeed, preliminary experiments show that the activities change substantially from batch to batch, and this fast-changing component conflicts with the large learning rate that the feedback weights require to track the slowly varying forward weights.

Finally, while our parallel implementation serves to validate the wall-clock time gains already achievable with a standard PyTorch implementation, substantially larger gains beyond MLP workloads will likely require custom CUDA kernels. This also motivates exploring implementations on custom massively parallel hardware, such as Intel Loihi (Davies et al., 2018), the Graphcore IPU (Jia et al., 2019), or photonic processors, with the latter already explored for DFA in the work of Filipovich et al. (2022).

6 Conclusion

In this work, we presented DFP, a novel direct feedback algorithm that addresses credit assignment in deep neural networks from a perspective complementary to that of DFA. While the latter enables learning by implicitly aligning the forward weights with fixed random direct connections, DFP instead introduces dynamics on the feedback weights to drive their alignment toward the forward weights, explicitly coupling the two. We demonstrated analytically that, under fast feedback and slow forward dynamics, DFP's feedback matrices are guaranteed to eventually deliver gradients matching those of BP in the linear regime and closely approximating them in the nonlinear regime. Our experiments empirically confirmed this result, exhibiting exact gradient alignment with BP in linear networks and higher alignment than DFA in nonlinear ones. This is reflected in classification performance, as DFP not only outperforms previous direct feedback variants, but also scales to more complex settings substantially better than DFA, narrowing the gap with BP while preserving the update locality of both the forward and feedback weights, and thus its parallelizability. Exploiting this property, DFP empirically achieved a speedup over BP in almost all evaluated MLP depths and widths. DFP therefore provides a mathematically grounded algorithm that extends DFA and further narrows the remaining gap between direct feedback learning methods and BP.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the text (main manuscript and appendices), to assist in writing the code, especially for plot generation or the parallel implementation on multiple devices starting from the authors' codebase, and to double-check the correctness of both the code and the mathematical claims and proofs, which were formulated and written by the authors. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, or propose or refine hypotheses, and the remaining tasks with required disclosure are not applicable to this work. We have reviewed all AI-assisted work. LLM-generated code and text were verified for correctness by the first author. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work fully adheres to the ICLR Code of Ethics. This study does not involve human subjects and relies exclusively on well-known and standard open-source datasets. The research presents no conflicts of interest and does not raise any privacy, security, or safety concerns. All experiments, results, and comparisons have been conducted and presented with scientific integrity, ensuring accuracy, transparency, and reproducibility.

REFERENCES

Mohamed Akrout, Collin Wilson, Peter Humphreys, Timothy Lillicrap, and Douglas B Tweed. Deep learning without weight transport. *Advances in neural information processing systems*, 32, 2019.

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

Davide Casnici, Martin Lefebvre, Justin Dauwels, and Charlotte Frenkel. Accelerated predictive coding networks via direct Kolen-Pollack feedback alignment. *arXiv preprint arXiv:2602.15571*, 2026.

Dan C Cireşan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *International conference on medical image computing and computer-assisted intervention*, pp. 411–418. Springer, 2013.

Brian Crafton, Abhinav Parihar, Evan Gebhardt, and Arijit Raychowdhury. Direct feedback alignment with sparse connections for local learning. *Frontiers in neuroscience*, 13:525, 2019.

Stéphane d'Ascoli, Levent Sagun, Giulio Biroli, and Joan Bruna. Finding the needle in the haystack with convolutions: on the benefits of architectural bias. *Advances in Neural Information Processing Systems*, 32, 2019.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

Timothy Dozat. Incorporating Nesterov momentum into Adam. *ICLR Workshop*, 2016.

Benjamin Ellenberger, Paul Haider, Federico Benitez, Jakob Jordan, Kevin Max, Ismael Jaras, Laura Kriener, and Mihai A Petrovici. Backpropagation through space, time and the brain. *Nature Communications*, 17(1):66, 2025. doi: 10.1038/s41467-025-66666-z.

Matthew J Filipovich, Zhimu Guo, Mohammed Al-Qadasi, Bicky A Marquez, Hugh D Morison, Volker J Sorger, Paul R Prucnal, Sudip Shekhar, and Bhavin J Shastri. Silicon photonic architecture for training deep neural networks with direct feedback alignment. *Optica*, 9(12):1323–1332, 2022.

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

Charlotte Frenkel, David Bol, and Giacomo Indiveri. Bottom-up and top-down approaches for the design of neuromorphic processing systems: Tradeoffs and synergies between natural and artificial intelligence. *Proceedings of the IEEE*, 111(6):623–652, 2023.

Donghyeon Han and Hoi-jun Yoo. Efficient convolutional neural network training with direct feedback alignment. *arXiv preprint arXiv:1901.01986*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Frank Charles Hoppensteadt. Singular perturbations on the infinite interval. *Transactions of the American Mathematical Society*, 123(2):521–535, 1966.

Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. In *International conference on machine learning*, pp. 1627–1635. PMLR, 2017.

Zhe Jia, Blake Tillman, Marco Maggioni, and Daniele Paolo Scarpazza. Dissecting the Graphcore IPU architecture via microbenchmarking. *arXiv preprint arXiv:1912.03413*, 2019.

Jinseok Kim, Jongeun Koo, Taesu Kim, and Jae-Joon Kim. Efficient synapse memory structure for reconfigurable digital neuromorphic hardware. *Frontiers in neuroscience*, 12:829, 2018.

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

John F Kolen and Jordan B Pollack. Backpropagation without weight transport. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 3, pp. 1375–1380. IEEE, 1994.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ya Le, Xuan Yang, et al. Tiny ImageNet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Yann LeCun. MNIST handwritten digit database. *ATT Labs*, 2010.

Timothy P Lillicrap and Adam Santoro. Backpropagation through time and the brain. *Current opinion in neurobiology*, 55:82–89, 2019.

Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random feedback weights support learning in deep neural networks. *arXiv preprint arXiv:1411.0247*, 2014.

Seppo Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16(2):146–160, 1976.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Estelle Massart. Orthogonal regularizers in deep learning: how to handle rectangular matrices? In *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 1294–1299. IEEE, 2022.

Hesham Mostafa, Vishwajith Ramesh, and Gert Cauwenberghs. Deep supervised learning using local errors. *Frontiers in neuroscience*, 12:608, 2018.

Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Robert E O'Malley et al. *Singular perturbation methods for ordinary differential equations*, volume 89. Springer, 1991.

Alexander G Ororbia. Brain-inspired machine intelligence: A survey of neurobiologically-plausible credit assignment. *arXiv preprint arXiv:2312.09257*, 2023.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Maria Refinetti, Stéphane d'Ascoli, Ruben Ohana, and Sebastian Goldt. Align, then memorise: the dynamics of learning with feedback alignment. In *International Conference on Machine Learning*, pp. 8925–8935. PMLR, 2021.

Arani Roy, Marco P Apolinario, Shristi Das Biswas, and Kaushik Roy. Feedback alignment meets low-rank manifolds: A structured recipe for local learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2984–2992, 2026.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *2nd International Conference on Learning Representations (ICLR)*, 2014.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

Jiahao Song, Xiyuan Tang, Xin Qiao, Yuan Wang, Runsheng Wang, and Ru Huang. A 28 nm 16 kb bit-scalable charge-domain transpose 6T SRAM in-memory computing macro. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 70(5):1835–1845, 2023.

Andrei Nikolaevich Tikhonov. Systems of differential equations containing small parameters in the derivatives. *Matematicheskii sbornik*, 73(3):575–586, 1952.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ferdinand Verhulst. *Methods and applications of singular perturbations: boundary layers and multiple timescale dynamics*. Springer, 2005.

