PRINCIPLED BACKPROPAGATION-FREE TRAINING VIA LAYERWISE FENCHEL-YOUNG LIFTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training neural networks is typically performed using backpropagation, which tightly couples layers and requires storing the full computational graph. In this work, we study lifted training approaches, where intermediate activations are treated as optimization variables, and show that their underlying structure admits an exact characterization. Specifically, we prove that the compositional structure of neural networks is equivalent to the vanishing of Fenchel-Young gaps induced by Legendre functions, providing a principled measure of deviation from the forward pass. This perspective reveals that lifted activations act as regularized predictors, inducing an implicit regularization effect that improves generalization in practice. Building on this foundation, we propose LIFTY, a memory-efficient backpropagation-free training algorithm with closed-form updates supporting a broad class of monotone activations and convex regularizers.

1 INTRODUCTION

Training neural networks amounts to solving a highly non-convex optimization problem involving a composition of nonlinear transformations. The standard approach relies on backpropagation, which computes gradients through repeated applications of the chain rule across layers. While effective, this mechanism tightly couples all layers of the network and requires storing the full computational graph. Moreover, this recursive dependence can lead to vanishing or exploding gradients and strong sensitivity to depth and initialization (Glorot & Bengio, 2010; Schoenholz et al., 2017).
This has motivated a growing body of work on alternatives to backpropagation. These include *feedback alignment* and its variants, which replace exact gradient signals by fixed or random feedback pathways (Lillicrap et al., 2016; Nøkland, 2016), as well as a range of *biologically plausible* learning rules based on local updates (Whittington & Bogacz, 2017), target propagation (Lee et al., 2015; Meulemans et al., 2020), or equilibrium dynamics (Scellier & Bengio, 2017). While these approaches relax the requirement of exact gradient computation, they often come at the cost of weaker theoretical guarantees or less structured optimization procedures.

In this paper, we focus on a second family of methods, called *lifted* approaches which reformulate training by introducing intermediate activations as optimization variables and replacing the compositional structure of the network by layerwise constraints. A wide range of relaxations have been proposed to handle these constraints, including quadratic penalties (Carreira-Perpinan & Wang, 2014), ADMM-based methods (Zeng et al., 2018), proximal formulations (Li et al., 2020; Combettes & Pesquet, 2020), and Fenchel- or Bregman-type penalties (Gu et al., 2020; Wang & Benning, 2023). Despite their diversity, these approaches typically rely on penalty terms that are heuristic, activation-specific, or lack a clear underlying principle, which limits their interpretability and generality.
The main property we exhibit, which underlies the various contributions of this paper is that the compositional structure of neural networks admits an *exact characterization* through Fenchel-Young gaps induced by Legendre functions. Building on the representation of monotone activations as gradients of convex conjugates associated with Legendre functions (Frecon et al., 2022), we show that enforcing consistency with the forward pass is equivalent to driving a canonical Fenchel-Young gap to zero. This characterization is intrinsic to the activation function: the gap is uniquely determined by the underlying Legendre geometry and vanishes if and only if the compositional structure is satisfied. In this view, each layer is a regularized prediction problem (Blondel et al., 2020), and deviations from the forward pass are naturally quantified by this gap.
This characterization provides a principled foundation for lifted training: layerwise penalties arise

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.

Table 1: Positioning of LIFTY with respect to the state of the art. Unlike prior works, it provides interpretable updates, convergence guarantees, and closed-form implementations for many activations. We restrict the comparison to baselines and settings where *closed-form updates* are available.

Method	Activations	Regularization	Guarantees	Interpretable
(Gu et al., 2020)	ReLU	✗	✗	✗
(Wang & Benning, 2023)	1-homogeneous	ℓ_2^2 (activations)	✗	✗
LIFTY (ours)	monotone	✓(weights)	✓(unregularized)	✓

from the geometry of the activation function rather than from heuristic design, and can therefore be understood as canonical losses. As a consequence, it enables the systematic construction of training objectives for a broad class of activation functions, unifies several existing approaches, and induces an implicit regularization effect that improves generalization in practice. Building on these insights, we derive an efficient backpropagation-free training algorithm with closed-form updates and convergence guarantees.

Our contributions center on two advances that, to the best of our knowledge, have not been achieved by previous lifted approaches:

- **Connecting lifted and original training problems.** We establish conditions under which limit points of our lifted training algorithm satisfy the forward-pass constraints and are stationary points of the original unregularized neural network training objective.
- **A unified treatment of monotone activations and general weight regularizers.** We devise geometry-aware closed form updates, benefiting from convergence guarantees while supporting monotone activations and general convex weight regularizers.

