Two Tales of Single-Phase Contrastive Hebbian Learning

Rasmus Høier [1]   **Christopher Zach** [1]

Abstract

The search for "biologically plausible" learning algorithms has converged on the idea of representing gradients as activity differences. However, most approaches require a high degree of synchronization (distinct phases during learning) and introduce substantial computational overhead, which raises doubts regarding their biological plausibility as well as their potential utility for neuromorphic computing. Furthermore, they commonly rely on applying infinitesimal perturbations (nudges) to output units, which is impractical in noisy environments. Recently it has been shown that by modelling artificial neurons as dyads with two oppositely nudged compartments, it is possible for a fully local learning algorithm named "dual propagation" to bridge the performance gap to backpropagation, without requiring separate learning phases or infinitesimal nudging. However, the algorithm has the drawback that its numerical stability relies on symmetric nudging, which may be restrictive in biological and analog implementations. In this work we first provide a solid foundation for the objective underlying the dual propagation method, which also reveals a surpising connection with adversarial robustness. Second, we demonstrate how dual propagation is related to a particular adjoint state method, which is stable regardless of asymmetric nudging.

1. Introduction

Credit assignment using fully local alternatives to back-propagation is interesting both as potential models of biological learning as well as for their applicability for energy efficient analog neuromorphic computing (Kendall & Kumar, 2020; Yi et al., 2022). A pervasive idea in this field is the idea of representing error signals via activity differences, referred to as NGRAD (Neural Gradient Representation by Activity Differences) approaches (Lillicrap et al., 2020). However, a commonly overlooked issue in NGRAD approaches is the requirement of applying infinitesimal perturbations (or *nudging*) to output neurons in order to propagate error information through the network. This is problematic as analog and biological neural networks are inherently noisy, potentially causing the error signal to vanish if insufficient nudging is applied. In many local learning methods the output units are *positively* nudged to reduce a target loss, but utilizing additional negative nudging (output units increase a target loss) can be beneficial to improve accuracy (e.g. (Laborieux et al., 2021)).

The vanishing error signal problem is addressed by coupled learning (Stern et al., 2021), which proposes to replace the clamped output units of contrastive Hebbian learning with a convex combination of the label and the free phase outputs. Unfortunately, coupled learning has been shown to perform worse than equilibrium propagation on CIFAR10 and CIFAR100 (Scellier et al., 2023), and it does not necessarily approximate gradient descent on the output loss function (Stern et al., 2021). Holomorphic equilibrium propagation (Laborieux & Zenke, 2022; 2023) mitigates the need for infinitesimal nudging required in standard equilibrium propagation (Scellier & Bengio, 2017) at the cost of introducing complex-valued parameters. Whether this is a suitable approach for either biological or analog neural networks is an open question. Dual propagation (Høier et al., 2023) (DP), an algorithm similar in spirit to contrastive Hebbian learning, equilibrium propagation and coupled learning, is compatible with non-infinitesimal nudging by default. This method infers two sets of oppositely nudged and mutually tethered states simultaneously. However, utilization of symmetric nudging is a necessary condition for the convergence of its inference step.

Contributions   DP is compatible with strong feedback and only requires a single inference phase, which are appealing features with regards to biological plausibility and potential applications to analog neuromorphic computing. However, the lack of convergence guarantees in the case of asymmetric nudging is clearly unsettling as exact symmetry is hard to realize outside of digital computers. Further,—unlike digital computers—neuromorphic, analog or otherwise highly

---

[1]Department of Electrical Engineering, Chalmers University of Technology, Sweden. Correspondence to: Rasmus Høier <hier@chalmers.se>.

distributed computing hardware typically performs continuous computations and runs asynchronously. Consequently, numerical stability of an energy-based inference and learning method is of essential importance. For this reason we derive an improved variant of dual propagation, which overcomes this strict symmetry requirement. In summary the contributions of this work are:

- A derivation of DP based on repeated relaxations of the optimal value reformulation (Outrata, 1988; Dempe & Zemkoho, 2013).
- A new Lagrangian based derivation of dual propagation, which recovers the original dual propagation algorithm in the case of symmetric nudging, but leads to a slightly altered (and more robust) method in the case of asymmetric nudging.
- We experimentally investigate the robustness of these algorithms to asymmetric nudging and strong feedback, and we further demonstrate the impact of asymmetric nudging on the estimated Lipschitz constant.

2. Related Work

CHL, EP and lifted networks In contrastive Hebbian learning (CHL) (Movellan, 1991; Xie & Seung, 2003) and equilibrium propagation (EP) (Scellier & Bengio, 2017) neuronal activations are found via an energy minimization procedure. Inference is carried out twice, once with and once without injecting label information at the output layer. While CHL clamps the output units to the true targets, EP applies nudging towards a lower loss. The difference between the activity in each of these two inference phases is used to represent neuronal error vectors. To speed up convergence and ensure that inferred states represent the same local energy basin, this is typically (but not always) done sequentially, e.g. the second inference phase is initialized with the solution found in the first phase. A better gradient estimate can be obtained by introducing an additional oppositely nudged inference phase, yielding a three-phased algorithm (Laborieux et al., 2021).

Dual propagation (Høier et al., 2023) (DP) is another algorithm in the NGRAD family, in which each neuron has two intrinsic states corresponding to positively and negatively nudged compartments as illustrated in Fig. 1. The algorithm is based on a specific network potential and uses the respective stationarity conditions to obtain an easy-to-implement inference method: each neuron maintains two internal states representing the neural activity as (weighted) mean and the error signal as difference, respectively (which makes it an instance of an NGRAD method). The mean state is sent "upstream" to the next layer while the difference is passed downstream to the preceding layer, where it nudges the receiving neural state. The method essentially interleaves or "braids" the two inference phases and makes it possible to
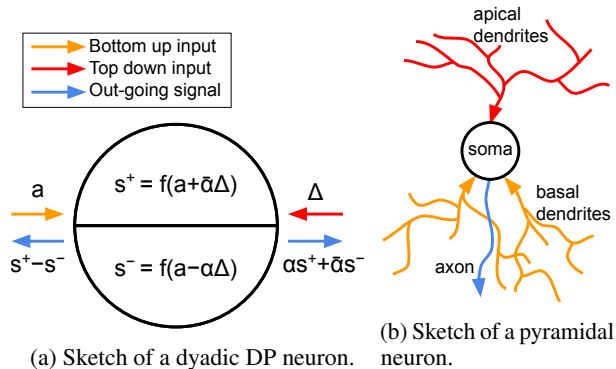

(a) Sketch of a dyadic DP neuron. (b) Sketch of a pyramidal neuron.

Figure 1. **(a)** Illustration of a dyadic neuron (note that all quantities are scalar). The two internal states, s^+ and s^-, receive the same bottom-up input a but the top down input Δ nudges them in opposite directions. The difference and weighted mean of these internal states are then propagated downstream and upstream respectively. **(b)** In a pyramidal neuron bottom-up signal arrive at the basal dendrites and top-down signal arrive at the apical dendrites. Concerns regarding DP and biological plausibility are discussed in section 7.

infer both states simultaneously. When the update sequence is chosen appropriately, as few as two updates per neuron are sufficient, making the algorithm comparable to back-propagation in terms of runtime and 10-100X faster than CHL and EP. In practice, DP is applied to feed forward networks and EP typically to Hopfield models, which has implications for how inference is carried out (fixed-point method vs energy minimization) and on the hardware the algorithms are suitable for. Another difference to EP and CHL is, that dual propagation infers both sets of states simultaneously (and not in sequential phases). The underlying network potential used in DP is parametrized by a coefficient $\alpha \in [0, 1]$ determining the weighted mean. The stability of the resulting inference method hinges on choosing $\alpha = 1/2$.

Casting deep learning as an optimization task over explicit activations and weights is the focus of a diverse set of back-propagation alternatives sometimes collectively referred to as *lifted neural networks* (Carreira-Perpinan & Wang, 2014; Askari et al., 2018; Gu et al., 2020; Li et al., 2020; Choromanska et al., 2019; Zach & Estellers, 2019; Høier & Zach, 2020). Predictive coding networks (e.g. (Whittington & Bogacz, 2017; Salvatori et al., 2023)) can be also understood as instances of lifted neural networks. Although members of this group have different origins, they are algorithmically closely related to CHL and EP (Zach, 2021), but vary e.g. in their suitability for digital hardware.

Weak and strong feedback While a number of CHL-inspired learning methods for neural networks are shown to be equivalent to back-propagation when the feedback parameter β approaches zero (i.e. infinitesimal nudging takes place, as discussed in e.g. (Xie & Seung, 2003; Scellier &

Bengio, 2017; Zach & Estellers, 2019; Zach, 2021)), practical implementations use a finite but small value for β, whose magnitude is further limited—either explicitly or implicitly. CHL implicitly introducing weak feedback via its layer discounting in order to approximate a feed-forward neural network, and both CHL and EP rely on weak feedback to stay in the same energy basin for the free and the clamped solutions. The iterative solver suggested for the LPOM model (Li et al., 2020) also depends on sufficiently weak feedback to ensure convergence of the fixed-point inference scheme. In contrast to these restrictions, the feedback parameter β in dual propagation is weakly constrained and its main effect is to influence the resulting finite difference approximation for activation function derivatives.

3. Background

A feed-forward network is a composition of L layer computations $s_{k-1} \mapsto f_k(W_{k-1}s_{k-1})$, where f_k is the activation function of layer k and W_{k-1} is the associated weight matrix. s_0 is the input provided to the network and $s_k := f_k(W_{k-1}s_{k-1})$ is the vector of neural states in layer k. The set of weight matrices $\theta = \{W_k\}_{k=1}^{L-1}$ form the trainable parameters, which are optimized during the training stage to minimize a target loss ℓ. Training is usually—at least on digitial hardware—conducted via back-propagation of errors, which is a specific instance of the chain rule.

In (Høier et al., 2023) a localized, derivative-free approach for supervised learning in feed-forward neural networks is proposed, which is based on the following network potential $\mathcal{L}_\alpha^{DP}$ (Eq. 1) for a parameter $\alpha \in [0,1]$,

$$\mathcal{L}_\alpha^{DP}(\theta) = \min_{s^+} \max_{s^-} \alpha\ell(s_L^+) + \bar{\alpha}\ell(s_L^-)$$
$$+ \tfrac{1}{\beta}\sum_{k=1}^{L}\left(E_k(s_k^+, \bar{s}_{k-1}) - E_k(s_k^-, \bar{s}_{k-1})\right). \tag{1}$$

The terms E_k in $\mathcal{L}_\alpha^{DP}$ are specifically chosen as $E_k(s_k, s_{k-1}) = G_k(s_k) - s_k^\top W_{k-1}s_{k-1}$, where G_k is typically a strongly convex function for the "resting" energy (and relates to the activation function via $\nabla G_k = f_k^{-1}$ and $\nabla G_k^* = f_k$). This choice leads—via the corresponding stationarity conditions—to particularly simple fixed-point updates for $s_k^\pm$ for the inference of neural states,

$$s_k^+ \leftarrow f_k\big(W_{k-1}\bar{s}_{k-1} + \alpha W_k^\top(s_{k+1}^+ - s_{k+1}^-)\big)$$
$$s_k^- \leftarrow f_k\big(W_{k-1}\bar{s}_{k-1} - \bar{\alpha}W_k^\top(s_{k+1}^+ - s_{k+1}^-)\big), \tag{2}$$

where $\bar{\alpha} := 1 - \alpha$ and $\bar{s}_k := \alpha s_k^+ + \bar{\alpha}s_k^-$. The state of the output units is determined by solving

$$s_L^+ \leftarrow \arg\min_{s_L} \alpha\ell(s_L) + E_L(s_L, \bar{s}_{L-1})$$
$$s_L^- \leftarrow \arg\min_{s_L} -\bar{\alpha}\ell(s_L) + E_L(s_L, \bar{s}_{L-1}). \tag{3}$$

Gradient-based learning of the weight matrices $\theta = (W_k)_{k=0}^{L-1}$ is based on the quantity $\partial_{W_k}\mathcal{L}_\alpha^{DP} \propto (s_{k+1}^+ - s_{k+1}^-)\bar{s}_k^\top$ once the neural states $s_k^\pm$ are inferred.