Matthew Bailey Webster, Jonghyun Choi, and Changwook Ahn. Learning the connections in direct feedback alignment, 2021.

Paul J. Werbos. Backpropagation: past and future. In *IEEE 1988 International Conference on Neural Networks*, pp. 343–353, 1988. doi: 10.1109/ICNN.1988.23866.

James CR Whittington and Rafal Bogacz. Theories of error back-propagation in the brain. *Trends in cognitive sciences*, 23(3):235–250, 2019.

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

Heesung Yang, Soha Lee, and Hyeyoung Park. Direct feedback learning with local alignment support. *IEEE Access*, 12:81388–81397, 2024.

Yiyang Yuan, Bingxin Zhang, Yiming Yang, Yishan Luo, Qirui Chen, Shidong Lv, Hao Wu, Cailian Ma, Ming Li, Jinshan Yue, et al. 14.5 a 28nm 192.3 TFLOPS/W accurate/approximate dual-mode-transpose digital 6T-SRAM CIM macro for floating-point edge training and inference. In *2025 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 68, pp. 258–260. IEEE, 2025.

A Appendix

A.1 Convergence under separation of timescales

In this section, we analyze why a separation of timescales constitutes a sufficient condition, under the mentioned assumptions, for the feedback matrices to converge to the chain of transposed forward weights that defines the linear BP gradient mapping. In contrast to DFA, where the forward weights align with the feedback matrices to enable learning, DFP addresses the problem from the opposite perspective, forcing the feedback matrices to track the forward weights. As will be demonstrated in this proof, a sufficient condition for achieving this objective is the presence of a separation of timescales, whereby the dynamics of the feedback matrices evolve on a much faster timescale than those of the forward weights. Intuitively, if the forward weights evolve too rapidly, the feedback matrices cannot track their changes. Consequently, the feedback matrices must evolve on an (ideally infinitely) faster timescale than the forward weights. This setting is known in the literature as a singular perturbation problem, and Tikhonov's theorem provides a proof of convergence of such systems. For completeness, we first show how our problem is a singular perturbation system to use Tikhonov's theorem, and then we also provide a complete proof of convergence, showing the generic formulation for the time evolution at a generic layer ℓ. Importantly, throughout the whole section, for the sake of notational brevity, we do not explicitly denote the time dependence of variables unless strictly necessary. Nevertheless, we highlight that all quantities, including forward weights, feedback weights, gradients, and tracking errors, are time-varying (for instance, $\Psi_\ell(t)$ will be written simply as Ψ_ℓ).

Singular perturbation formulation. We begin by introducing the general formulation. Let us consider a system in singular perturbation form:

$$\begin{aligned} \dot{x} &= f(x, z) \\ \varphi \dot{z} &= g(x, z), \end{aligned} \tag{17}$$

where $0 < \varphi \ll 1$ multiplies the time derivative of z, inducing a separation of timescales between the slow variable x and the fast variable z. In the limit $\varphi \to 0$, the second equation degenerates into $g(x, z) = 0$. Denoting its root by $z = \psi(x)$, the degenerate system is defined as

$$\begin{aligned} \dot{x} &= f(x, \psi(x)) \\ z &= \psi(x), \end{aligned} \tag{18}$$

where the slow variable x then evolves according to the root of $g(x, z)$. The fast variable z, in turn, is governed by the associated system $\dot{z} = g(x, z)$ obtained by treating the slow x as a constant. Tikhonov's theorem states that, as $\varphi \to 0$, the solution of the full system in Equation 17 converges to that of the degenerate system, provided $z = \psi(x)$ is an asymptotically stable equilibrium of the associated system (Tikhonov, 1952; O'Malley et al., 1991; Verhulst, 2005).

We now show that the coupled DFP dynamics of Equation 14 instantiate exactly this structure, with the feedback matrices as the fast variable converging to the chain of transposed forward weights, and the forward weights as the slow variable evolving, in the limit, under exact (linear) BP gradients. Recalling the learning rules

$$\begin{aligned} \frac{d\Psi_\ell}{dt} &= -\eta_\Psi \left(\Psi_\ell - \Theta_\ell^\top \Psi_{\ell+1} \right), \\ \frac{d\Theta_\ell}{dt} &= -\eta_\Theta \left(\phi'(\hat{x}_{\ell+1}) \odot (\Psi_{\ell+1} \delta_L)\, x_\ell^\top + \eta_\perp \nabla_{\Theta_\ell} \mathcal{L}_\perp \right), \end{aligned} \tag{19}$$

to exhibit the singular perturbation structure explicitly, we define the timescale ratio as

$$\varphi = \frac{\eta_\Theta}{\eta_\Psi}, \tag{20}$$

and measure time in units s of the slow forward timescale, such that $\frac{ds}{dt} = \eta_\Theta$. By applying the chain rule and substituting Equation 20 into Equation 19, we obtain:

$$\begin{aligned} \varphi \frac{d\Psi_\ell}{ds} &= -\left(\Psi_\ell - \Theta_\ell^\top \Psi_{\ell+1} \right), \\ \frac{d\Theta_\ell}{ds} &= -\left(\phi'(\hat{x}_{\ell+1}) \odot (\Psi_{\ell+1} \delta_L)\, x_\ell^\top + \eta_\perp \nabla_{\Theta_\ell} \mathcal{L}_\perp \right), \end{aligned} \tag{21}$$

which is exactly the canonical form of Equation 17, with the small parameter $\varphi = \eta_\Theta/\eta_\Psi$ multiplying the derivative of the fast variable Ψ. Since the feedback loss $\mathcal{L}_{\Psi_\ell}$ of Equation 8 is strictly convex in Ψ_ℓ, its unique minimum $\Psi_\ell = \Theta_\ell^\top \Psi_{\ell+1}$ is an asymptotically stable equilibrium of the associated system, satisfying Tikhonov's stability condition. Therefore, according to Tikhonov's theorem, in the limit $\varphi \to 0$, the feedback dynamics collapse to the root of the associated system

$$\Psi_\ell = \Theta_\ell^\top \Psi_{\ell+1}, \tag{22}$$

which recursively yields

$$\Psi_\ell = \prod_{i=\ell}^{L-1} \Theta_i^\top = \Psi_\ell^\star. \tag{23}$$

Tikhonov's theorem therefore guarantees that the feedback matrices remain within $O(\varphi)$ of their moving equilibrium over any finite training interval. Theorem 1, however, concerns the infinite-horizon limit $T \to \infty$. Extending Tikhonov's guarantee to the infinite interval requires the additional assumption of uniform asymptotic stability of the slow system (Hoppensteadt, 1966), that is, of the forward weights. Such a property cannot be guaranteed in general, as the gradient descent is on an objective that is non-convex in the individual weight matrices, even in the linear regime, and acts through the matrices $\Psi_\ell^\star$ corresponding to a chain of transposed forward weights. Nevertheless, the orthogonality regularizer of Equation 13 constrains each factor toward the orthogonal manifold, keeping the singular values of the chain close to 1 and the dynamics well-conditioned, free of vanishing or exploding values, making this stability assumption reasonable in our setting (note that the orthogonality-loss gradients act only on the slow forward weights and do not alter the feedback update rule). We nevertheless prove the infinite-horizon result directly, without invoking the infinite-interval extension, by deriving a uniform tracking bound whose lag term remains bounded independently of T. Taking the successive limits $T \to \infty$ and $\varphi \to 0$ then drives the tracking error to zero, yielding exact alignment with the (linear) BP mapping.

Extensive proof. To formalize this argument, we first derive the convergence proof for the last three layers, and we then introduce a generic formulation demonstrating that it converges for every layer in the network.