These two contributions are tightly coupled and critically rely on our geometry-aware formulation and optimization of the lifted training problem. We incorporate them into LIFTY, a memory-efficient, backpropagation-free training algorithm requiring only one update per block at each iteration. All proofs are provided in the supplementary material.

Notations. For each layer l, let $\boldsymbol{X}_l \in \mathbb{R}^{n_l \times n_b}$ whose columns correspond to the inputs of the l-th layer, where $n_l \in \mathbb{N}^+$ is the layer width and $n_b \in \mathbb{N}^+$ is the batch size. Let $\boldsymbol{W}_l \in \mathbb{R}^{n_{l+1} \times n_l}$ and $c_l \in \mathbb{R}^{n_{l+1}}$ denote respectively the weight matrix and bias vector of the l-th layer. For brevity, we will write $\boldsymbol{W}_l \boldsymbol{X}_l$ instead of $\boldsymbol{W}_l \boldsymbol{X}_l + c_l$. In addition, we denote by small letters the functions applied pointwise, e.g., $f(\boldsymbol{X}_l) \in \mathbb{R}^{n_l \times n_b}$, and by capital letters their corresponding pointwise sum, e.g., $F(\boldsymbol{X}_l) = \sum_{i=1}^{n_b} \sum_{j=1}^{n_l} f(\boldsymbol{X}_l)_{i,j}$. Given some Hilbert space $\mathcal{H}$, we denote by $\Gamma_0(\mathcal{H})$ the space of functions from $\mathcal{H}$ to $]-\infty, +\infty]$ which are lower semicontinuous, proper and convex.

2 RELATED WORK ON LIFTED TRAINING

Given a dataset $(\boldsymbol{X}, \boldsymbol{y}) = \{X_i, y_i\}_{i=1}^n$ with $X_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}^c$, training a L-layer feedforward network amounts to

$$\underset{\{\boldsymbol{W}_l\}_{l=1}^L}{\text{minimize}} \ \mathcal{L}\big(\boldsymbol{W}_L \sigma\big(\boldsymbol{W}_{L-1} \cdots \sigma(\boldsymbol{W}_1 \boldsymbol{X})\big), \boldsymbol{y}\big)\,, \tag{1}$$

where σ is a pointwise nonlinearity, and $\mathcal{L}$ is a task-specific loss. This is a challenging non-convex problem typically solved via backpropagation, which computes gradients by repeated application of the chain rule across layers. The resulting gradient estimates suffer from well-known instabilities such as vanishing or exploding with depth, making optimization of deep networks difficult.

To address this, one common strategy is *variable splitting*, which simplifies such complex problems by decoupling the activations $\{\boldsymbol{X}_l\}$ and enforcing the network compositionality $\{\boldsymbol{X}_{l+1} = \sigma\left(\boldsymbol{W}_l \boldsymbol{X}_l\right)\}$ via penalties. The first work in this line traces back to (Carreira-Perpinan & Wang, 2014) where the authors replaced the hard constraints with soft quadratic penalties of the form

$$\underset{\{\boldsymbol{W}_l\}_{l=1}^L, \{\boldsymbol{X}_l\}_{l=1}^{L-1}}{\text{minimize}} \ \mathcal{L}(\boldsymbol{W}_L \boldsymbol{X}_L, \boldsymbol{y}) + \frac{\mu}{2} \sum_{l=1}^{L-1} \|\boldsymbol{X}_{l+1} - \sigma(\boldsymbol{W}_l \boldsymbol{X}_l)\|_F^2, \tag{2}$$

where $\boldsymbol{X}_1 \coloneqq \boldsymbol{X}$ is the input, $\{\boldsymbol{X}_i\}_{i=2}^L$ are intermediate layer activations, and $\mu > 0$ gradually increases along the iterations of their algorithmic solution. This 2-splitting formulation, where the problem is split into two types of variables (weights and activations), was later extended to a 3-splitting formulation in Zeng et al. (2018), where the authors also studied the global convergence of block coordinate descent schemes for such 2 and 3 splitting formulations. Additional 3-splitting techniques can be found in Taylor et al. (2016); Zhang et al. (2016) by adding Lagrange multipliers.