The choice of $\alpha = 1$ is equivalent to the LPOM formulation (Li et al., 2020), that builds on the following network potential,

$$\mathcal{L}^{LPOM}(\theta) = \min_{s^+} \ell(s_L^+)$$
$$+ \tfrac{1}{\beta}\sum_{k=1}^{L}\left(E_k(s_k^+, s_{k-1}^+) - E_k(f_k(W_{k-1}s_{k-1}^+), s_{k-1}^+)\right). \tag{4}$$

Eq. 4 can be obtained from Eq. 1 by closed form maximization w.r.t. all s_k^-, as s_k^- in this setting solely depends on s_{k-1}^+. Analogously, the choice $\alpha = 0$ can be interpreted as "inverted" LPOM model (after minimizing w.r.t. all s_k^+ variables). Unfortunately, the straightforward update rules in Eq. 2 lead to a stable method only when $\alpha = 1/2$. In Section F we discuss the necessary adaptation of Eq. 2 (which are also used in our experiments) when $\alpha \in \{0, 1\}$. In this work we (i) provide explanations why the choice $\alpha = 1/2$ is special (via connecting it to the adjoint state method in Section 5 and via a more explicit analysis in Section G) and (ii) demonstrate why selecting $\alpha \neq 1/2$ can be beneficial.

4. A Relaxation Perspective on Dual Propagation

In this section we show how $\mathcal{L}_\alpha^{DP}$ can be obtained exactly by repeated application of certain relaxations of the optimal value reformulation (Outrata, 1988; Dempe & Zemkoho, 2013) (summarized in Section 4.1). Obtaining $\mathcal{L}_\alpha^{DP}$ requires the application of two (to our knowledge) novel relaxations for bilevel programs as described in Sections 4.2 and 4.3.

4.1. The Optimal Value Reformulation and its Relaxation

First we describe a basic (and known) reformulation for a class of bilevel minmization problems. We consider a general bilevel program of the form

$$\min_\theta \ell(s^*) \qquad \text{s.t. } s^* = \arg\min_s E(s; \theta) \tag{5}$$

In the context of this work we assume that the minimizer of $E(s; \theta)$ is unique for all θ, which corresponds to layer computation in deep neural networks to be proper (single-valued) functions. In a first step the constraint in Eq. 5 can restated as

$$\min_{\theta, s} \ell(s) \qquad \text{s.t. } E(s; \theta) \leq \min_{s'} E(s'; \theta), \tag{6}$$

which is the *optimal value reformulation* (Outrata, 1988; Dempe & Zemkoho, 2013). As $E(s; \theta) \geq \min_{s'} E(s'; \theta)$

by construction, the inequality constraint is always (possibly only weakly) active. By fixing the corresponding non-negative Lagrange multiplier to $1/\beta$ for a $\beta > 0$ we obtain a *relaxed optimal value reformulation* ("ROVR"),

$$\min_{\theta, s} \ell(s) + \tfrac{1}{\beta}\left(E(s;\theta) - \min_{s'} E(s';\theta)\right)$$
$$= \min_{\theta, s} \max_{s'} \ell(s) + \tfrac{1}{\beta}\left(E(s;\theta) - E(s';\theta)\right). \tag{7}$$

Repeated application of this conversion from lower-level minimization task to higher-level penalizers on deeply nested problems yields a large variety of known learning methods (Zach, 2021). When $\beta \to 0$, a number of works have shown the connection between Eq. 7, back-propagation and implicit differentiation (Xie & Seung, 2003; Scellier & Bengio, 2017; Gould et al., 2019; Zach, 2021).

4.2. A Saddlepoint Relaxed Optimal Value Reformulation

A different relaxation for a bilevel program can be obtained by rewriting Eq. 5 as follows,

$$\min_{s^+} \max_{s^-} \ell(\alpha s^+ + \bar{\alpha} s^-) \quad \text{s.t. } s^+ = \arg\min_s E(s;\theta)$$
$$s^- = \arg\min_s E(s;\theta), \tag{8}$$

where $\alpha \in [0,1]$ and $\bar{\alpha} = 1 - \alpha$ as before. The reformulation above replaces the solution s^* of the inner problem with an (apparently superfluous) convex combination of two related solutions s^+ and s^-. Under the uniqueness assumption on the minimizer of $E(\cdot;\theta)$, this program is equivalent to Eq. 5 (as $s^* = s^+ = s^-$ by construction). The main effect of introducing $s^\pm$ in Eq. 8 is, that it allows two independent applications of the ROVR. Applying the standard ROVR on s^- (a maximization task) yields the first relaxation,

$$\min_{s^+} \max_{s^-} \ell(\alpha s^+ + \bar{\alpha} s^-) - \tfrac{1}{\beta}(E(s^-;\theta) - \min_s E(s;\theta))$$
$$\text{s.t. } s^+ = \arg\min_s E(s;\theta), \tag{9}$$

and a second application of the ROVR on s^+ (with the same multiplier $1/\beta$) results in

$$J_{\alpha,\beta}^{\text{SPROVR}}(\theta) := \min_{s^+} \max_{s^-} \ell(\alpha s^+ + \bar{\alpha} s^-)$$
$$+ \tfrac{1}{\beta}(E(s^+;\theta) - E(s^-;\theta)) \tag{10}$$

as the $\min_s E(s;\theta)$ terms cancel each other. We can reverse the order of min and max in Eq. 8, which also induces a reversal of min and max in Eq. 10. Strong duality may not be satisfied in general, but the stationarity conditions w.r.t. s^+ and s^- are unaffected by permuting min and max. We call the algebraic conversion from Eq. 5 to Eq. 10 the *saddle-point ROVR or SPROVR*.

In order to model deep feed-forward networks we extend the bilevel program to a multi-level program with with L

nesting levels,

$$\min_{\theta} \ell(s_L^*) \quad \text{s.t. } s_k^* = \arg\min_{s_k} E_k(s_k, s_{k-1}^*; \theta) \tag{11}$$
$$s_1^* = \arg\min_{s_1} E_1(s_1; \theta),$$

where each $E_k(\cdot, s_{k-1}; \theta)$ is assumed to have a unique minimizer for all s_{k-1} and θ. By applying SPROVR repeatedly from the outermost level to inner ones, we arrive at the following relaxed reformulation,

$$\min_{\theta} \min_{\{s_k^+\}} \max_{\{s_k^-\}} \ell(\bar{s}_L) + \tfrac{1}{\beta}\left(E_1(s_1^+;\theta) - E_1(s_1^-;\theta)\right)$$
$$+ \tfrac{1}{\beta}\sum_{k=2}^L \left(E_k(s_k^+; \bar{s}_{k-1}; \theta) - E_k(s_k^-; \bar{s}_{k-1};\theta)\right). \tag{12}$$

We recall that the short-hand notation $\bar{s}_k$ stands for $\bar{s}_k = \alpha s_k^+ + \bar{\alpha} s_k^-$, but the choice of the coefficient α can in principle be layer-dependent (by replacing α with α_k). Similarly, the value of multiplier $1/\beta$ may depend on the layer k. Traversing the SPROVR from the inside towards the outermost level leads to a different (and less relevant) relaxation.

4.3. An Adversarial Relaxed Optimal Value Reformulation

In this section we present a different instance of an optimal value reformulation, that yields centered approaches to contrastive Hebbian learning and is further closely linked to adversarial training. We start with the bilevel program,

$$J_{\alpha,\beta}^{\text{adv}}(\theta) = \ell(s^*) \quad \text{s.t. } s^* = \arg\min_s E(s;\theta) - \bar{\alpha}\beta\ell(s), \tag{13}$$

where $\beta > 0$ is a step size parameter and $\alpha \in [0,1]$ further modulates the step length. The solution s^* is a minimizer of an adversarially perturbed inner problem (and assumed to exist), and therefore Eq. 13 is generally not equivalent to Eq. 5. A simple illustrative example is given by setting $E(s;\theta) = \|s - \theta\|^2/2$ and $\ell(s) = g^\top s$, then $\arg\min_s E(s;\theta) = \theta$ but s^* in Eq. 13 is given by $s^* = \theta + \bar{\alpha}\beta g$, i.e. s^* takes a gradient ascent step from $\theta = \arg\min_s E(s;\theta)$ to increase the (linear) target loss ℓ (somewhat in analogy with the fast gradient method). On the other hand, the outer problem aims to reduce the main loss ℓ for this perturbed sample s^*.

Using the optimal value reformulation (cf. Section 4.1) this is equivalent to an inequality constrained program,

$$\min_{s^*} \ell(s^*)$$
$$\text{s.t. } E(s^*;\theta) - \bar{\alpha}\beta\ell(s^*) \leq \min_s E(s;\theta) - \bar{\alpha}\beta\ell(s). \tag{14}$$

Fixing the (non-negative) multiplier of the Lagrangian relaxation to $1/\beta$ yields (after algebraic manipulations)

$$J_{\alpha,\beta}^{\text{AROVR}}(\theta) := \min_{s^+} \max_{s^-} \alpha\ell(s^+) + \bar{\alpha}\ell(s^-)$$
$$+ \tfrac{1}{\beta}\left(E(s^+;\theta) - E(s^-;\theta)\right), \tag{15}$$

which e.g. generalizes the centered variant of EP. Increasing the effective step length $\bar{\alpha}\beta$ in $J_{\alpha,\beta}^{\mathrm{adv}}$ by increasing $\bar{\alpha}$ allows larger adversarial perturbations and therefore makes the learning task harder. This property carries over to $J_{\alpha,\beta}^{\mathrm{AROVR}}$ as shown in the following proposition (proven in the appendix).

Proposition 4.1. *Let $0 < \beta \le \beta'$ and $\alpha, \alpha' \in [0,1]$ with $\alpha \le \alpha'$. Then the following holds: (a) $J_{\alpha,\beta}^{AROVR}(\theta) \le J_{\alpha',\beta}^{AROVR}(\theta)$ and (b) $\beta J_{\alpha,\beta}^{AROVR}(\theta) \le \beta' J_{\alpha,\beta'}^{AROVR}(\theta)$.*

Original EP ($\alpha = 1$) and centered EP (Laborieux et al., 2021; Scellier et al., 2023) ($\alpha = 1/2$) are two particular instances of $J_{\alpha,\beta}^{\mathrm{AROVR}}$. Hence, Prop. 4.1 raises the question of how much the better gradient estimate and how much the stricter loss $J_{1/2,\beta}^{\mathrm{AROVR}}$ contribute to the advantages of centered EP.