As we have seen in Section 3, DFP's fixed point for the feedback matrices corresponds to Equation 11. Therefore, we can define the instantaneous tracking error between our current Ψ_ℓ and the instantaneous desired value $\Psi_\ell^\star$ as

$$\mathcal{E}_\ell = \Psi_\ell - \Psi_\ell^\star, \tag{24}$$

to analyze its dynamics throughout training. We can proceed by analyzing the tracking error dynamics, given by:

$$\begin{aligned} \dot{\mathcal{E}}_\ell &= \dot{\Psi}_\ell - \dot{\Psi}_\ell^\star \\ &= -\eta_\Psi \left(\Psi_\ell - \Theta_\ell^\top \Psi_{\ell+1}\right) - \dot{\Psi}_\ell^\star, \end{aligned} \tag{25}$$

where we have substituted the gradient defined in Equation 10. Now, we add and subtract the same quantity $\Theta_\ell^\top \Psi_{\ell+1}^\star$, and use the recursive formulation of the feedback matrices in Equation 7, obtaining

$$\begin{aligned} \dot{\mathcal{E}}_\ell &= -\eta_\Psi \left(\Psi_\ell - \Theta_\ell^\top \Psi_{\ell+1}^\star + \Theta_\ell^\top \Psi_{\ell+1}^\star - \Theta_\ell^\top \Psi_{\ell+1}\right) - \dot{\Psi}_\ell^\star \\ &= -\eta_\Psi \left(\Psi_\ell - \Psi_\ell^\star + \Theta_\ell^\top \Psi_{\ell+1}^\star - \Theta_\ell^\top \Psi_{\ell+1}\right) - \dot{\Psi}_\ell^\star \\ &= -\eta_\Psi \left(\Psi_\ell - \Psi_\ell^\star - \Theta_\ell^\top \left(\Psi_{\ell+1} - \Psi_{\ell+1}^\star\right)\right) - \dot{\Psi}_\ell^\star. \end{aligned} \tag{26}$$

By applying the definition of the tracking error in Equation 24 to Equation 26, we obtain:

$$\dot{\mathcal{E}}_\ell = -\eta_\Psi \mathcal{E}_\ell + \eta_\Psi \Theta_\ell^\top \mathcal{E}_{\ell+1} - \dot{\Psi}_\ell^\star. \tag{27}$$

We can now define an auxiliary function $f_\ell(t)$ as

$$f_\ell(t) = \eta_\Psi \Theta_\ell^\top \mathcal{E}_{\ell+1} - \dot{\Psi}_\ell^\star, \tag{28}$$

allowing us to rewrite Equation 27 in a more compact way:

$$\dot{\mathcal{E}}_\ell + \eta_\Psi \mathcal{E}_\ell = f_\ell(t). \tag{29}$$

Noticing that Equation 29 takes the form $\dot{y}(t) + p(t)y(t) = f(t)$, we can proceed by using the integrating factor method, and for clarity, in this part of the proof we explicitly denote time dependence where required. For readers unfamiliar with this method, we define the integrating factor $r(t)$ as:

$$r(t) = e^{\int p(t)\, dt}. \tag{30}$$

Multiplying both sides by $r(t)$ results in

$$r(t)\dot{y}(t) + r(t)p(t)y(t) = r(t)f(t), \tag{31}$$

and by applying the derivative product rule, the left-hand side is rewritten as:

$$\frac{d}{dt}\left(r(t)y(t)\right) = r(t)f(t). \tag{32}$$

In our specific case, by multiplying both sides by the integrating factor $r(t) = e^{\eta_\Psi t}$, we obtain:

$$e^{\eta_\Psi t}\dot{\mathcal{E}}_\ell + e^{\eta_\Psi t}\eta_\Psi \mathcal{E}_\ell = e^{\eta_\Psi t} f_\ell(t), \tag{33}$$

which can now be written as

$$\frac{d}{dt}\left(e^{\eta_\Psi t}\mathcal{E}_\ell\right) = e^{\eta_\Psi t} f_\ell(t). \tag{34}$$

Continuing by integrating both sides we obtain:

$$\int_0^T \frac{d}{dt}\left(e^{\eta_\Psi t}\mathcal{E}_\ell\right) dt = \int_0^T e^{\eta_\Psi t} f_\ell(t) dt, \tag{35}$$

where the left-hand side is equal to the difference of $e^{\eta_\Psi t}\mathcal{E}_\ell$ evaluated at $t = T$ and $t = 0$:

$$e^{\eta_\Psi T}\mathcal{E}_\ell(T) - \mathcal{E}_\ell(0) = \int_0^T e^{\eta_\Psi t} f_\ell(t) dt. \tag{36}$$

By moving $\mathcal{E}_\ell(0)$ to the right-hand side of the equation, and by multiplying everything by $e^{-\eta_\Psi T}$, we obtain:

$$\mathcal{E}_\ell(T) = e^{-\eta_\Psi T}\mathcal{E}_\ell(0) + \int_0^T e^{\eta_\Psi (t-T)} f_\ell(t) dt. \tag{37}$$

Finally, by substituting back $f_\ell(t)$ from Equation 28, we obtain the final form of our tracking error after a specific training time T:

$$\begin{aligned} \mathcal{E}_\ell(T) &= e^{-\eta_\Psi T}\mathcal{E}_\ell(0) + \int_0^T e^{\eta_\Psi (t-T)} \left(\eta_\Psi \Theta_\ell^\top \mathcal{E}_{\ell+1} - \dot{\Psi}_\ell^\star\right) dt \\ &= e^{-\eta_\Psi T}\mathcal{E}_\ell(0) + \eta_\Psi \int_0^T e^{\eta_\Psi (t-T)} \Theta_\ell^\top \mathcal{E}_{\ell+1} dt - \int_0^T e^{\eta_\Psi (t-T)} \dot{\Psi}_\ell^\star dt. \end{aligned} \tag{38}$$

Now, we can define three auxiliary variables for brevity in the following derivations:

$$\begin{aligned} S_\ell &= e^{-\eta_\Psi T}\, \mathcal{E}_\ell(0) \\ P_\ell &= \eta_\Psi \int_0^T e^{\eta_\Psi (t-T)}\, \Theta_\ell^\top \mathcal{E}_{\ell+1}\, dt \\ Q_\ell &= \int_0^T e^{\eta_\Psi (t-T)}\, \dot{\Psi}_\ell^\star\, dt. \end{aligned} \tag{39}$$

Here, we observe that the tracking error of the feedback matrices for a generic layer ℓ can be decomposed into three components. The first two, S_ℓ and P_ℓ, represent the "memory" terms: S_ℓ incorporates the error contribution from the current layer, while P_ℓ integrates the effect from higher layers. The remaining component, Q_ℓ, integrates the forward weights lag term.