In another line, by leveraging the insight that many activation functions can be expressed as proximity operators (Combettes & Pesquet, 2020), Li et al. (2020) introduced the concept of *lifted proximal operator machines*. Their approach builds on the reformulation:

$$\underset{\{\boldsymbol{W}_l\}_{l=1}^L, \{\boldsymbol{X}_l\}_{l=2}^L}{\text{minimize}} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{X}_L, \boldsymbol{y}) \text{ s.t. } \boldsymbol{X}_{l+1} = \arg\min \left\{ F + \frac{1}{2}\|\cdot - \boldsymbol{W}_l \boldsymbol{X}_l\|_F^2 \right\}, \forall l \in \{1, \dots, L-1\}, \tag{3}$$

where the activation function σ is represented as the proximity operator of an appropriately chosen convex function f, i.e., $\sigma = \text{prox}_f = \arg\min_y f(y) + \frac{1}{2}\|y - \cdot\|^2$. They subsequently proposed a relaxation of the form

$$\underset{\{\boldsymbol{W}_l\}_{l=1}^L, \{\boldsymbol{X}_l\}_{l=2}^L}{\text{minimize}} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{X}_L, \boldsymbol{y}) + \sum_{l=1}^{L-1} \mu_l \left(F(\boldsymbol{X}_{l+1}) + \frac{1}{2}\|\boldsymbol{X}_l - \boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}\|_F^2 + G(\boldsymbol{W}_l \boldsymbol{X}_l) \right), \tag{4}$$

where each $\mu_l > 0$ and where the added term, with $g\colon x \mapsto \int_0^x (\sigma(y) - y)\mathrm{d}y$, is "retro-engineered" so that the constraints $\{\boldsymbol{X}_{l+1} = \sigma(\boldsymbol{W}_l \boldsymbol{X}_l)\}$ satisfy the optimality conditions of equation 4. The premises of this idea were already present in Askari et al. (2018), where the authors only considered the ReLU activation function and projected the lifted variables $\{\boldsymbol{X}_l\}$ onto the non-negative orthant.

Differently, Gu et al. (2020) relied on Fenchel duality to define a penalty term $b_\phi(v, u) = \phi(u) + \phi^*(v) - uv$ where ϕ is handcrafted as $\phi(u) = \int^u \sigma^{-1}(y)\mathrm{d}y$ so that b_ϕ is bi-convex. They show that, for such choice, the recursive constraints satisfy $\boldsymbol{X}_{l+1} = \sigma(\boldsymbol{W}_l \boldsymbol{X}_l) \Leftrightarrow B_\phi(\boldsymbol{W}_l \boldsymbol{X}_l, \boldsymbol{X}_{l+1}) \leq 0$ in order to motivate the following relaxation

$$\underset{\{\boldsymbol{W}_l\}_{l=1}^L, \{\boldsymbol{X}_l\}_{l=2}^L}{\text{minimize}} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{X}_L, \boldsymbol{y}) + \sum_{l=1}^{L-1} \mu_l B_\phi(\boldsymbol{W}_l \boldsymbol{X}_l, \boldsymbol{X}_{l+1}). \tag{5}$$

Independently, a more principled framework was recently devised in (Wang & Benning, 2023) where b_ϕ was connected with the proximal form of the activation function σ through $\phi = f + \frac{1}{2}\|\cdot\|^2$, leading to the penalty

$$B_\phi(\boldsymbol{W}_l \boldsymbol{X}_l,\ \boldsymbol{X}_{l+1}) = \left(\tfrac{1}{2}\|\cdot\|^2 + F\right)(\boldsymbol{X}_{l+1}) + \left(\tfrac{1}{2}\|\cdot\|^2 + F\right)^*(\boldsymbol{W}_l \boldsymbol{X}_l) - \langle \boldsymbol{X}_{l+1},\, \boldsymbol{W}_l \boldsymbol{X}_l \rangle, \tag{6}$$

whose authors coined as *generalized Bregman distance*. A key advantage is that differentiating B_ϕ with respect to $\boldsymbol{W}_{l-1}$ does not require computing the derivative of σ. The authors then minimize equation 5 using a proximal gradient descent scheme, which empirically show slow convergence.

Despite their diversity, the above approaches share several key limitations: (i) the penalty terms are often heuristically designed or lead to algorithmic schemes that lack a clear interpretation; (ii) convergence to the solution of the original problem equation 1 is not guaranteed; (iii) they are often restricted to specific activation functions, most notably ReLU-type nonlinearities.

In this work, we close this gap by revealing a fundamental connection that has been overlooked: we prove that the layer-wise penalty functions are *Fenchel-Young losses* induced by Legendre functions. This unification is consequential: it reveals that each layer acts as a regularized predictor (Blondel et al., 2020) which naturally regularizes the neural network and prevents from overfitting. In addition, it enables the design of a closed-form, theoretically grounded, backpropagation-free algorithm with convergence guarantees. Beyond optimization, this framework naturally accommodates many activation functions and regularization terms on the network weights, a flexibility not allowed by prior works (see Table 1).