Without further assumption we cannot expect claim (b) to be strengthened to $J_{\alpha,\beta}^{\mathrm{AROVR}}(\theta) \le J_{\alpha,\beta'}^{\mathrm{AROVR}}(\theta)$ as a larger adversarial step is offset by a smaller multiplier $1/\beta'$. If the total step size $\bar{\alpha}\beta$ remains constant, then a stronger relation can nevertheless be shown:

Proposition 4.2. *Let $0 < \beta' < \beta$ and α' such that $\bar{\alpha}\beta = \bar{\alpha}'\beta'$. Then $J_{\alpha,\beta}^{AROVR}(\theta) \le J_{\alpha',\beta'}^{AROVR}(\theta)$.*

The two relaxations, $J_{\alpha,\beta}^{\mathrm{AROVR}}$ and $J_{\alpha,\beta}^{\mathrm{SPROVR}}$ (Eq. 10) look very similar, the only difference being the location of the averaging step (averaging the argument or the function value of the target loss ℓ). Generally we expect $J_{\alpha,\beta}^{\mathrm{AROVR}}$ to upper bound $J_{\alpha,\beta}^{\mathrm{SPROVR}}$, which is always the case when ℓ is convex. The huge advantage of SPROVR is that it leads to tractable objectives when applied to deeply nested problems (linear in the problem depth L). Applying AROVR in a similar manner on Eq. 11 yields an exponential number of terms in the resulting objective unless $\alpha = 1$ or $\alpha = 0$ (where it actually coincides with SPROVR). We now can easily identify $\mathcal{L}_\alpha^{DP}$ (Eq. 1) as result of applying both SPROVR and AROVR on a deeply nested program:

Corollary 4.3. *The DP objective $\mathcal{L}_\alpha^{DP}$ (Eq. 1) can be obtained by first applying AROVR on the outermost level of Eq. 11, followed by subsequent applications of SPROVR for the remaining levels.*

In Section 6.1 we verify whether different choices for α have any impact on the robustness of a trained model. Since any differences are expected to vanish for small choices of β, the large β setting (we choose $\beta = 1/2$) is of main interest. Note that unlike standard adversarial training (which perturbs solely the input to the network), the lifted network potential $\mathcal{L}_\alpha^{DP}$ is based on perturbations of the network's hidden activations. Adversarial perturbations of the inputs can be achieved by adding an auxiliary constraint $s_0 = x$, e.g. by extending $\mathcal{L}_\alpha^{DP}$ with e.g. $E_0(s_0, x) = \|s_0 - x\|^2$.

In our experiments, a significant reduction of the Lipschitz estimate is in fact observed when decreasing α (and there-

fore increasing the "adversarial" step size $\bar{\alpha}\beta$). We do of course not suggest to replace advanced adversarial training with the loss $J_{0,\beta}^{\mathrm{AROVR}}$, but the main goal of this section is to demonstrate that a non-infinitesimal choice of β has a theoretical and practical impact and therefore goes beyond a pure approximation of back-propagation (respectively implicit differentiation). One may also speculate that the hypothesized NGRAD-based learning in biological systems gains some robustness by leveraging a similar mechanism.

5. A Lagrangian Perspective on Dual Propagation

In this section we derive a variant of dual propagation, which we refer to as $\mathrm{DP}^\top$, which turns out to be robust to asymmetric nudging (i.e. choosing $\alpha \in [0,1]$ not necessarily equal to $1/2$). Our starting point is a modified version of LeCun's classic Lagrangian-based derivation of backpropagation (Lecun, 1988). We assume (i) that the activation functions f_k are all invertible (which can be relaxed), and (ii) that f_k has a symmetric derivative (i.e. $f_k'(x) = f_k'(x)^\top$). The second assumption clearly holds e.g. for element-wise activation functions. Our initial Lagrangian relaxation can now be stated as follows,

$$
\begin{aligned}
\mathcal{L}(\theta) = \min_s \max_\delta \; & \ell(s_L) \\
& + \sum_{k=1}^L \delta_k^\top \left(f_k^{-1}(s_k) - W_{k-1}s_{k-1} \right).
\end{aligned}
\tag{16}
$$

Note that the multiplier δ_k corresponds to the constraint $f_k^{-1}(s_k) = W_{k-1}s_{k-1}$ (in contrast to the constraint $s_k = f_k(W_{k-1}s_{k-1})$ employed in (Lecun, 1988)). The main step is to reparamtrize s_k and δ_k in terms of s_k^+ and s_k^-,

$$
s_k = \alpha s_k^+ + \bar{\alpha}s_k^- \qquad \delta_k = s_k^+ - s_k^- \tag{17}
$$

for a parameter $\alpha \in [0,1]$ (and $\bar{\alpha} := 1-\alpha$). In the following we use the short-hand notations $\bar{s}_k := \alpha s_k^+ + \bar{\alpha}s_k^-$ and $a_k := W_{k-1}\bar{s}_{k-1}$.

Proposition 5.1. *Assume that the activation functions f_k, $k = 1, \ldots, L-1$, are invertible, have symmetric derivatives and behave locally linear. Then the Lagrangian corresponding to the reparametrization in Eq. 17 is given by*

$$
\begin{aligned}
\mathcal{L}_\alpha^{DP^\top}(\theta) = \min_{s^+} \max_{s^-} \; & \ell(\bar{s}_L) \\
& + \sum_{k=1}^L (s_k^+ - s_k^-)^\top \left(f_k^{-1}(\bar{s}_k) - W_{k-1}\bar{s}_{k-1} \right),
\end{aligned}
\tag{18}
$$

and the optimal $s_k^\pm$ in (18) satisfy

$$
\begin{aligned}
s_k^+ &\leftarrow f_k\left(W_{k-1}\bar{s}_{k-1} + \bar{\alpha}W_k^\top(s_{k+1}^+ - s_{k+1}^-)\right) \\
s_k^- &\leftarrow f_k\left(W_{k-1}\bar{s}_{k-1} - \alpha W_k^\top(s_{k+1}^+ - s_{k+1}^-)\right)
\end{aligned}
\tag{19}
$$

for internal layers $k = 1, \ldots, L-1$.

For the output layer activities $s_L^\pm$ we absorb f_L into the target loss (if necessary) and therefore solve

$$\min_{s_L^+} \max_{s_L^-} \ell(\bar{s}_L) + (s_L^+ - s_L^-)^\top (\bar{s}_L - a_L) \qquad (20)$$

with corresponding necessary optimality conditions

$$0 = \alpha \ell'(\bar{s}_L) + \bar{s}_L - a_L + \alpha(s_L^+ - s_L^-)$$
$$0 = \bar{\alpha} \ell'(\bar{s}_L) - \bar{s}_L + a_L + \bar{\alpha}(s_L^+ - s_L^-). \qquad (21)$$

Adding these equations reveals $\ell'(\bar{s}_L) + s_L^+ - s_L^- = 0$. Inserting this in any of the equations in (21) also implies $\bar{s}_L = a_L = W_{L-1}\bar{s}_{L-1}$. Thus, by combining these relations and with $g := \ell'(W_{L-1}\bar{s}_{L-1})$, the updates for $s_L^\pm$ are given by

$$s_L^+ \leftarrow W_{L-1}\bar{s}_{L-1} - \bar{\alpha}g \quad s_L^- \leftarrow W_{L-1}\bar{s}_{L-1} + \alpha g. \quad (22)$$

For the β-weighted least-squares loss, $\ell(s_L) = \frac{\beta}{2}\|s_L - y\|^2$, the above updates reduce to $s_L^+ \leftarrow a_L - \bar{\alpha}\beta(a_L - y)$ and $s_L^- \leftarrow a_L + \alpha\beta(a_L - y)$.

The neural activities $s_k^\pm$ together encode the forward and the adjoint state. Since the updates in Eq. 19 are based on the adjoint state method, we denote the resulting algorithm adjoint-DP or just DP$^\top$.

Relation to the original dual propagation The update equations in (19) coincide with the original dual propagation rules if $\alpha = 1/2$ (Høier et al., 2023), although the underlying objectives $\mathcal{L}_\alpha^{\text{DP}^\top}$ (18) and $\mathcal{L}_\alpha^{\text{DP}}$ are fundamentally different. If $\alpha \neq 1/2$, then α and $\bar{\alpha}$ switch places w.r.t. the error signals (but not in the definition of $\bar{s}$) compared to the original dual propagation method.[1] The updates in (19) for $\alpha = 0$ correspond to an algorithm called "Fenchel backpropagation" discussed in (Zach, 2021).

Both the objective of the original dual propagation (1) and the objective of the improved variant (18) can be expressed in a general contrastive form, but the underlying potentials E_k are somewhat different,

DP: $E_k(s_k^+, \bar{s}_{k-1}) - E_k(s_k^-, \bar{s}_{k-1})$
$$= G_k(s_k^+) - G_k(s_k^-) - (s_k^+ - s_k^-)^\top W_{k-1}\bar{s}_{k-1}$$

DP$^\top$: $E_k(s_k^\pm, \bar{s}_{k-1}) - E_k(s_k^\mp, \bar{s}_{k-1})$
$$= (s_k^+ - s_k^-)^\top \nabla G_k(\bar{s}_k) - (s_k^+ - s_k^-)^\top W_{k-1}\bar{s}_{k-1}.$$

Here G_k is a convex mapping ensuring that $\arg\min_{s_k} E_k(s_k, s_{k-1})$ provides the desired activation function f_k. The relation between f_k and G_k is given by $f_k = \nabla G_k^*$ and $f_k^{-1} = \nabla G_k$ (where G_k^* is the convex

conjugate of G_k). Activations function induced by G_k have automatically symmetric Jacobians as $f_k' = \nabla^2 G_k^*$ under mild assumptions. The main difference between these two flavors of DP can be restated as follows,

$$\Delta E_k = G_k(s_k^+) - G_k(s_k^-) - (s_k^+ - s_k^-)^\top \nabla G_k(\bar{s}_k)$$
$$= D_{G_k}(s_k^+ \| \bar{s}_k) - D_{G_k}(s_k^- \| \bar{s}_k), \qquad (23)$$

where D_{G_k} is the Bregman divergence induced by G_k. The difference of Bregman divergences can have any sign, and $\Delta E_k = 0$ if $D_{G_k}(s_k^+ \| \bar{s}_k) = D_{G_k}(s_k^- \| \bar{s}_k)$. Hence, for the choice $\alpha \in \{0, 1\}$ (i.e. $\bar{s}_k = s_k^+$ or $\bar{s}_k = s_k^-$), ΔE_k can only be zero when $s_k^+ = s_k^-$. Fixing $\alpha = 1/2$ and G_k to any convex quadratic function implies $\Delta E_k = 0$ (as D_{G_k} becomes the squared Mahalanobis distance in this setting).

Non-invertible activation function If f_k is not invertible at the linearization point (such as the softmax function), then D_k is singular and the constraint $f_k^{-1}(s_k) = W_{k-1}s_{k-1}$ is converted into a constraint that $D_k^+ s_k$ is restricted to a linear subspace,

$$D_k^+(s_k - s_k^0) = W_{k-1}s_{k-1} + N_k v_k, \qquad (24)$$

where N_k spans the null space of D_k and v_k is an arbitrary vector. Going through the similar steps as above leads to the same updates for $s_k^\pm$.

Starting from Lecun's Lagrangian If we start from the more natural Lagrangian

$$\mathcal{L}(\theta) = \min_s \max_\delta \ell(s_L)$$
$$+ \sum_{k=1}^{L} \delta_k^\top (s_k - f_k(W_{k-1}s_{k-1})), \qquad (25)$$

instead from (16), then the back-propagated signal $s_{k+1}^- - s_{k+1}^+$ cannot be easily moved inside the activation function f_k. The update equations for $s_k^\pm$ are now of the less-convenient form

$$s_k^+ \leftarrow f_k(W_{k-1}\bar{s}_{k-1}) + \bar{\alpha}W_k^\top f_{k+1}'(\bar{s}_k)(s_{k+1}^- - s_{k+1}^+)$$
$$s_k^- \leftarrow f_k(W_{k-1}\bar{s}_{k-1}) - \alpha W_k^\top f_{k+1}'(\bar{s}_k)(s_{k+1}^- - s_{k+1}^+),$$

which still require derivatives of the (next-layer) activation functions, which makes it less appealing to serve as an NGRAD method.

6. Numerical Validation

Our numerical experiments aim to verify two claims: first, in Section 6.1 we demonstrate that choosing different values for α makes a significant difference in the learned weight matrices. Second, Section 6.2 validates that the proposed DP$^\top$ method is efficient enough to enable successful training beyond toy-sized DNNs for arbitrary choices of α [2].

---

[1] This partial exchange of roles of α and $\bar{\alpha}$ is somewhat analogous to the observation that e.g. forward differences in the primal domain become backward differences in the adjoint.