Tracking Error Convergence at $\ell = L-1$. Starting from the last layer $\ell = L-1$, we now evaluate the tracking error in the limit $T \to \infty$, as we are interested in the convergence of DFP's feedback matrices. To do so, we first define the boundary condition $\mathcal{E}_L = 0$, as we have no tracking error coming from the next layer at $\ell = L-1$, thus reducing the first integral to 0. Furthermore, as mentioned in Section 3, as a boundary condition we also have $\Psi_L = I$, and thus $\dot{\Psi}_L = 0$. As a consequence of these boundary conditions, we have that $P_{L-1} = 0$, resulting in:

$$\begin{aligned}\mathcal{E}_{L-1}(T) &= S_{L-1} - Q_{L-1} \\ &= e^{-\eta_\Psi T}\mathcal{E}_{L-1}(0) - \int_0^T e^{\eta_\Psi (t-T)}\dot{\Theta}_{L-1}^\top dt.\end{aligned} \tag{40}$$

Before evaluating the limit, we expand the time derivative of Θ_{L-1}, which evolves according to the continuous-time update in Equation 14. We denote the DFP gradient proxy by $\tilde{\nabla}_{\Theta_\ell}\mathcal{L}$. Note that, for $\ell = L-1$, the projected error coincides with the true gradient since $\Psi_L = I$, and thus $\tilde{\nabla}_{\Theta_{L-1}}\mathcal{L} = \nabla_{\Theta_{L-1}}\mathcal{L}$. By assuming that the gradient's norm can be bounded by an arbitrarily large finite quantity, that is, $\exists B_{L-1} : \|\tilde{\nabla}_{\Theta_{L-1}}\mathcal{L}\|_2 \leq B_{L-1}$, $\forall t \geq 0$, which also absorbs the orthogonality gradient when present, we can therefore write

$$\begin{aligned}\mathcal{E}_{L-1}(T) &= e^{-\eta_\Psi T}\mathcal{E}_{L-1}(0) + \eta_\Theta \int_0^T e^{\eta_\Psi (t-T)}\tilde{\nabla}_{\Theta_{L-1}}\mathcal{L}^\top dt \\ \|\mathcal{E}_{L-1}(T)\|_2 &= \left\| e^{-\eta_\Psi T}\mathcal{E}_{L-1}(0) + \eta_\Theta \int_0^T e^{\eta_\Psi (t-T)}\tilde{\nabla}_{\Theta_{L-1}}\mathcal{L}^\top dt \right\|_2 \\ &\leq e^{-\eta_\Psi T}\|\mathcal{E}_{L-1}(0)\|_2 + \eta_\Theta B_{L-1}\int_0^T e^{\eta_\Psi (t-T)}dt \\ &\leq e^{-\eta_\Psi T}\|\mathcal{E}_{L-1}(0)\|_2 + \frac{\eta_\Theta}{\eta_\Psi}B_{L-1}\left(1 - e^{-\eta_\Psi T}\right),\end{aligned} \tag{41}$$

where we have applied the triangle inequality and the integral triangle inequality. Computing the limit on both sides, we obtain

$$\begin{aligned}\lim_{T\to\infty}\|\mathcal{E}_{L-1}(T)\|_2 &\leq \lim_{T\to\infty}\left(e^{-\eta_\Psi T}\|\mathcal{E}_{L-1}(0)\|_2 + \frac{\eta_\Theta}{\eta_\Psi}B_{L-1}\left(1 - e^{-\eta_\Psi T}\right)\right) \\ &\leq \|\mathcal{E}_{L-1}(0)\|_2 \lim_{T\to\infty} e^{-\eta_\Psi T} + \frac{\eta_\Theta}{\eta_\Psi}B_{L-1}\lim_{T\to\infty}\left(1 - e^{-\eta_\Psi T}\right),\end{aligned} \tag{42}$$

and in the limit $T \to \infty$, the term $e^{-\eta_\Psi T}$ vanishes, yielding the following asymptotic bound on the tracking error's norm:

$$\lim_{T\to\infty}\|\mathcal{E}_{L-1}(T)\|_2 \leq \frac{\eta_\Theta}{\eta_\Psi}B_{L-1}. \tag{43}$$

Therefore, the tracking error can be made arbitrarily small by ensuring a sufficiently large timescale separation between the forward and feedback dynamics, and, given the definition of the timescale ratio in Equation 20, it vanishes in the limit:

$$\lim_{\varphi\to 0}\lim_{T\to\infty}\|\mathcal{E}_{L-1}(T)\|_2 \leq \lim_{\varphi\to 0}\varphi B_{L-1} = 0. \tag{44}$$

Tracking Error Convergence at $\ell = L-2$. When considering the layer $L-2$, we have to take into account all terms in Equation 38, having:

$$\begin{aligned}\mathcal{E}_{L-2}(T) &= S_{L-2} + P_{L-2} - Q_{L-2} \\ &= e^{-\eta_\Psi T}\mathcal{E}_{L-2}(0) + \eta_\Psi \int_0^T e^{\eta_\Psi (t-T)}\Theta_{L-2}^\top \mathcal{E}_{L-1}dt - \int_0^T e^{\eta_\Psi (t-T)}\dot{\Psi}_{L-2}^\star dt.\end{aligned} \tag{45}$$

Similarly to what we did for $\mathcal{E}_{L-1}$, we can demonstrate that in the limit $T \to \infty$, if $\varphi \to 0$, then $\mathcal{E}_{L-2} \to 0$. By applying the triangle inequality, it is sufficient to demonstrate that

$$\lim_{\varphi\to 0}\lim_{T\to\infty}\|\mathcal{E}_{L-2}(T)\|_2 \leq \lim_{\varphi\to 0}\lim_{T\to\infty}\left(\|S_{L-2}\|_2 + \|P_{L-2}\|_2 + \|Q_{L-2}\|_2\right) = 0. \tag{46}$$

Starting with S_{L-2}, we have

$$\|S_{L-2}\|_2 = e^{-\eta_\Psi T}\|\mathcal{E}_{L-2}(0)\|_2. \tag{47}$$

Taking the limit as $T \to \infty$ gives

$$\lim_{T\to\infty} \|S_{L-2}\|_2 = \lim_{T\to\infty} e^{-\eta_\Psi T}\|\mathcal{E}_{L-2}(0)\|_2 = 0, \tag{48}$$

and therefore the same applies in the limit $\varphi \to 0$. Proceeding with P_{L-2}, we have

$$\begin{aligned}\|P_{L-2}\|_2 &= \eta_\Psi \left\| \int_0^T e^{\eta_\Psi (t-T)}\, \Theta_{L-2}^\top \mathcal{E}_{L-1} dt \right\|_2 \\ &\leq \eta_\Psi \int_0^T e^{\eta_\Psi (t-T)}\, \left\|\Theta_{L-2}^\top\right\|_2 \|\mathcal{E}_{L-1}\|_2\, dt.\end{aligned} \tag{49}$$

As the weights of a neural network are typically penalized by a weight decay (and in our case through an orthogonality loss), it is reasonable to assume that $\exists A_{L-2} : \|\Theta_{L-2}^\top\|_2 \leq A_{L-2},\ \forall t \geq 0$. Furthermore, from Equation 41, we have that

$$\begin{aligned}\|\mathcal{E}_{L-1}(t)\|_2 &\leq e^{-\eta_\Psi t}\|\mathcal{E}_{L-1}(0)\|_2 + \varphi B_{L-1}\left(1 - e^{-\eta_\Psi t}\right) \\ &\leq e^{-\eta_\Psi t}\|\mathcal{E}_{L-1}(0)\|_2 + \varphi B_{L-1},\end{aligned} \tag{50}$$

as $1 - e^{-\eta_\Psi t} \in [0, 1]$ for $t \geq 0$, and $\varphi, B_{L-1} \in \mathbb{R}^+$. Therefore we can further bound $\|P_{L-2}\|_2$ by substituting A_{L-2} and Equation 50 into Equation 49:

$$\begin{aligned}\|P_{L-2}\|_2 &\leq \eta_\Psi A_{L-2} \int_0^T e^{\eta_\Psi (t-T)} \left(e^{-\eta_\Psi t}\|\mathcal{E}_{L-1}(0)\|_2 + \varphi B_{L-1}\right) dt \\ &\leq \eta_\Psi A_{L-2} e^{-\eta_\Psi T}\|\mathcal{E}_{L-1}(0)\|_2 \int_0^T dt + \eta_\Psi \varphi A_{L-2} B_{L-1} \int_0^T e^{\eta_\Psi (t-T)} dt \\ &\leq A_{L-2}\left(\eta_\Psi T e^{-\eta_\Psi T}\|\mathcal{E}_{L-1}(0)\|_2 + \varphi B_{L-1}\left(1 - e^{-\eta_\Psi T}\right)\right).\end{aligned} \tag{51}$$