3 Regularized lifted training via Fenchel-Young gaps

In this section, we present our lifting approach to address the training problem in equation 1 with the assumption that all activation functions σ are equal. It builds on the characterization of monotone

Table 2: Examples of activations σ induced by Fenchel-Young gaps $b_\psi(u, v) = \psi^*(u)+\psi(v)-\langle u, v\rangle$.

Activation σ	L_σ	$\operatorname{dom}\psi$	ψ	ψ^*
ISRU	1	$[-1, 1]$	$t \mapsto -\sqrt{1-t^2}$	$t \mapsto \sqrt{1+t^2}$
sigmoid	$\frac{1}{4}$	$[0, 1]$	$t \mapsto t\log t + (1-t)\log(1-t)$	$t \mapsto \log(1+\mathrm{e}^t)$
tanh	1	$[-1, 1]$	$t \mapsto \frac{1}{2}\log(1-t^2) + t\operatorname{arctanh}(t)$	$\log\cosh$
sin	1	$[-1, 1]$	$t \mapsto \sqrt{1-t^2} + t\arcsin(t)$	$-\cos$
$\mathrm{LeakyReLU}_\alpha$	1	$\mathbb{R}$	$t \mapsto \frac{t^2}{2}\big(\mathbf{1}_{t>0} + \alpha^{-1}\mathbf{1}_{t\le 0}\big)$	$t \mapsto \frac{t^2}{2}\big(\mathbf{1}_{t>0} + \alpha\mathbf{1}_{t\le 0}\big)$

activation functions via Legendre functions (Frecon et al., 2022) whose definition is recalled in Definition 2 of the appendix. We make the following assumption.

Assumption 1 (Activation function)**.** The activation function σ can be written as $\sigma = \operatorname{prox}_{\psi - \frac{1}{2}\|\cdot\|^2}$ where $\psi : \mathbb{R} \to \mathbb{R}$ is a $1/L_\sigma$-strongly convex Legendre function.

Remark 1. It follows from Assumption 1 that the activation function σ is L_σ-Lipschitz.

This assumption is satisfied for many popular activations such as the ones listed in Table 2. The connection between σ and ψ becomes clearer by applying the definition of the proximal operator, resulting in a novel viewpoint based on regularized predictor (Blondel et al., 2020).

Lemma 1 (Regularized prediction)**.** Under Assumption 1,

$$\boldsymbol{X}_{l+1} = \sigma(\boldsymbol{W}_l\boldsymbol{X}_l) = \arg\max_{\boldsymbol{X}} \langle \boldsymbol{X}, \boldsymbol{W}_l\boldsymbol{X}_l\rangle - \Psi(\boldsymbol{X}), \tag{7}$$

where Ψ is the coordinate-wise sum of ψ.

The objective in equation 7 balances a *regularization* term $\Psi(x)$, which penalizes outputs far from some reference geometry, against an *affinity* term that rewards alignment with the preactivation. Different choices of Ψ yield different activation. Since Ψ is strictly convex under Assumption 1, the maximizer is unique and corresponds to $\boldsymbol{X}_{l+1}$.

The value of this supremum defines the *Fenchel conjugate* of Ψ (Fenchel & Blackett, 1953; Bauschke & Combettes, 2017):

$$\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l) := \sup_{\boldsymbol{X}} \langle \boldsymbol{X}, \boldsymbol{W}_l\boldsymbol{X}_l\rangle - \Psi(\boldsymbol{X}) \ge \langle \boldsymbol{X}, \boldsymbol{W}_l\boldsymbol{X}_l\rangle - \Psi(\boldsymbol{X}), \quad \forall \boldsymbol{X} \in \operatorname{dom}\Psi. \tag{8}$$

By strict convexity of Ψ, the supremum is uniquely attained and, by Danskin's theorem, $\sigma(\boldsymbol{W}_l\boldsymbol{X}_l) = \nabla\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l)$. Moreover Ψ^* is L_σ-gradient Lipschitz. This conjugate representation is standard for regularized predictors (Blondel et al., 2020), but here it takes on a new meaning: the gap between the LHS and RHS in equation 8 is a Fenchel-Young gap B_Ψ (Blondel et al., 2020; Sklaviadis et al., 2025), which is non-negative by Fenchel's inequality and vanishes *if and only if* $\boldsymbol{X} = \nabla\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l) = \boldsymbol{X}_{l+1}$.