[2] The code used in our experiments is available at: github.com/Rasmuskh/dualprop_icml_2024

Table 1. Test accuracy and Lipschitz properties of an MLP trained with DP and DP$^\top$ for different choice of α (and fixed $\beta = 1/2$). $\hat{L}$ is the spectral norm of the product of weight matrices. Results are averaged over 3 random seeds. The additional results under DP and BP correspond to adding input layer robustness (DP) and FGSM-based adversarial training (BP), respectively.

		DP			DP$^\top$			BP
	α	1	½	0	1	½	0	
MNIST	Acc	95.85 ± 0.68	98.09 ± 0.06	98.46 ± 0.07 98.43 ± 0.11	98.06 ± 0.15	97.9 ± 0.1	98.21 ± 0.13	98.05 ± 0.12 98.43 ± 0.04
	$\hat{L}$	522 ± 288	67.2 ± 12.6	29.9 ± 4.2 15.87 ± 6.34	73.4 ± 2.7	78.1 ± 9.9	43.2 ± 19.4	55.1 ± 14.4 36.16 ± 4.23
F-MNIST	Acc	83.36 ± 0.74	88.32 ± 0.05	89.16 ± 0.43 88.29 ± 0.79	88.26 ± 0.46	88.35 ± 0.37	88.18 ± 0.27	88.37 ± 0.56 88.64 ± 0.25
	$\hat{L}$	157.7 ± 57.8	34.3 ± 4.0	19.97 ± 3.9 8.05 ± 1.03	52.0 ± 23.6	35.1 ± 7.0	29.4 ± 2.7	37.7 ± 10.2 12.68 ± 0.81

Implementation The adjoint variant DP$^\top$ literally implements the updates stated in Eq. 19. In order to enable DP for values $\alpha \neq 1/2$, we leverage stabilized fixed-point updates (Section F), Not using stabilized updates yields failures to learn successfully as indicated in Table 2 (with details given in Section A.1). Using stable fixed-point updates comes at a significant computational cost as more iterations are required to obtain sufficiently accurate neural states. Consequently we apply these only in the experiments related to the Lipschitz behavior described in Section 6.1.

6.1. The impact of α on the Lipschitz continuity

Section 4.3 suggests that the choice of α matters in particular in the strong feedback setting (i.e. β is not close to zero). We ran experiments on MNIST and FashionMNIST using a 784-512($\times$ 2)-10 MLP with ReLU activation functions, and Table 1 reports test accuracy and Lipschitz estimates (computed as spectral norm of the accumulated weight matrix $W_2W_1W_0$ as ReLU is 1-Lipschitz) after 20 epochs of ADAM-based training (with learning rate 0.001 and default parameters otherwise). No additional loss terms such as weight decay were used. As indicated by Prop. 4.1, a lower value of α yields a significantly smaller Lipschitz constant (as well as improved test accuracy) for the DP method. The DP$^\top$ approach leads to a similar but far less pronounced behavior regarding the Lipschitz estimate. The small differences between the Lipschitz estimates for the different choices of α have no consistent impact on the test accuracy.

As pointed out in Section 4.3, the DP objective $\mathcal{L}_{\alpha=0}^{DP}$ is robust only w.r.t. perturbations in the internal layers. Consequently, the very first layer is not necessarily adversarially robust, and additional robustness can be achieved by extending $\mathcal{L}_0^{DP}$ with a term for the input layer, introducing another term $E_0(s, x) = \|s - x\|^2/2$. Table 1 also includes the resulting accuracy and Lipschitz estimates for this extended variant of $\mathcal{L}_0^{DP}$ as well as the results for simple adversarial training (using FGSM (Goodfellow et al., 2015) as approximate inner maximization; the step size $\varepsilon = 0.01$ chosen to roughly match the test accuracy of DP). Adding

Table 2. Mean test accuracy in percent for the original and the improved dual propagation methods using $\alpha \in \{0, 1\}$. (X) indicates that the particular experiment did not converge. Results are averaged over five random seeds.

	$\alpha = 0$		$\alpha = 1$	
Iters	DP$^\top$	DP	DP$^\top$	DP
1	98.3 ± 0.1	98.50 ± 0.1	98.4 ± 0.1	97.9 ± 0.1
30	98.4 ± 0.0	X	98.5 ± 0.1	X

input layer robustness significantly reduces the Lipschitz estimate of the DP-trained network (at a small reduction of test accuracy). In the setting $\alpha = 1/2$, DP and DP$^\top$ use identical update rules for the hidden units, but treat the output units differently (Eq. 3 vs. Eq. 22), hence the two methods behave very similar but not identical in practice. Finally, we remark that the difference between all tested methods largely vanishes in the weak feedback setting as they all mimic back-propagation when $\beta \to 0^+$.

6.2. VGG16 experiments

We also train a 16 layer VGG network using DP$^\top$ with a crossentropy classification loss on the CIFAR10 and CIFAR100 datasets. In contrast to (Høier et al., 2023) we do not employ the efficient, backpropagation-like forward-backward traversal of the layers. Instead we apply 17 forward-only passes through the layers, which—due to the dyadic neurons—allows errors to gradually propagate to all layers. We employ standard data augmentation (random crops and horizontal flips) and carry out all experiments with 3 random seeds and report mean and std. deviation. 10% of the training data is hold out as validation set for model selection. The hyperparameters are listed in Section B.

Unlike the MLP setting, we find now that the choice of β has some influence on the stability of the training method. Note that (Høier et al., 2023) actually reported results for $\beta = 1/$batch-size (instead of $\beta = 1$) as the (linearized) *mean* CE loss was used. We instead use the summed CE loss (so the feedback signal is not scaled by the batchsize).

Table 3. Table of top-1 and top-5 accuracies in percent for VGG16 networks trained with $DP^\top$ for different choices of β and α. (†) Indicates an experiment where a reasonable model was found before training eventually diverged. (X) indicates an experiment where no reasonable model was learned.

	β	1.0			0.1			0.01			BP
	α	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	
CIFAR10	Top-1	91.34±0.16	X	X	92.41±0.07	†89.3±0.17	X	92.26±0.22	92.19±0.32	92.19±0.08	92.36±0.16
CIFAR100	Top-1	69.3±0.08	X	X	70.05±0.23	69.25±0.28	X	69.33±0.24	69.38±0.10	68.83±0.06	69.1±0.1
	Top-5	89.6±0.11	X	X	89.42±0.16	88.54±0.18	X	88.46±0.17	88.23±0.18	88.28±0.13	88.24±0.22

(a) $\alpha = 0.0$, $\beta = 0.01$

(b) $\alpha = 0.0$, $\beta = 0.01$

(c) $\alpha = 0.0$, $\beta = 1.0$
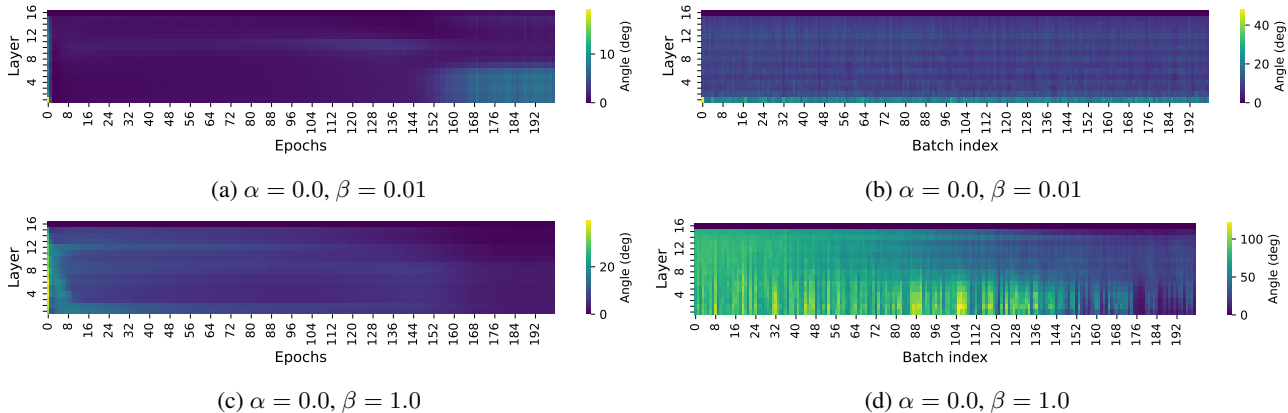
(d) $\alpha = 0.0$, $\beta = 1.0$

Figure 2. CIFAR100: Angle of gradient estimates relative to BP gradients, for $DP^\top$ with $\alpha = 0$ and $\beta = 0.01$ and $\beta = 1.0$. Angles are plotted across layers and epochs (**left column**). The **right column** zooms in on the first 200 minibatches (i.e. a fifth of an epoch).

Results for $DP^\top$ and $\beta \in \{0.01, 0.1, 1.0\}$ in Table 3 show that the choice of α has no noticeable impact in the weak feedback setting ($\beta = 0.01$), which is to be expected. In the moderate feedback setting ($\beta = 0.1$), $DP^\top$ with $\alpha = 1$ fails, and $DP^\top$ with $\alpha = 0.5$ eventually diverges on CIFAR10 after reaching an acceptable model. The choice $\beta = 1.0$ requires $\alpha = 0$ for successful training and leads to a drop in performance. Fig. 2 compares the angle between $DP^\top$ gradient estimates and backprop gradient estimates when $\alpha = 0$ in the strong ($\beta = 1$) and weak ($\beta = 0.01$) feedback setting. It can be seen in Figs. 2a and 2c, that $DP^\top$ with $\beta = 0.01$ always provides excellent gradient estimates, whereas the gradients are initially not well aligned when $\beta = 1$. The difference is especially pronounced during the first 200 batches (a fifth of an epoch) in Figs. 2b and 2d.

Imagenet32x32 We restrict the more compute intensive ImageNet32x32 experiments to the setting $\alpha = 1/2$ and $\beta = 0.01$. We use 5% of the training data for validation and model selection, and use the public validation dataset to evaluate the selected model. With top-1 accuracy of 41.59 ± 0.11 and top-5 accuracy of 65.63 ± 0.19 it is apparent that the repeated inference scheme maintains performance on par with (or slightly above) the results reported for DP and BP in (Høier et al., 2023).

7. Discussion

Biological plausibility of the dyadic neuron model In the DP framework neurons have two internal compartments, which integrate both bottom-up and top-down input signals. Introducing distinct neural compartments is somewhat biologically plausible as recent neuroscience research suggests that dendrites are compartmental structures capable of performing fairly complex operations (Chavlis & Poirazi, 2021; Beniaguev et al., 2021). It has been proposed in (Guerguiev et al., 2017; Richards & Lillicrap, 2019) that distal apical dendrites (integrating top-down feedback signals) might be responsible for credit assignment in pyramidal neurons, whereas basal dendrites integrate local feed-forward and recurrent inputs. The DP algorithm does not fully fit this picture as the state updates require integrating bottom-up input and feed-back input jointly. Further, DP relies on some form of multiplexing within a neuron in order to communicate both the neural state difference "downstream" and the mean state "upstream." This distinction is illustrated in Fig. 1. In spite of these caveats, the dyadic neuron model shows that—by taking a small step away from the single compartment based McCulloch Pitts neuron model—one can nevertheless achieve asynchronous propagation of meaningful errors entirely driven by local neural activities. While still far from biological reality this may be useful for goals such as on-chip training of neuromorphic hardware.

Compartmental interpretation of related models The contrastive, lifted and difference target propagation based models can all be interpreted as compartmental models (either compartments within a neuron or within a small neural circuitry). In EP and CHL, neurons need to store the activities of the two temporally distinct inference phases. In dual propagation and in LPOM, neurons are required to maintain the nudged internal states. Neurons in difference target propagation are expected to have compartments for storing predictions, targets and possibly even noise perturbed states (Lee et al., 2015). Works such as (Guerguiev et al., 2017; Sacramento et al., 2018) explicitly focus on building biologically inspired compartmental neuron models, although these methods incur even higher computational costs by also modelling spiking behaviour. The segregated dendrite model (Guerguiev et al., 2017) requires two temporally distinct phases making it closer to a neuroscientists' realization of CHL or EP. The dendritic cortical microcircuit model (Sacramento et al., 2018) is single phased and implementation-wise similar to fully asymmetric DP.