We can now evaluate the limit of Equation 51 for $T \to \infty$, obtaining

$$\begin{aligned}\lim_{T\to\infty} \|P_{L-2}\|_2 &\leq \lim_{T\to\infty} A_{L-2}\left(\eta_\Psi T e^{-\eta_\Psi T}\|\mathcal{E}_{L-1}(0)\|_2 + \varphi B_{L-1}\left(1 - e^{-\eta_\Psi T}\right)\right) \\ &\leq \varphi A_{L-2} B_{L-1},\end{aligned} \tag{52}$$

and by evaluating the limit $\varphi \to 0$ we finally obtain:

$$\lim_{\varphi\to 0} \lim_{T\to\infty} \|P_{L-2}\|_2 \leq \lim_{\varphi\to 0} \varphi A_{L-2} B_{L-1} = 0. \tag{53}$$

Before proceeding with Q_{L-2}, we first write the general bound for the time derivative of $\Psi_\ell^\star$. The latter, by applying the derivative multiplication rule, can be written in general as

$$\begin{aligned}\dot{\Psi}_\ell^\star &= \frac{d}{dt}\left(\prod_{i=\ell}^{L-1} \Theta_i^\top\right) \\ &= \sum_{j=\ell}^{L-1} \left(\prod_{i=\ell}^{j-1} \Theta_i^\top\right) \dot{\Theta}_j^\top \left(\prod_{k=j+1}^{L-1} \Theta_k^\top\right) \\ &= -\eta_\Theta \sum_{j=\ell}^{L-1} \left(\prod_{i=\ell}^{j-1} \Theta_i^\top\right) \tilde{\nabla}_{\Theta_j} \mathcal{L}^\top \Psi_{j+1}^\star,\end{aligned} \tag{54}$$

and, by applying the triangle inequality and norm submultiplicativity, it can be bounded as follows:

$$\|\dot{\Psi}_\ell^\star\|_2 \leq \eta_\Theta \sum_{j=\ell}^{L-1} \left(\prod_{i=\ell}^{j-1} \|\Theta_i^\top\|_2\right) \|\tilde{\nabla}_{\Theta_j} \mathcal{L}^\top \Psi_{j+1}^\star\|_2. \tag{55}$$

Let us assume that $\exists A_\ell < \infty$ such that

$$\|\Theta_\ell\|_2 \leq A_\ell, \quad \forall \ell \in \{0, \ldots, L-1\},\ \forall t \geq 0, \tag{56}$$

and $\exists B_\ell < \infty$ such that

$$\|\tilde{\nabla}_{\Theta_\ell}\mathcal{L}^\top \Psi^\star_{\ell+1}\|_2 \leq B_\ell, \quad \forall \ell \in \{1, \dots, L-1\},\ \forall t \geq 0. \tag{57}$$

We can then define the recursive constant bound C_ℓ as

$$C_\ell = \sum_{j=\ell}^{L-1} \left(\prod_{i=\ell}^{j-1} A_i \right) B_j, \tag{58}$$

and write the uniform bound on the norm of $\dot{\Psi}^\star_\ell$ using Equation 55 and Equation 58:

$$\|\dot{\Psi}^\star_\ell\|_2 \leq \eta_\Theta C_\ell. \tag{59}$$

We now proceed with Q_{L-2}, which can be bounded using Equation 59 as

$$\begin{aligned} \|Q_{L-2}\|_2 &\leq \int_0^T e^{\eta_\Psi (t-T)} \|\dot{\Psi}^\star_{L-2}\|_2 dt \\ &\leq \eta_\Theta C_{L-2} \int_0^T e^{\eta_\Psi (t-T)} dt \\ &\leq \varphi C_{L-2}(1 - e^{-\eta_\Psi T}). \end{aligned} \tag{60}$$

By taking the limit $T \to \infty$, we obtain

$$\lim_{T\to\infty} \|Q_{L-2}\|_2 \leq \lim_{T\to\infty} \varphi C_{L-2}(1 - e^{-\eta_\Psi T}) = \varphi C_{L-2}, \tag{61}$$

and, successively, as $\varphi \to 0$,

$$\lim_{\varphi\to 0} \lim_{T\to\infty} \|Q_{L-2}\|_2 \leq \lim_{\varphi\to 0} \varphi C_{L-2} = 0. \tag{62}$$

Now, by using Equation 48, Equation 53, and Equation 62 in Equation 46 we obtain:

$$\lim_{\varphi\to 0} \lim_{T\to\infty} \|\mathcal{E}_{L-2}(T)\|_2 = 0, \tag{63}$$

completing the demonstration that, under separation of timescales, the tracking error converges to 0 for $\ell = L-2$ as well.

Tracking Error Convergence at $\ell = L-3$. To obtain the general formulation allowing us to demonstrate that the convergence guarantee holds for the entire network, we need to prove it for layer $\ell = L-3$ as well, as the tracking error at $\ell = L-2$ still inherits a null P term from the boundary conditions. Thus, it does not fully reflect the general case, where the P term of each layer recursively incorporates the one of the layer above, generating the polynomial memory term that appears in the general formulation.

Considering now the layer $\ell = L-3$, Equation 38 takes the following form:

$$\begin{aligned} \mathcal{E}_{L-3}(T) &= S_{L-3} + P_{L-3} - Q_{L-3} \\ &= e^{-\eta_\Psi T}\mathcal{E}_{L-3}(0) + \eta_\Psi \int_0^T e^{\eta_\Psi (t-T)} \Theta^\top_{L-3} \mathcal{E}_{L-2} dt - \int_0^T e^{\eta_\Psi (t-T)} \dot{\Psi}^\star_{L-3} dt. \end{aligned} \tag{64}$$

Again, we need to demonstrate that in the limit $T \to \infty$, if $\varphi \to 0$, then $\mathcal{E}_{L-3} \to 0$. As done previously, we can equivalently show that

$$\lim_{\varphi\to 0} \lim_{T\to\infty} \|\mathcal{E}_{L-3}(T)\|_2 \leq \lim_{\varphi\to 0} \lim_{T\to\infty} \Big(\|S_{L-3}\|_2 + \|P_{L-3}\|_2 + \|Q_{L-3}\|_2 \Big) = 0. \tag{65}$$

Starting with S_{L-3}, we again have:

$$\lim_{T\to\infty} \|S_{L-3}\|_2 = \lim_{T\to\infty} e^{-\eta_\Psi T} \|\mathcal{E}_{L-3}(0)\|_2 = 0, \tag{66}$$

independently of φ. Proceeding with P_{L-3}, we directly write its norm bound:

$$\|P_{L-3}\|_2 \leq \eta_\Psi A_{L-3} \int_0^T e^{\eta_\Psi (t-T)} \|\mathcal{E}_{L-2}\|_2 dt. \tag{67}$$