Timescales of neural dynamics When training a network with BP, the weights of a layer can not be updated until the forward pass has completed, and the error has at least been back-propagated to the layer in question. This strict synchronization requirement has motivated several algorithms (Jaderberg et al., 2017; Nøkland & Eidnes, 2019), which—to varying degree—relax the need for precise synchronization. One way to avoid synchronisation across layers entirely is to use auxiliary losses at each layer, which often is applied in unsupervised learning methods such as ART (Grossberg, 1987), deep belief networks (Hinton et al., 2006; Bengio et al., 2006) or—more recently—soft-Hebbian learning (Journé et al., 2023). In the supervised learning setting, many lifted neural network approaches (such as (Carreira-Perpinan & Wang, 2014; Whittington & Bogacz, 2017; Li et al., 2020), that result in a joint minimization problem over neural states and synaptic weights) in principle rely on minimal synchronization (though in practice, inference of the neural activities is usually run to convergence to avoid storing intermediate neural states).

Among biologically inspired alternatives to BP, difference target propagation (Lee et al., 2015) has similar synchronization constraints as BP, while CHL and EP require temporally distinct inference phases to converge before any weight update. DP only has a single inference phase and does not impose a strict schedule on the order in which layers are traversed, but still requires that meaningful error signals are able to reach layers before their weights are updated (see e.g. the "random-DP" variant in (Høier et al., 2023)). In a setting with asynchronous neurons this corresponds to a requirement that synaptic plasticity takes place at a (much) slower timescale than neural activities (which is at the core

of continuous-time neuron models used in computational neuroscience). This agrees with the observation that neural activity, short-term plasticity (e.g. working memory and attention) and long-term plasticity take place at entirely different timescales (10ms, 100ms-minutes and minutes-hours respectively, e.g. (Tsodyks et al., 1998; Abbott & Regehr, 2004; Citri & Malenka, 2008)).

Weight transport One of the key objections to BP is that it is biologically implausible for the forwards and backwards pathways to be symmetric (using W to propagate forwards and $W^\top$ to propagate errors backwards). This is known as the weight transport problem (Crick, 1989; Grossberg, 1987), and variations of this issue is present in many NGRAD algorithms, including DP. The weight transport problem is not addressed in the current work, but we note that DP has previously been shown to be compatible with Kolen-Pollack learning (Kolen & Pollack, 1994) of the feedback weights (Høier et al., 2023). There are a number of other interesting attempts at solving the weight transport problem in the literature such as random feedback alignment (Lillicrap et al., 2014; Nøkland, 2016; Frenkel et al., 2021) and stochastic alignment methods (Akrout et al., 2019; Ernoult et al., 2022).

8. Conclusion

Fully local activity difference based learning algorithms are essential for achieving on-device training of neuromorphic hardware. However, the majority of works require distinct inference phases and are costly to simulate. The dual propagation formulation overcomes these issues but also relies on symmetric nudging (at each layer), which may itself be too strict a requirement in noisy hardware. Furthermore, a rigorous derivation of the DP objective has so far been missing. In this work we first present the theoretical foundation for dual propagation and argue (in theory and practice) why asymmetric nudging can be beneficial, despite requiring slow inference methods (instead of fast fixed-point dynamics) on digital hardware. Further, we present an improved variant of dual propagation derived from a suitable Lagrangian, which recovers the dynamics of dual propagation in the case of symmetric nudging ($\alpha = 1/2$). In the case of asymmetric nudging (such as $\alpha \in \{0, 1\}$) the new variant leads to slightly modified fixed-point dynamics, which are in fact significantly more stable as demonstrated in our experiments.

tre at Linköping University and the Knut and Alice Wallenberg foundation. We would also like to thank Maxence Ernoult for helpful discussions.

Impact statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abbott, L. and Regehr, W. G. Synaptic computation. *Nature*, 431(7010):796–803, 2004.

Akrout, M., Wilson, C., Humphreys, P., Lillicrap, T., and Tweed, D. B. Deep learning without weight transport. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Askari, A., Negiar, G., Sambharya, R., and Ghaoui, L. E. Lifted neural networks. *arXiv preprint arXiv:1805.01532*, 2018.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 2006.

Beniaguev, D., Segev, I., and London, M. Single cortical neurons as deep artificial neural networks. *Neuron*, 109 (17):2727–2739, 2021.

Carreira-Perpinan, M. and Wang, W. Distributed optimization of deeply nested systems. In *Artificial Intelligence and Statistics*, pp. 10–19, 2014.

Chavlis, S. and Poirazi, P. Drawing inspiration from biological dendrites to empower artificial neural networks. *Current opinion in neurobiology*, 70:1–10, 2021.

Choromanska, A., Cowen, B., Kumaravel, S., Luss, R., Rigotti, M., Rish, I., Diachille, P., Gurev, V., Kingsbury, B., Tejwani, R., et al. Beyond backprop: Online alternating minimization with auxiliary variables. In *International Conference on Machine Learning*, pp. 1193–1202. PMLR, 2019.

Citri, A. and Malenka, R. C. Synaptic plasticity: multiple forms, functions, and mechanisms. *Neuropsychopharmacology*, 33(1):18–41, 2008.

Crick, F. The recent excitement about neural networks. *Nature*, 337:129–132, 1989.

Dempe, S. and Zemkoho, A. B. The bilevel programming problem: reformulations, constraint qualifications and optimality conditions. *Mathematical Programming*, 138 (1):447–473, 2013.

Ernoult, M. M., Normandin, F., Moudgil, A., Spinney, S., Belilovsky, E., Rish, I., Richards, B., and Bengio, Y. Towards scaling difference target propagation by learning backprop targets. In *International Conference on Machine Learning*, pp. 5968–5987. PMLR, 2022.

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

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Gould, S., Hartley, R., and Campbell, D. Deep declarative networks: A new hope. *arXiv preprint arXiv:1909.04866*, 2019.

Grossberg, S. Competitive learning: From interactive activation to adaptive resonance. *Cognitive science*, 11(1): 23–63, 1987.

Gu, F., Askari, A., and El Ghaoui, L. Fenchel lifted networks: A lagrange relaxation of neural network training. In *International Conference on Artificial Intelligence and Statistics*, pp. 3362–3371. PMLR, 2020.

Guerguiev, J., Lillicrap, T. P., and Richards, B. A. Towards deep learning with segregated dendrites. *eLife*, 6:e22901, dec 2017. ISSN 2050-084X. doi: 10.7554/eLife.22901.

Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, 18 (7):1527–1554, 2006.

Høier, R. and Zach, C. Lifted regression/reconstruction networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2020.

Høier, R., Staudt, D., and Zach, C. Dual propagation: Accelerating contrastive hebbian learning with dyadic neurons. In *International Conference on Machine Learning*, 2023.

Jaderberg, M., Czarnecki, W. M., Osindero, S., Vinyals, O., Graves, A., Silver, D., and Kavukcuoglu, K. Decoupled neural interfaces using synthetic gradients. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1627–1635. JMLR. org, 2017.

Journé, A., Rodriguez, H. G., Guo, Q., and Moraitis, T. Hebbian deep learning without feedback. In *The Eleventh International Conference on Learning Representations*, 2023.

Kendall, J. D. and Kumar, S. The building blocks of a brain-inspired computer. *Applied Physics Reviews*, 7(1): 011305, 2020.

Kolen, J. and Pollack, J. Backpropagation without weight transport. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 3, pp. 1375–1380 vol.3, 1994. doi: 10.1109/ICNN.1994. 374486.

Laborieux, A. and Zenke, F. Holomorphic equilibrium propagation computes exact gradients through finite size oscillations. *Advances in Neural Information Processing Systems*, 35:12950–12963, 2022.

Laborieux, A. and Zenke, F. Improving equilibrium propagation without weight symmetry through jacobian homeostasis. *arXiv preprint arXiv:2309.02214*, 2023.

Laborieux, A., Ernoult, M., Scellier, B., Bengio, Y., Grollier, J., and Querlioz, D. Scaling equilibrium propagation to deep convnets by drastically reducing its gradient estimator bias. *Frontiers in neuroscience*, 15:129, 2021.

Lecun, Y. A theoretical framework for back-propagation. In *Proceedings of the 1988 Connectionist Models Summer School, CMU, Pittsburg, PA*, pp. 21–28. Morgan Kaufmann, 1988.

Lee, D.-H., Zhang, S., Fischer, A., and Bengio, Y. Difference target propagation. In *Joint european conference on machine learning and knowledge discovery in databases*, pp. 498–515. Springer, 2015.

Li, J., Xiao, M., Fang, C., Dai, Y., Xu, C., and Lin, Z. Training neural networks by lifted proximal operator machines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. Random feedback weights support learning in deep neural networks. *arXiv preprint arXiv:1411.0247*, 2014.

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., and Hinton, G. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

Movellan, J. R. Contrastive hebbian learning in the continuous hopfield model. In *Connectionist models*, pp. 10–17. Elsevier, 1991.

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

Nøkland, A. and Eidnes, L. H. Training neural networks with local error signals. In *International conference on machine learning*, pp. 4839–4850. PMLR, 2019.

Outrata, J. V. A note on the usage of nondifferentiable exact penalties in some special optimization problems. *Kybernetika*, 24(4):251–258, 1988.

Richards, B. A. and Lillicrap, T. P. Dendritic solutions to the credit assignment problem. *Current opinion in neurobiology*, 54:28–36, 2019.

Sacramento, J., Ponte Costa, R., Bengio, Y., and Senn, W. Dendritic cortical microcircuits approximate the backpropagation algorithm. *Advances in neural information processing systems*, 31, 2018.

Salvatori, T., Mali, A., Buckley, C. L., Lukasiewicz, T., Rao, R. P., Friston, K., and Ororbia, A. Brain-inspired computational intelligence via predictive coding. *arXiv preprint arXiv:2308.07870*, 2023.

Scellier, B. and Bengio, Y. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017.

Scellier, B., Ernoult, M., Kendall, J., and Kumar, S. Energybased learning algorithms for analog computing: a comparative study. *arXiv preprint arXiv:2312.15103*, 2023.

Stern, M., Hexner, D., Rocks, J. W., and Liu, A. J. Supervised learning in physical networks: From machine learning to learning machines. *Physical Review X*, 11(2): 021045, 2021.

Tsodyks, M., Pawelzik, K., and Markram, H. Neural networks with dynamic synapses. *Neural computation*, 10 (4):821–835, 1998.

Whittington, J. C. and 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.

Xie, X. and Seung, H. S. Equivalence of backpropagation and contrastive hebbian learning in a layered network. *Neural computation*, 15(2):441–454, 2003.

Yi, S.-i., Kendall, J. D., Williams, R. S., and Kumar, S. Activity-difference training of deep neural networks using memristor crossbars. *Nature Electronics*, pp. 1–7, 2022.

Zach, C. Bilevel programs meet deep learning: A unifying view on inference learning methods. *arXiv preprint arXiv:2105.07231*, 2021.

Zach, C. and Estellers, V. Contrastive learning for lifted networks. In Sidorov, K. and Hicks, Y. (eds.), *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 163.1–163.12. BMVA Press, 9 2019. doi: 10.5244/C.33. 163.

A. Additional Results

Fig. 3 illustrate additional statistics on the alignment of back-propagation induced gradients (w.r.t. the network parameters) and the ones obtained by the proposed DP improvement.

A.1. Divergence of DP

The original dual propagation and the improved variant differ in the case $\alpha \in [0, 1]$, $\alpha \neq 1/2$, with the boundary cases $\alpha \in \{0, 1\}$ being of particular interest. The most efficient way to implement dual propagation is by sequentially updating neurons in every layer from input to output and then from output to input (akin to a forward and a backward traversal, which are nevertheless part of the sole inference phase using the same dynamics) before performing a weight update. However, to show the unstable behaviour of the original dual propagation formulation in the case of asymmetric nudging we instead perform repeated inference passes through the layers (which also better matches a continuous/asynchronous compute model). Each pass (or iteration) corresponds to updating all neurons from input to output layer and back.

The results of these experiments are summarized in Tab 2, where we trained a 784-1000(×4)-10 MLP with ReLU activation functions on MNIST using the least-squares loss. The nudging strength β is 1/2, which is also compatible with the original DP method. We observe that inference in the original variant of DP diverges when applying asymmetric nudging and multiple inference iterations. This is not surprising as inference for dual propagation is only guaranteed to converge in the symmetric setting. The new variant of DP on the other hand yields stable inference results in all cases.

B. CNN experimental details

The experiments of Section 6.2 were carried out with a VGG16 network and the following hyper parameters:

Table 4. Hyper parameters

	Epochs			Learning rate			Momentum	Weight-decay	Batchsize
Dataset	Warmup	Decay	Total	Initial	Peak	Final			
CIFAR10	10	120	130	0.001	0.025	2e-6	0.9	5e-4	100
CIFAR100	10	190	200	0.001	0.015	2e-6	0.9	5e-4	50
Imagenet32x32	10	120	130	0.001	0.020	2e-6	0.9	5e-4	250

C. Proof of Proposition 5.1

Proof. The first-order optimality conditions for s_k and δ_k ($k = 1, \ldots, L - 1$) in Eq. 16 are given by

$$\text{(I)} \quad 0 = (f_k^{-1})'(s_k)\delta_k - W_k^\top \delta_{k+1} \qquad\qquad \text{(II)} \quad 0 = f_k^{-1}(s_k) - W_{k-1}s_{k-1}. \tag{26}$$

Reparametrization in terms of $s_k^\pm$ (Eq. 17) and expanding (II) + α(I) and (II) − $\bar{\alpha}$(I) yields

$$\begin{aligned}
0 = & \ f_k^{-1}(\alpha s_k^+ + \bar{\alpha} s_k^-) - a_k + \alpha(f_k^{-1})'(\bar{s}_k)^\top(s_k^+ - s_k^-) - \alpha W_k^\top(s_{k+1}^+ - s_{k+1}^-) \\
0 = & -f_k^{-1}(\alpha s_k^+ + \bar{\alpha} s_k^-) + a_k + \bar{\alpha}(f_k^{-1})'(\bar{s}_k)^\top(s_k^+ - s_k^-) - \bar{\alpha} W_k^\top(s_{k+1}^+ - s_{k+1}^-),
\end{aligned} \tag{27}$$

which are also the KKT conditions for $\mathcal{L}_\alpha^{DP+}$ in Eq. 18. It remains to manipulate these to obtain the desired update equations. Adding the equations above results in

$$0 = (f_k^{-1})'(\bar{s}_k)^\top(s_k^+ - s_k^-) - W_k^\top(s_{k+1}^+ - s_{k+1}^-). \tag{28}$$

Dual propagation is (via its use of finite differences) intrinsically linked to a (locally) linear assumption on f_k. Hence, we assume $f_k(a) = s_k^0 + D_k a + O(\|a - a_k\|^2)$ with $s_k^0 = f_k(a_k) - D_k a_k$ and $D_k = f_k'(a_k)$. The local linear assumption allows us to neglect the higher order terms. Consequently we also assume that f_k^{-1} is locally linear and therefore $(f_k^{-1})'(\bar{s}_k) \approx D_k^{-1}$ is independent of $\bar{s}_k$. Hence, we arrive at
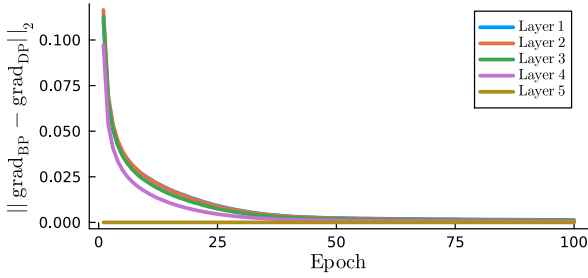
$$0 \approx D_k^{-\top}(s_k^+ - s_k^-) - W_k^\top(s_{k+1}^+ - s_{k+1}^-) \iff s_k^+ - s_k^- \approx D_k^\top W_k^\top(s_{k+1}^+ - s_{k+1}^-). \tag{29}$$
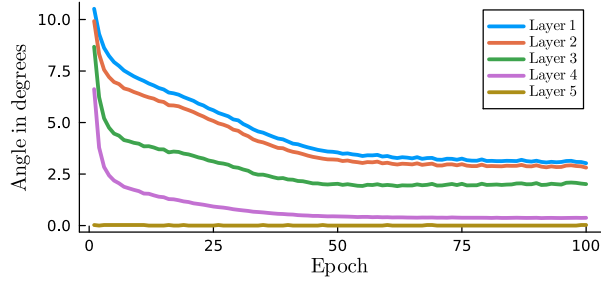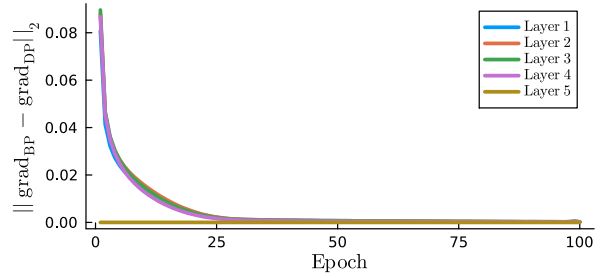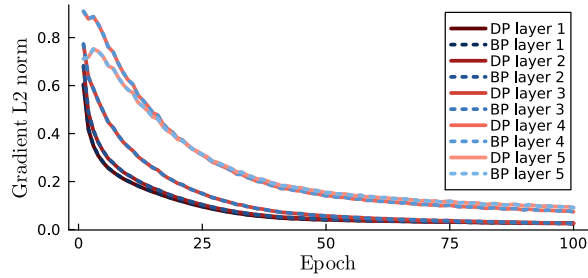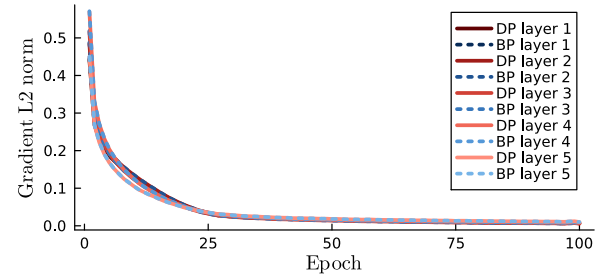
(a) $\alpha = 0$
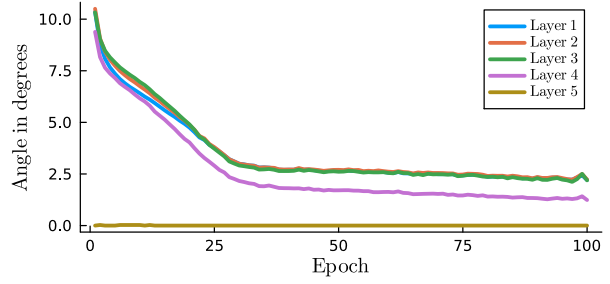
(b) $\alpha = 1$

(c) $\alpha = 0.0$

(d) $\alpha = 1.0$

(e) $\alpha = 0.0$

(f) $\alpha = 1.0$

Figure 3. MNIST experiments employing asymetric α. Results are averaged over five random seeds. **Top:** Alignment between the parameter updates obtained with back-propagation and with the improved DP variant (using 30 inference iterations and asymmetric nudging with $\alpha \in \{0, 1\}$). **Middle:** L2 norm of difference between BP and DP gradients. **Bottom:** L2 norms of BP and DP gradients when using 30 inference iterations and asymmetric nudging ($\alpha = 0.0$ and $\alpha = 1.0$). Results are averaged over five random seeds.

We insert this into the second of the necessary optimality conditions (27) and obtain

$$0 = f_k^{-1}(\alpha s_k^+ + \bar{\alpha} s_k^-) - a_k = f_k^{-1}\big(s_k^- + \alpha(s_k^+ - s_k^-)\big) - a_k$$
$$= f_k^{-1}\big(s_k^- + \alpha D_k^\top W_k^\top(s_{k+1}^+ - s_{k+1}^-)\big) - a_k = D_k^{-1}\big(s_k^- - s_k^0 + \alpha D_k^\top W_k^\top(s_{k+1}^+ - s_{k+1}^-)\big) - a_k.$$

The last line is equivalent to

$$s_k^- = D_k a_k + s_k^0 - \alpha D_k^\top W_k^\top(s_{k+1}^+ - s_{k+1}^-) \approx f_k\big(W_{k-1}\bar{s}_{k-1} - \alpha W_k^\top(s_{k+1}^+ - s_{k+1}^-)\big). \tag{30}$$

Analogously, $s_k^+ \approx f_k(W_{k-1}\bar{s}_{k-1} + \bar{\alpha} W_k^\top(s_{k+1}^+ - s_{k+1}^-))$. In summary we obtain the update rules shown in (19). $\qquad\square$

D. Proof of Proposition 4.1

For convenience we restate Prop. 4.1:

Proposition D.1. *Let $0 < \beta \leq \beta'$ and $\alpha, \alpha' \in [0, 1]$ with $\alpha \leq \alpha'$. Then the following holds: (a) $J_{\alpha,\beta}^{EP}(\theta) \leq J_{\alpha',\beta}^{EP}(\theta)$ and (b) $\beta J_{\alpha,\beta}^{EP}(\theta) \leq \beta' J_{\alpha,\beta'}^{EP}(\theta)$.*

This is a consequence of the following lemma.

Lemma D.2. *Let*

$$s_1 = \arg\min_s \alpha_1 f(z) + g(z) \qquad\qquad s_2 = \arg\min_s \alpha_2 f(z) + g(z) \tag{31}$$

for $\alpha_2 > \alpha_1$. Then $f(s_2) \leq f(s_1)$.

Proof. Optimality of $s_{1,2}$ implies

$$\alpha_1 f(s_1) + g(s_1) \leq \alpha_1 f(s_2) + g(s_2)$$
$$\alpha_2 f(s_2) + g(s_2) \leq \alpha_2 f(s_1) + g(s_1).$$

Adding these inequalities yields

$$\alpha_1 f(s_1) + \alpha_2 f(s_2) \leq \alpha_1 f(s_2) + \alpha_2 f(s_1) \iff (\alpha_2 - \alpha_1)f(s_2) \leq (\alpha_2 - \alpha_1)f(s_1), \tag{32}$$

i.e. $f(s_2) \leq f(s_1)$ since $\alpha_2 - \alpha_1 > 0$. $\qquad\square$

Proof of the proposition. **Claim (a)** In the following we absorb $1/\beta$ into E and omit the dependency of E on θ for notational brevity. $\alpha \leq \alpha'$ implies $\bar{\alpha} \leq \bar{\alpha}' \leq 0 \leq \alpha \leq \alpha'$. Further, we define

$$s_\alpha := \arg\min_s \alpha\ell(s) + E(u) \qquad s_{-\bar{\alpha}} := \arg\min_s -\bar{\alpha}\ell(s) + E(s)$$
$$s_{\alpha'} := \arg\min_s \alpha'\ell(s) + E(u) \qquad s_{-\bar{\alpha}'} := \arg\min_s -\bar{\alpha}'\ell(s) + E(s). \tag{33}$$

We identify $f(s) = \ell(s)$ and $g(s) = E(s, \theta)$ and apply the lemma to deduce that

$$\ell(s_{\alpha'}) \leq \ell(s_\alpha) \leq \ell(s_{-\bar{\alpha}'}) \leq \ell(s_{-\bar{\alpha}}). \tag{34}$$