From the proof derived at layer $\ell = L-2$, we have that:

$$\begin{aligned}\|\mathcal{E}_{L-2}(T)\|_2 \leq\ & e^{-\eta_\Psi T}\|\mathcal{E}_{L-2}(0)\|_2 \\ & + A_{L-2}\left(\eta_\Psi T e^{-\eta_\Psi T}\|\mathcal{E}_{L-1}(0)\|_2 + \varphi B_{L-1}\left(1-e^{-\eta_\Psi T}\right)\right) \\ & + \varphi C_{L-2}(1-e^{-\eta_\Psi T}),\end{aligned} \tag{68}$$

which can be used to further bound P_{L-3}, by considering the error at a general time $T = t$. Therefore, we can now write Equation 67 as follows:

$$\begin{aligned}\|P_{L-3}\|_2 \leq\ & \eta_\Psi A_{L-3}\int_0^T e^{\eta_\Psi(t-T)}e^{-\eta_\Psi t}\|\mathcal{E}_{L-2}(0)\|_2\,dt \\ & + \eta_\Psi A_{L-3}A_{L-2}\int_0^T e^{\eta_\Psi(t-T)}\left(\eta_\Psi t e^{-\eta_\Psi t}\|\mathcal{E}_{L-1}(0)\|_2 + \varphi B_{L-1}(1-e^{-\eta_\Psi t})\right)dt \\ & + \eta_\Psi A_{L-3}\varphi C_{L-2}\int_0^T e^{\eta_\Psi(t-T)}(1-e^{-\eta_\Psi t})\,dt,\end{aligned} \tag{69}$$

resulting in:

$$\begin{aligned}\|P_{L-3}\|_2 \leq\ & \eta_\Psi A_{L-3}e^{-\eta_\Psi T}\|\mathcal{E}_{L-2}(0)\|_2\int_0^T dt \\ & + \eta_\Psi^2 A_{L-3}A_{L-2}e^{-\eta_\Psi T}\|\mathcal{E}_{L-1}(0)\|_2\int_0^T t dt \\ & + \eta_\Psi A_{L-3}A_{L-2}\varphi B_{L-1}\left(\int_0^T e^{\eta_\Psi(t-T)}dt - e^{-\eta_\Psi T}\int_0^T dt\right) \\ & + \eta_\Psi A_{L-3}\varphi C_{L-2}\left(\int_0^T e^{\eta_\Psi(t-T)}dt - e^{-\eta_\Psi T}\int_0^T dt\right).\end{aligned} \tag{70}$$

Proceeding by evaluating the integrals, we obtain

$$\begin{aligned}\|P_{L-3}\|_2 \leq\ & \eta_\Psi A_{L-3}Te^{-\eta_\Psi T}\left(\|\mathcal{E}_{L-2}(0)\|_2 + A_{L-2}\frac{\eta_\Psi T}{2}\|\mathcal{E}_{L-1}(0)\|_2\right) \\ & + \varphi A_{L-3}\left(A_{L-2}C_{L-1}+C_{L-2}\right)\left(1-e^{-\eta_\Psi T}(1+\eta_\Psi T)\right),\end{aligned} \tag{71}$$

as $B_\ell = C_\ell$ for $\ell = L-1$. Taking the limit $T\to\infty$ of $\|P_{L-3}\|_2$, we obtain

$$\lim_{T\to\infty}\|P_{L-3}\|_2 \leq \varphi A_{L-3}\left(A_{L-2}C_{L-1}+C_{L-2}\right), \tag{72}$$

and taking the limit $\varphi\to 0$ finally results in

$$\lim_{\varphi\to 0}\lim_{T\to\infty}\|P_{L-3}\|_2 \leq \lim_{\varphi\to 0}\varphi A_{L-3}\left(A_{L-2}C_{L-1}+C_{L-2}\right) = 0. \tag{73}$$

Considering now Q_{L-3}, we bound its time derivative $\dot{\Psi}^\star_{L-3}$, obtaining

$$\begin{aligned}\|Q_{L-3}\|_2 &\leq \int_0^T e^{\eta_\Psi(t-T)}\|\dot{\Psi}^\star_{L-3}\|_2 dt \\ &\leq \eta_\Theta C_{L-3}\int_0^T e^{\eta_\Psi(t-T)}dt \\ &\leq \varphi C_{L-3}(1-e^{-\eta_\Psi T}).\end{aligned} \tag{74}$$

By first taking the limit $T\to\infty$, we obtain

$$\lim_{T\to\infty}\|Q_{L-3}\|_2 \leq \lim_{T\to\infty}\varphi C_{L-3}(1-e^{-\eta_\Psi T}) = \varphi C_{L-3}, \tag{75}$$

taking then the limit $\varphi\to 0$, we obtain

$$\lim_{\varphi\to 0}\lim_{T\to\infty}\|Q_{L-3}\|_2 \leq \lim_{\varphi\to 0}\varphi C_{L-3} = 0. \tag{76}$$

Finally, we now substitute Equation 66, Equation 73, and Equation 76 into Equation 65, demonstrating that the tracking error at layer $\ell = L-3$ also converges to 0 in the limit, under separation of timescales:

$$\lim_{\varphi\to 0}\lim_{T\to\infty}\|\mathcal{E}_{L-3}(T)\|_2 = 0. \tag{77}$$

Convergence Generalization. We can now generalize the results we obtained for the previous steps, starting from the general bound on the ℓ-th layer tracking-error norm:

$$\lim_{\varphi\to 0}\lim_{T\to\infty}\|\mathcal{E}_\ell(T)\|_2 \leq \lim_{\varphi\to 0}\lim_{T\to\infty}\left(\|S_\ell\|_2 + \|P_\ell\|_2 + \|Q_\ell\|_2\right). \tag{78}$$

As we have derived before, we have that:

$$\|S_\ell\|_2 = e^{-\eta_\Psi T}\|\mathcal{E}_\ell(0)\|_2, \tag{79}$$

representing the vanishing initial condition error, always satisfying:

$$\lim_{T\to\infty}\|S_\ell\|_2 = \lim_{T\to\infty} e^{-\eta_\Psi T}\|\mathcal{E}_\ell(0)\|_2 = 0, \quad \forall \ell \in \{1,\dots,L-1\}. \tag{80}$$

Continuing with $\|P_\ell\|_2$, we observe that the bounds in Equation 51 and Equation 71 share the same structure once each lag contribution is relaxed to its limiting form, since $1 - e^{-\eta_\Psi T}\omega(T) \leq 1 - e^{-\eta_\Psi T}$ for any polynomial $\omega(T) \geq 1$. Therefore, we can write the general recursive formulation as follows:

$$\|P_\ell\|_2 \leq \underbrace{e^{-\eta_\Psi T}\xi_\ell(T)}_{\text{Memory Term}} + \underbrace{\varphi D_\ell\left(1 - e^{-\eta_\Psi T}\right)}_{\text{Lag Term}}, \tag{81}$$

where $\xi_\ell(T)$ is the memory term recursively incorporating initial condition errors from higher layers, represented by the following polynomial function in T of degree $L-\ell-1$:

$$\xi_\ell(T) = \sum_{i=\ell+1}^{L-1}\left(\prod_{j=\ell}^{i-1} A_j\right)\frac{(\eta_\Psi T)^{i-\ell}}{(i-\ell)!}\|\mathcal{E}_i(0)\|_2. \tag{82}$$

We can see the pattern originating Equation 82 by observing the definition of P_ℓ in Equation 39, where its norm is constrained by the tracking error of the subsequent layer, $\mathcal{E}_{\ell+1}$, whose bound can be expressed in terms of bounds on $S_{\ell+1}, P_{\ell+1}, Q_{\ell+1}$. The terms derived from $Q_{\ell+1}$ and from the lag component of $P_{\ell+1}$ contribute to the lag term of P_ℓ, while $S_{\ell+1}$ and the memory component of $P_{\ell+1}$ form $\xi_\ell(T)$. Because the forward weights and gradients are assumed to be bounded by constant values, the latter can be factored out of the integral at each recursion step. Therefore, the integration repeatedly acts upon the time variable t, resulting in the factorial and polynomial structures in Equation 82. The lag term of Equation 81 is instead characterized by the tracking bound-constant D_ℓ, which recursively incorporates the bounds on the drift of the forward weights of the higher layers, defined as

$$D_\ell = A_\ell\left(D_{\ell+1} + C_{\ell+1}\right), \tag{83}$$

with $D_{L-1} = 0$. We can now see that, in the limit $T \to \infty$, the memory component of Equation 81 is suppressed by the exponential term $e^{-\eta_\Psi T}$, which eventually overcomes the polynomial $\xi_\ell(T)$, while the lag term converges to φD_ℓ:

$$\lim_{T\to\infty}\|P_\ell\|_2 \leq \lim_{T\to\infty}\left(e^{-\eta_\Psi T}\xi_\ell(T) + \varphi D_\ell\left(1 - e^{-\eta_\Psi T}\right)\right) = \varphi D_\ell. \tag{84}$$