Now

$$\begin{aligned}
J_{\alpha,\beta}^{\mathrm{EP}}(\theta) &= \alpha\ell(s_\alpha) + E(s_\alpha) + \bar{\alpha}\ell(s_{-\bar{\alpha}}) - E(s_{-\bar{\alpha}}) \\
&\leq \alpha\ell(s_{\alpha'}) + E(s_{\alpha'}) + \bar{\alpha}\ell(s_{-\bar{\alpha}}) - E(s_{-\bar{\alpha}}) \\
&= \alpha'\ell(s_{\alpha'}) + E(s_{\alpha'}) + (\alpha - \alpha')\ell(s_{\alpha'}) + \bar{\alpha}\ell(s_{-\bar{\alpha}}) - E(s_{-\bar{\alpha}}) \\
&\leq \alpha'\ell(s_{\alpha'}) + E(s_{\alpha'},) + (\alpha - \alpha')\ell(s_{-\bar{\alpha}}) + \bar{\alpha}\ell(s_{-\bar{\alpha}}) - E(s_{-\bar{\alpha}}) \\
&= \alpha'\ell(s_{\alpha'}) + E(s_{\alpha'}) + \bar{\alpha}'\ell(s_{-\bar{\alpha}}) - E(s_{-\bar{\alpha}}) \\
&\leq \alpha'\ell(s_{\alpha'}) + E(s_{\alpha'}) + \bar{\alpha}'\ell(s_{-\bar{\alpha}'}) - E(s_{-\bar{\alpha}'}) \\
&= J_{\alpha',\beta}^{\mathrm{EP}}(\theta),
\end{aligned} \tag{35}$$

where we used the definition of s_α in the first inequality, $\ell(s_{\alpha'}) \leq \ell(s_{-\bar{\alpha}})$ (together with $\alpha - \alpha' \geq 0$) for the second one, and finally the definition of $s_{-\bar{\alpha}'}$ to obtain the last inequality.

Claim (b) We proceed similar to above and define the non-negative quantities

$$\gamma := \alpha\beta \qquad \bar{\gamma} := \bar{\alpha}\beta \qquad \gamma' := \alpha\beta' \qquad \bar{\gamma}' := \bar{\alpha}\beta'. \tag{36}$$

The assumption $\beta \leq \beta'$ implies

$$-\bar{\gamma}' \leq -\bar{\gamma} \leq 0 \leq \gamma \leq \gamma'. \tag{37}$$

We also introduce

$$
\begin{aligned}
s_\gamma &:= \arg\min_s \gamma\ell(s) + E(s) & s_{-\bar{\gamma}} &:= \arg\min_s -\bar{\gamma}\ell(s) + E(s) \\
s_{\gamma'} &:= \arg\min_s \gamma'\ell(s) + E(s) & s_{-\bar{\gamma}'} &:= \arg\min_s -\bar{\gamma}'\ell(s) + E(s).
\end{aligned}
\tag{38}
$$

From the lemma we conclude that

$$\ell(s_{\gamma'}) \leq \ell(s_\gamma) \leq \ell(s_{-\bar{\gamma}}) \leq \ell(s_{-\bar{\gamma}'}). \tag{39}$$

$$
\begin{aligned}
\beta J^{\text{EP}}_{\alpha,\beta}(\theta) &= \gamma\ell(s_\gamma) + E(s_\gamma) + \bar{\gamma}\ell(s_{-\bar{\gamma}}) - E(s_{-\bar{\gamma}}) \\
&\leq \gamma\ell(s_{\gamma'}) + E(s_{\gamma'}) + \bar{\gamma}\ell(s_{-\bar{\gamma}}) - E(s_{-\bar{\gamma}}) \\
&= \gamma'\ell(s_{\gamma'}) + E(s_{\gamma'}) + (\gamma - \gamma')\ell(s_{\gamma'}) + \bar{\gamma}\ell(s_{-\bar{\gamma}}) - E(s_{-\bar{\gamma}}) \\
&\leq \gamma'\ell(s_{\gamma'}) + E(s_{\gamma'}) + (\gamma - \gamma')\ell(s_{-\bar{\gamma}}) + \bar{\gamma}\ell(s_{-\bar{\gamma}}) - E(s_{-\bar{\gamma}}) \\
&= \gamma'\ell(s_{\gamma'}) + E(s_{\gamma'}) + \underbrace{(\gamma - \gamma' + \bar{\gamma})}_{=\bar{\gamma}'}\ell(s_{-\bar{\gamma}}) - E(s_{-\bar{\gamma}}) \\
&\leq \gamma'\ell(s_{\gamma'}) + E(s_{\gamma'}) + \bar{\gamma}'\ell(s_{-\bar{\gamma}'}) - E(s_{-\bar{\gamma}'}) \\
&= \beta' J^{\text{EP}}_{\alpha,\beta'}(\theta).
\end{aligned}
\tag{40}
$$

We applied the definition of s_γ (first inequality), $\ell(s_{\gamma'}) \leq \ell(s_{-\bar{\gamma}})$ (2nd inequality) and the definition of $s_{\bar{\gamma}'}$. □

E. Proof of Proposition 4.2

For convenience we restate Prop. 4.2:

Proposition E.1. *Let $0 < \beta' < \beta$ and α' such that $\bar{\alpha}\beta = \bar{\alpha}'\beta'$. Then $J^{EP}_{\alpha,\beta}(\theta) \leq J^{EP}_{\alpha',\beta'}(\theta)$.*

Proof. Let $\gamma := \bar{\alpha}\beta = \bar{\alpha}'\beta'$. We further introduce

$$\tilde{U}_{-\gamma}(s) := -\gamma\ell(s) + U(s) \tag{41}$$

and

$$
\begin{aligned}
s_{\alpha\beta} &= \arg\min_s \alpha\ell(s) + \tfrac{1}{\beta}U(s) \\
s_{\alpha'\beta'} &= \arg\min_s \alpha'\ell(s) + \tfrac{1}{\beta'}U(s) \\
s_{-\gamma} &= \arg\min_s -\bar{\alpha}\ell(s) + \tfrac{1}{\beta}U(s) = \arg\min_s \tilde{U}_{-\gamma}(s).
\end{aligned}
\tag{42}
$$

Now

$$
\begin{aligned}
J_{\alpha,\beta}^{\text{EP}}(\theta) &= \alpha\ell(s_{\alpha\beta}) + \tfrac{1}{\beta}U(s_{\alpha\beta}) + \bar{\alpha}\ell(s_{-\gamma}) - \tfrac{1}{\beta}U(s_{-\gamma}) \\
&= \ell(s_{\alpha\beta}) + \tfrac{1}{\beta}\tilde{U}_{-\gamma}(s_{\alpha\beta}) - \tfrac{1}{\beta}\tilde{U}_{-\gamma}(s_{-\gamma}) \\
&\le \ell(s_{\alpha'\beta'}) + \tfrac{1}{\beta}\tilde{U}_{-\gamma}(s_{\alpha'\beta'}) - \tfrac{1}{\beta}\tilde{U}_{-\gamma}(s_{-\gamma}) \\
&= \ell(s_{\alpha'\beta'}) + \tfrac{1}{\beta}\underbrace{\left(\tilde{U}_{-\gamma}(s_{\alpha'\beta'}) - \tilde{U}_{-\gamma}(s_{-\gamma})\right)}_{\ge 0} \\
&\le \ell(s_{\alpha'\beta'}) + \tfrac{1}{\beta'}\left(\tilde{U}_{-\gamma}(s_{\alpha'\beta'}) - \tilde{U}_{-\gamma}(s_{-\gamma})\right) \\
&= \alpha'\ell(s_{\alpha'\beta'}) + \tfrac{1}{\beta'}U(s_{\alpha'\beta'}) + \bar{\alpha}'\ell(s_{-\gamma}) - \tfrac{1}{\beta'}U(s_{-\gamma}) = J_{\alpha',\beta'}^{\text{EP}}(\theta),
\end{aligned}
\tag{43}
$$

where we used the minimality of $s_{\alpha\beta}$ and $s_{-\gamma}$, and that $1/\beta < 1/\beta'$ to obtain the inequalities. $\qquad\square$

F. Stabilized Fixed-Point Iterations

The LPOM model considered in (Li et al., 2020) uses specific layerwise potentials E_k and also layer-dependent values of β,

$$
\mathcal{L}^{\text{LPOM}}(\theta) = \min_{\{s_k\}} \ell(s_L) + \sum_k \tfrac{1}{\beta_k}\left(G_k(s_k) - s_k^\top W_{k-1}s_{k-1} + G_k^*(W_{k-1}s_{k-1})\right),
\tag{44}
$$

where we use s_k instead of s_k^+ as the s_k^- are already eliminated. The LPOM authors suggest fixed-point iterations of the form

$$
s_k^{(t)} \leftarrow f_k\left(W_{k-1}s_{k-1} + \tfrac{\beta_k}{\beta_{k+1}}W_k^\top\left(s_{k+1} - f_{k+1}(W_k s_k^{(t-1)})\right)\right),
\tag{45}
$$

which can be derived from the stationarity condition w.r.t. s_k,

$$
\begin{aligned}
0 &= \tfrac{1}{\beta_k}\left(\nabla G_k(s_k) - W_{k-1}s_{k-1}\right) + \tfrac{1}{\beta_{k+1}}\left(W_k^\top \nabla G_{k+1}^*(W_k s_k) - W_k^\top s_{k+1}\right) \\
&= \tfrac{1}{\beta_k}\left(f_k^{-1}(s_k) - W_{k-1}s_{k-1}\right) + \tfrac{1}{\beta_{k+1}}W_k^\top\left(f_{k+1}(W_k s_k) - s_{k+1}\right).
\end{aligned}
\tag{46}
$$

In (Li et al., 2020) it is assumed that each f_k is Lipschitz-continuous (w.l.o.g. 1-Lipschitz continuous) and that $\{\beta_k\}_k$ are chosen such that

$$
\tfrac{\beta_k}{\beta_{k+1}}\|W_k^\top W_K\|_2 < 1,
\tag{47}
$$

i.e. $\beta_k \ll \beta_{k+1}$ and thereby introducing discounting of later layers (and enabling only weak feedback from the target loss ℓ). In our setting we choose $\beta_0 = \cdots = \beta_{L-1} = 1$ and $\beta_L = \beta$. Hence, the scheme in Eq. 45 is only guaranteed to converge as long as $\|W_k^\top W_k\|_2 < 1$. For many standard activation functions (in particular ReLU, leaky ReLU and hard sigmoid) the underlying G_k is of the form $G_k(s_k) = \|s_k\|^2/2 + \iota_C(s_k)$ for a convex set C, where $\iota_C(s_k)$ is the functional form of the constraint $s_k \in C$. Therefore it is convenient to add a quadratic dampening term when optimizing w.r.t. s_k. Let

$$
\Phi_k(s_k; s_{k-1}, s_{k+1}, \theta) := G_k(s_k) - s_k^\top W_{k-1}s_{k-1} + G_{k+1}^*(W_k s_k) - s_{k+1}^\top W_k s_k
\tag{48}
$$

$$
\tilde{\Phi}_k(s_k; s_k^0, s_{k-1}, s_{k+1}, \theta) := G_k(s_k) - s_k^\top W_{k-1}s_{k-1} + (s_k - s_k^0)^\top W_k^\top f_{k+1}(W_k s_k^0) - s_{k+1}^\top W_k s_k
\tag{49}
$$

be the terms in $\mathcal{L}^{\text{LPOM}}$ dependent on s_k and its (partially) linearized surrogate, respectively. Now consider the iteration

$$
\begin{aligned}
s_k^{(t+1)} &:= \arg\min_{s_k} \tilde{\Phi}_k(s_k; s_k^{(t)}, s_{k-1}, s_{k+1}, \theta) + \tfrac{L}{2}\|s_k - s_k^0\|^2 \\
&= \Pi_C\left(\frac{W_{k-1}s_{k-1} + W_k^\top\left(s_{k+1} - f_{k+1}(W_k s_k^{(t-1)})\right) + L s_k^{(t)}}{1 + L}\right).
\end{aligned}
\tag{50}
$$

If $s_k^{(t)}$ is a stationary point of $\tilde{\Phi}_k$, then $s_k^{(t+1)} = s_k^{(t)}$ is also a stationary point of Φ_k. If G_{k+1} is 1-strongly convex (and therefore $\nabla G_{k+1} = f_{k+1}$ is 1-Lipschitz continuous) and $L \ge \|W_k^\top W_k\|_2$, then $\tilde{\Phi}_k(\cdot; s_k^0, s_{k-1}, s_{k+1}, \theta) + \tfrac{L}{2}\|\cdot - s_k^0\|^2$ is a

majorizer of Φ_k, and the sequence $(s_k^{(t)})_t$ decreases monotonically Φ_k. It can be further shown that Eq. 50 is a contraction if $L > \max(0, \|W^\top W\|_2 - 1)/2$.