By taking the limit $\varphi \to 0$, we therefore have that:

$$\lim_{\varphi\to 0}\lim_{T\to\infty}\|P_\ell\|_2 \leq \lim_{\varphi\to 0}\varphi D_\ell = 0, \quad \forall \ell \in \{1,\dots,L-1\}. \tag{85}$$

For the term $\|Q_\ell\|_2$, Equation 59 gives the general bound

$$\|Q_\ell\|_2 \leq \varphi C_\ell\left(1 - e^{-\eta_\Psi T}\right), \tag{86}$$

and taking first the limit $T \to \infty$ and subsequently $\varphi \to 0$, we obtain

$$\lim_{\varphi\to 0}\lim_{T\to\infty}\|Q_\ell\|_2 \leq \lim_{\varphi\to 0}\varphi C_\ell = 0, \quad \forall \ell \in \{1,\dots,L-1\}. \tag{87}$$

Finally, we conclude the proof by applying Equation 80, Equation 85, and Equation 87 to Equation 78, obtaining

$$\lim_{\varphi\to 0}\lim_{T\to\infty}\|\mathcal{E}_\ell(T)\|_2 = 0, \quad \forall \ell \in \{1,\dots,L-1\}. \tag{88}$$

This completes the proof showing that, as $T \to \infty$, a separation of timescales is a sufficient condition for the feedback matrices to converge to the chain of transposed forward weights throughout the

network, provided that both the gradients and the network parameters remain bounded. Following from Equation 88, we have that

$$\lim_{\varphi \to 0} \lim_{T \to \infty} \Psi_\ell = \Psi_\ell^\star, \tag{89}$$

demonstrating that (linear) BP's chain of transposed forward weights corresponds to the non-stationary fixed point for the dynamics of DFP's feedback matrices. In the limit of infinite timescale separation ($\varphi \to 0$), DFP updates converge exactly to BP updates under the linear regime, and under the nonlinear regime, this equivalence holds up to the local Jacobians of the activation functions. (Note that, in practice, for sufficiently small φ, this condition is already satisfied after a finite number of optimization steps and does not require taking the limits $T \to \infty$ and $\varphi \to 0$, as shown in Figure 2.)

A.2 Experimental Details

A.2.1 Training Setup

The experiments presented in Section 4 span image classification settings of increasing difficulty, from MNIST and Fashion-MNIST (LeCun, 2010; Xiao et al., 2017) to CIFAR-10, CIFAR-100 and Tiny ImageNet (Krizhevsky et al., 2009; Le et al., 2015). A 4-layer MLP and a VGG-7-like architecture are used to compare DFP against BP and the direct feedback algorithms described in Section 2.3, while the larger settings, VGG-9 and VGG-11 (Simonyan & Zisserman, 2015) and ResNet-20 (He et al., 2016) on CIFAR-100 and Tiny ImageNet, compare DFP against DFA, BP, and GrAPE. As no public implementation was available at the time of writing, we reimplemented GrAPE following Caillon et al. (2026), drawing the random perturbations in activity space. After hyperparameter search, our baseline matches or exceeds the accuracies reported by the original authors in all comparable settings, reaching 35.85% on ResNet-20 with CIFAR-100 against the 24.28% reported for the corresponding BP-free variant. Furthermore, we compare with its BP-free variant as the version with periodic use of BP in its updates reintroduces non-local learning.

All methods are implemented in Python using the PyTorch framework and trained on NVIDIA RTX A6000 GPUs. No normalization layers are used in the MLP and VGG-style networks, whereas the ResNet-20 experiments employ batch normalization, as it is integral to the residual architecture. Forward weights are optimized using the AdamW optimizer (Loshchilov & Hutter, 2017), while the NAdam optimizer (Dozat, 2016) is used for feedback weights. A cosine annealing learning rate schedule without warm restarts (Loshchilov & Hutter, 2016) is applied to both forward and feedback parameters. All datasets are standardized prior to training. For MNIST and Fashion-MNIST, no data augmentation is applied. For CIFAR-10 and CIFAR-100, standard data augmentation is used, including random cropping (size 32 with padding 4) and horizontal flipping with probability 0.5. Tiny ImageNet undergoes a similar augmentation pipeline, consisting of random cropping (size 64 with padding 8) followed by horizontal flipping with probability 0.5. In the MLP and VGG experiments, the batch size is fixed at 64. Training runs for 25 epochs on MNIST and Fashion-MNIST, 50 on CIFAR-10, 75 on CIFAR-100, and 125 on Tiny ImageNet. The activation functions are explored jointly with the hyperparameters, considering LeakyReLU and Tanh, except for DFP, which employs LeakyReLU in all MLP and VGG experiments. For the VGG experiments on Tiny ImageNet, all methods use LeakyReLU, except for DFA, which employs Tanh to avoid collapsing to chance accuracy. The ResNet-20 experiments follow a separate protocol, similar to that of Caillon et al. (2026): a ReLU activation for all methods, a batch size of 128, and 200 epochs with a multi-step learning rate schedule decaying by a factor of 0.2 at epochs 60, 120, and 160. The forward weights are optimized with SGD with momentum 0.9 for BP and with Adam (Kingma & Ba, 2015) for all other methods, while the feedback weights of DFP and GrAPE are optimized with SGD with momentum 0.9. Lastly, DFP networks use an exact initialization in all classification experiments, that is, orthogonal forward weights and feedback weights that minimize Equation 8.

A.2.2 Architectures and Hyperparameters

The architectures used in our classification experiments are detailed in Tables 4 and 5, and are identical across all algorithms. Table 6 reports the hyperparameters of the DFP networks (those of the remaining algorithms are available in our GitHub repository). Importantly, the ratio of the reported learning rates should not be read as the direct timescale separation φ of Theorem 1. Forward and feedback parameters are optimized with different optimizers, as previously described, whose

Table 4: Architectural details of the models used in the smaller experiments.

	MLP-4	VGG-7
Conv. channels	–	[64, 128, 256, 512]
Kernel sizes	–	[3, 3, 3, 3]
Strides	–	[1, 1, 1, 1]
Paddings	–	[1, 1, 1, 1]
Pool window	–	2×2
Pool stride	–	2
FC size	[512, 512, 512, output]	[512, 512, output]

Table 5: Architectural details of the models used in the larger experiments. H refers to the input image's height (32 for CIFAR-100 and 64 for Tiny ImageNet).

	VGG-9	VGG-11	ResNet-20
Conv. channels	[64, 128, 256×2, 512×2]	[64, 128×3, 256×2, 512×2]	[16, 16×6, 32×6, 64×6]
Kernel sizes	3	3	3
Strides	1	1	1, [1, 2, 2] at stage entry
Paddings	1	1	1
Pool window	2×2	2×2	global average
Pool stride	2	2	–
FC size	$[H \cdot 2^6, H \cdot 2^6$, output]	$[H \cdot 2^6, H \cdot 2^6$, output]	[output]

per-step behavior affects the update differently, so η_Θ/η_Ψ does not measure the effective ratio of the realized dynamics (for instance, the ResNet-20 experiments use SGD for the feedback update with a momentum factor of 0.9). Furthermore, since DFP is initialized exactly at the feedback equilibrium, the transient terms of the tracking bound disappear, and only the lag term remains.