We use the stabilized fixed-point scheme in Eq. 50 determine the neural states s_k in the setting $\alpha \in \{0, 1\}$ (i.e. the LPOM and inverted LPOM formulations), where we assign L to the estimate of $\|W_k^\top W_k\|_2$ obtained by 5 power iterations.

G. Analysis of the fixed point iterations

In the following we use an idealized setup, where we make the following assumptions:

1. We consider a trilevel program with a hidden and an output layer,

2. and we use second-order Taylor expansion for the relevant mappings.

The trilevel program is given by

$$\ell(y^*) \qquad \text{s.t. } y^* = \arg\min_y F(y) - y^\top W z^* \qquad z^* = \arg\min_z G(z) - z^\top x \tag{51}$$

and ℓ, F and G are given by

$$\ell(y) = \tfrac{1}{2} y^\top H_\ell y + b_\ell^\top y \qquad\qquad F(y) = \tfrac{1}{2} y^\top H_F y + b_F^\top y \qquad\qquad G(z) = \tfrac{1}{2} z^\top H_G z + b_G^\top z. \tag{52}$$

In the following subsection we derive the closed form expression for the update of $z^\pm$ in case of the AROVR and SPROVR relaxations.

G.1. Fixed-point updates based on AROVR

Applying the AROVR step on the outermost level in Eq. 51 yields

$$U(y^+, y^-, z^+, z^-) = \alpha\ell(y^+) + \bar\alpha\ell(y^-) + F(y^+) - F(y^-) - (y^+ - y^-)^\top W \bar z + G(z^+) - G(z^-) - (z^+ - z^-)^\top x. \tag{53}$$

$\bar z = \alpha' z^+ + \bar\alpha' z^-$ uses a potentially different coefficient α'. By using the assumption in Eq. 52, $y^\pm$ are given by

$$
\begin{aligned}
0 &= \alpha(H_\ell y^+ + b_\ell) + H_F y^+ + b_F - Wz &\implies y^+ &= (H_F + \alpha H_\ell)^{-1}(Wz - b_F - \alpha b_\ell) \\
0 &= -\bar\alpha(H_\ell y^- + b_\ell) + H_F y^- + b_F - Wz &\implies y^- &= (H_F - \bar\alpha H_\ell)^{-1}(Wz - b_F + \bar\alpha b_\ell).
\end{aligned} \tag{54}
$$

Consequently,

$$
\begin{aligned}
y^+ - y^- &= (H_F + \alpha H_\ell)^{-1}(Wz - b_F - \alpha b_\ell) - (H_F - \bar\alpha H_\ell)^{-1}(Wz - b_F + \bar\alpha b_\ell) \\
&= \big((H_F + \alpha H_\ell)^{-1} - (H_F - \bar\alpha H_\ell)^{-1}\big)(Wz - b_F) - \big(\alpha(H_F + \alpha H_\ell)^{-1} - \bar\alpha(H_F - \bar\alpha H_\ell)^{-1}\big) b_\ell.
\end{aligned} \tag{55}
$$

Only the term involving Wz is of interest as the remaining ones are independent of z. Now define

$$M := (H_F + \alpha H_\ell)^{-1} - (H_F - \bar\alpha H_\ell)^{-1}, \tag{56}$$

then

$$
\begin{aligned}
&(H_F + \alpha H_\ell)M = \mathrm{I} - (H_F + \alpha H_\ell)(H_F - \bar\alpha H_\ell)^{-1} \\
\iff &(H_F + \alpha H_\ell)M(H_F - \bar\alpha H_\ell) = (H_F - \bar\alpha H_\ell) - (H_F + \alpha H_\ell) = -H_\ell.
\end{aligned} \tag{57}
$$

or

$$M = -(H_F + \alpha H_\ell)^{-1} H_\ell (H_F - \bar\alpha H_\ell)^{-1}. \tag{58}$$

A similar calculation shows that

$$
\begin{aligned}
M' &:= \alpha(H_F + \alpha H_\ell)^{-1} - \bar\alpha(H_F - \bar\alpha H_\ell)^{-1} \\
&= (H_F + \alpha H_\ell)^{-1}\big((1 - 2\alpha)H_F - 2\alpha\bar\alpha H_\ell\big)(H_F - \bar\alpha H_\ell)^{-1}.
\end{aligned}
$$

Finally,

$$
\begin{aligned}
y^+ - y^- &= M(Wz - b_F) - M'b_\ell \\
&= (H_F + \alpha H_\ell)^{-1} \left(H_\ell (H_F - \bar{\alpha} H_\ell)^{-1}(b_F - Wz) - ((1 - 2\alpha)H_F - 2\alpha\bar{\alpha}H_\ell)(H_F - \bar{\alpha}H_\ell)^{-1}b_\ell \right).
\end{aligned}
\tag{59}
$$

With $H_F = \mathtt{I}$ and $H_\ell = \beta\mathtt{I}$ we obtain

$$
M = -\frac{\beta}{(1 + \alpha\beta)(1 - \bar{\alpha}\beta)} \, \mathtt{I}.
\tag{60}
$$

Using the quadratic approximation for G, $G(z) = \frac{1}{2}z^\top H_G z + b_G^\top z$, then

$$
\begin{aligned}
z^+ &\leftarrow H_G^{-1} \left(x - b_G + \alpha' W^\top (y^+ - y^-) \right) = g\left(x + \alpha' W^\top (y^+ - y^-) \right) \\
z^- &\leftarrow H_G^{-1} \left(x - b_G - \bar{\alpha}' W^\top (y^+ - y^-) \right) = g\left(x - \bar{\alpha}' W^\top (y^+ - y^-) \right)
\end{aligned}
\tag{61}
$$

and

$$
\begin{aligned}
\alpha' z^+ + \bar{\alpha}' z^- &\leftarrow H_G^{-1} \left(x - b_G + \left((\alpha')^2 - (\bar{\alpha}')^2 \right) W^\top (y^+ - y^-) \right) \\
&= H_G^{-1} \left(x - b_G + (2\alpha' - 1)W^\top (y^+ - y^-) \right) \\
&= g\left(x + (2\alpha' - 1)W^\top (y^+ - y^-) \right) \\
&= g\left(x + (2\alpha' - 1)W^\top (MW\bar{z} + v) \right)
\end{aligned}
\tag{62}
$$

for a vector v containing all terms in $y^+ - y^-$ not dependent on $\bar{z}$. The choice $\alpha' = 1/2$ therefore implies that $\bar{z}$ is a fixed point of the above relation.

G.2. Fixed-point updates based on SPROVR

The above reasoning applies to the AROVR formulation. The SPROVR model uses a different objective,

$$
U(y^+, y^-, z^+, z^-) = \ell(\alpha y^+ + \bar{\alpha}y^-) + F(y^+) - F(y^-) - (y^+ - y^-)^\top Wz + G(z^+) - G(z^-) - (z^+ - z^-)^\top x
\tag{63}
$$

with corresponding stationary conditions

$$
\alpha\ell'(\bar{y}) + \nabla F(y^+) - W\bar{z} = 0 \qquad\qquad \bar{\alpha}\ell'(\bar{y}) - \nabla F(y^-) + W\bar{z} = 0.
\tag{64}
$$

By using a quadratic model for ℓ and F, this translates to

$$
(I) \quad \alpha\left(H_\ell\bar{y} + b_\ell\right) + H_F y^+ + b_F - W\bar{z} = 0 \qquad (II) \quad \bar{\alpha}\left(H_\ell\bar{y} + b_\ell\right) - H_F y^- - b_F + W\bar{z} = 0.
\tag{65}
$$

Adding these relation yields

$$
H_\ell\bar{y} + b_\ell + H_F(y^+ - y^-) = 0 \iff y^+ - y^- = -H_F^{-1}\left(H_\ell\bar{y} + b_\ell\right).
\tag{66}
$$

Further, $\alpha(I) - \bar{\alpha}(II)$ results in

$$
\begin{aligned}
0 &= (\alpha^2 - \bar{\alpha}^2)H_\ell\bar{y} + (\alpha^2 - \bar{\alpha}^2)b_\ell + H_F\bar{y} + b_F - W\bar{z} \\
&= ((2\alpha - 1)H_\ell + H_F)\,\bar{y} + (2\alpha - 1)b_\ell + b_F - W\bar{z} \\
\iff \bar{y} &= ((2\alpha - 1)H_\ell + H_F)^{-1}\left(W\bar{z} - (2\alpha - 1)b_\ell - b_F\right)
\end{aligned}
\tag{67}
$$

and

$$
y^+ - y^- = -H_F^{-1}\left(H_\ell\left((2\alpha - 1)H_\ell + H_F\right)^{-1}\left(W\bar{z} - (2\alpha - 1)b_\ell - b_F\right) + b_\ell \right).
\tag{68}
$$

Let M be the matrix applied on $W\bar{z}$, i.e.

$$
M = -H_F^{-1}H_\ell\left((2\alpha - 1)H_\ell + H_F\right)^{-1},
\tag{69}
$$

then for $H_F = \mathtt{I}$ and $H_\ell = \beta\mathtt{I}$ we obtain

$$M = -\frac{\beta}{1 + (2\alpha - 1)\beta}\,\mathtt{I} = -\frac{\beta}{1 - \beta + 2\alpha\beta}\,\mathtt{I}. \tag{70}$$

Recall the value of M before (Eq. 60). As

$$(1 + \alpha\beta)(1 - \bar{\alpha}\beta) = 1 - \beta + 2\alpha\beta - \alpha\bar{\alpha}\beta^2$$

and $\alpha\bar{\alpha}\beta^2 \geq 0$, we conclude that the SPROVR leads to a dampened error signal compared to AROVR (which is intuitively expected). The updates for $z^\pm$ are given by

$$
\begin{aligned}
z^+ &\leftarrow H_G^{-1}\left(x - b_G + \alpha'W^\top(y^+ - y^-)\right) = g\big(x + \alpha'W^\top(MW\bar{z} + v)\big) \\
z^- &\leftarrow H_G^{-1}\left(x - b_G - \bar{\alpha}'W^\top(y^+ - y^-)\right) = g\big(x - \bar{\alpha}'W^\top(MW\bar{z} + v)\big)
\end{aligned} \tag{71}
$$

for a suitable vector v independent of $\bar{z}$. Finally,

$$
\begin{aligned}
\alpha'z^+ + \bar{\alpha}'z^- &\leftarrow H_G^{-1}\left(x - b_G + \left((\alpha')^2 - (\bar{\alpha}')^2\right)W^\top(y^+ - y^-)\right) \\
&= H_G^{-1}\left(x - b_G + (2\alpha' - 1)W^\top(y^+ - y^-)\right) \\
&= g\big(x + (2\alpha' - 1)W^\top(y^+ - y^-)\big) \\
&= g\big(x + (2\alpha' - 1)W^\top(MW\bar{z} + v)\big),
\end{aligned} \tag{72}
$$

and we conclude that $\bar{z}$ is again a fixed point for the updates when $\alpha' = 1/2$.

19