A.2.3 Implicit Conditioning of the Feedback Pathway

As mentioned in Section 3, Equation 7 can result in a degenerate singular value spectrum, with many values vanishing and a tail growing large (Saxe et al., 2014). This behavior is illustrated in Figure 4 for a VGG-9 network trained for one epoch on CIFAR-100. The first row shows the ratio of the maximum and minimum singular values of each feedback matrix, when the orthogonality loss is either applied or not. When $\eta_\perp = 0$, the feedback matrices' singular spectrum is left free to evolve unconstrained, and even with orthogonal initialization of the network, it diverges to very large ratios throughout learning. When instead the orthogonality loss is applied ($\eta_\perp = 0.08$), the spectrum of Ψ_ℓ is implicitly penalized for growing large ratios, and this is reflected by the more uniform ratio values across layers. The second row shows how this affects the norm of the projected gradient. Each column mirrors the behavior of the singular-value ratio of the associated feedback matrices, delivering gradients with a largely varying magnitude when the orthogonality loss is not applied and with a more uniform one when it is applied. This in turn affects learning, resulting in less stable and poorer performance when the orthogonality loss is not applied, as shown in Table 3.

A.2.4 Parallelization Setup

Following Algorithm 1, since the feedback updates are layer-independent, we stack the per-layer matrices along an additional batch-like dimension so that a single batched matrix multiplication computes the terms for all layers at once (the same applies to additions and subtractions). The same approach is used for the forward weight updates and the feedback error projection. The PyTorch framework allows computing batched matrix multiplications via `torch.bmm()`, removing the need for loops and their associated kernel-launch overhead, allowing the GPU to process layers concurrently. Importantly, `torch.bmm()` does not guarantee fully parallel execution across layers. Thus, our results should be interpreted as a lower bound on the actual speedup that DFP can achieve over BP in this setting. Across multiple GPUs, both methods are data-parallel, with DFP additionally leveraging its model-parallelizability.

BP is distributed via `DistributedDataParallel`, which reduces the gradients, overlapping with the backward pass as the latter proceeds. This is complemented by a ZeRO-style sharded

Table 6: Hyperparameters employed for the DFP networks in our experiments.

Dataset	**Model**	$\eta_\perp$	fb-lr	fb-decay	fw-lr	fw-decay
MNIST	MLP-4	2.781e−1	1.170e−4	8.220e−3	6.738e−4	8.402e−5
Fashion-MNIST	MLP-4	4.439e−1	1.270e−4	5.919e−4	7.426e−4	4.752e−5
CIFAR-10	MLP-4	1.027e−3	4.371e−2	3.037e−4	5.085e−4	2.048e−3
MNIST	VGG-7	2.361e−1	9.986e−2	2.757e−4	4.216e−4	1.237e−6
Fashion-MNIST	VGG-7	4.445e−1	5.554e−4	1.695e−4	3.801e−4	2.717e−3
CIFAR-10	VGG-7	3.888e−3	1.308e−2	1.399e−3	3.614e−4	1.282e−3
CIFAR-100	VGG-9	8.249e−2	1.007e−3	5.364e−6	2.204e−4	1.095e−4
CIFAR-100	VGG-11	1.105e−1	1.030e−3	4.512e−5	1.608e−4	8.553e−7
Tiny ImageNet	VGG-9	2.104e−1	1.292e−3	1.928e−5	1.558e−4	5.775e−7
Tiny ImageNet	VGG-11	9.424e−1	5.677e−4	5.330e−4	1.373e−4	1.735e−3
CIFAR-100	ResNet-20	4.713e−4	4.648e−3	1.244e−6	7.198e−4	3.036e−3
Tiny ImageNet	ResNet-20	2.145e−3	7.717e−4	1.963e−3	8.660e−4	3.498e−6

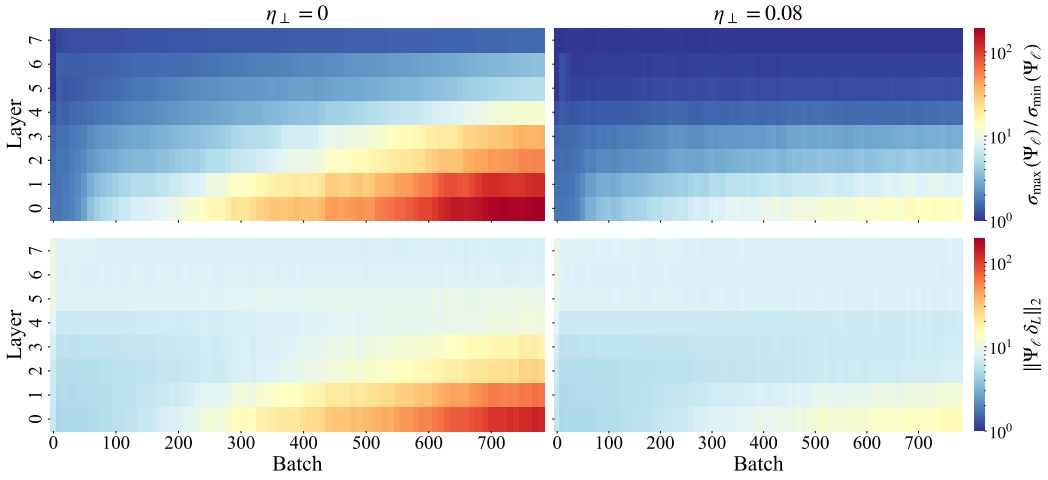

Figure 4: The first row shows the ratio between the maximum and minimum singular values of each feedback matrix of a VGG-9 network trained for one epoch on CIFAR-100. When the orthogonality loss is not applied (first column), the ratio varies wildly across layers. Applying the orthogonality loss (second column), instead, keeps the ratio substantially more uniform across layers. The second row shows the norm of the projected gradient for the previously described scenarios, with each column mirroring the behavior of the singular-value ratio of the associated feedback matrices.

optimizer (Rajbhandari et al., 2020), where parameters and gradients reside in flat buffers, with each GPU updating only its own contiguous shard, and a single in-place all-gather rebuilds the full parameter set.

DFP instead combines both data- and model-parallelism. Every GPU holds a full copy of the forward weights and processes its own local batch, while the stacked per-layer matrices are sharded, with each GPU owning and updating a contiguous slice of layers. After the forward pass, the activities and output errors are exchanged asynchronously, so that each GPU obtains the contributions of all local batches for the layers it owns. While this exchange is in flight, each GPU computes the orthogonality term of its owned layers, which depends only on locally held weights and requires no communication. The activities are then used to compute the forward-weight gradients of the owned layers over the global batch in a single batched matrix multiplication. The orthogonality gradient is added in place, and the optimizer updates only the owned slice. A final all-gather rebuilds the full updated weight set on every GPU, so that no gradient is ever exchanged. The sharded feedback matrices are instead updated at the beginning of each step, with the exchange of the updated shards hidden behind the forward pass.

For both DFP and BP, the update computations are compiled, reducing the kernel-launch overhead and yielding cleaner timing measurements. To ensure a fair comparison, each epoch is timed between two synchronization fences (`torch.distributed.barrier()` followed by `torch.cuda.synchronize()`), with all batches preloaded to device memory beforehand. All experiments run on four H100 NVIDIA GPUs with a local batch size of 256 (1024 globally).