Lemma 2 (Vanishing Fenchel-Young gap)**.** Let B_Ψ denotes Fenchel-Young gap associated to Ψ, i.e., $B_\Psi(\boldsymbol{U}, \boldsymbol{V}) := \Psi^*(\boldsymbol{U}) + \Psi(\boldsymbol{V}) - \langle \boldsymbol{U}, \boldsymbol{V}\rangle \ge 0$. Then, under Assumption 1, we have

$$\boldsymbol{X}_{l+1} = \sigma(\boldsymbol{W}_l\boldsymbol{X}_l) \quad \Leftrightarrow \quad B_\Psi(\boldsymbol{W}_l\boldsymbol{X}_l, \boldsymbol{X}_{l+1}) = 0. \tag{9}$$

This class of gaps is in fact familiar from supervised learning: the squared loss corresponds to $\zeta = \frac{1}{2}\|\cdot\|_F^2$, and the binary cross-entropy to $\zeta : t \mapsto t\log t + (1-t)\log(1-t)$, both of which are Fenchel-Young gaps for suitable ζ (Blondel et al., 2020).

While Blondel et al. (2020) use Fenchel-Young losses to construct (regularized) training objectives, we show that the same gap characterizes the layer's forward pass itself: the network's activation is precisely the unique point at which the Fenchel-Young gap vanishes. By leveraging both layer's forward passes and training losses characterization, we consider the following relaxation.

Definition 1 (Fenchel-Young Training)**.** Given some dataset $(\boldsymbol{X}_1, \boldsymbol{X}_{L+1}) := (\boldsymbol{X}, \boldsymbol{y})$, some parameters $\boldsymbol{\mu} = [\mu_l]_{l=1}^{L-1} \in \mathbb{R}_{>0}^{L-1}$, and regularization functions $G_l \in \Gamma_0(\mathbb{R}^{n_{l+1}\times n_l})$, solve

$$\underset{\boldsymbol{W},\boldsymbol{X}}{\text{minimize}}\, \mathcal{F}(\boldsymbol{W}, \boldsymbol{X}) := \left[B_\Phi(\boldsymbol{W}_L\boldsymbol{X}_L, \boldsymbol{X}_{L+1}) + \sum_{l=1}^{L-1} \mu_l B_\Psi(\boldsymbol{W}_l\boldsymbol{X}_l, \boldsymbol{X}_{l+1}) + \sum_{l=1}^{L} G_l(\boldsymbol{W}_l) \right], \tag{10}$$

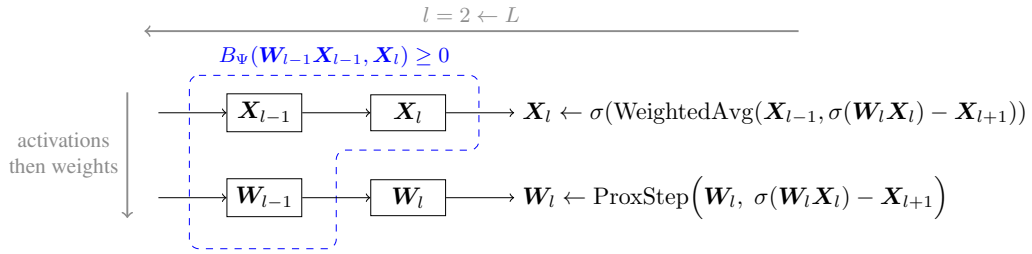


Figure 1: Illustration of the penalty B_ψ and the updates coupling the lifted variables in LIFTY. Activations are first updated sequentially: each $\boldsymbol{X}_l$ is obtained by applying the nonlinearity to a weighted average equation 14 between the previous-layer prediction $\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}$ and its current inverse-activation estimate $\sigma^{-1}(\boldsymbol{X}_l)$. Then, weights are updated sequentially via the proximal step equation 20, correcting the forward prediction error $\sigma(\boldsymbol{W}_l\boldsymbol{X}_l) - \boldsymbol{X}_{l+1}$ while incorporating regularization.

where $\boldsymbol{W} \coloneqq \{\boldsymbol{W}_l\}_{l=1}^L$, $\boldsymbol{X} \coloneqq \{\boldsymbol{X}_l\}_{l=2}^L$, and $B_\Phi = \mathcal{L}$ is the Fenchel-Young loss associated with the supervision loss.

Proposition 1. Any collection of activation variables satisfying the forward-pass recursion constraints $\{X_l = \sigma(W_{l-1}X_{l-1})\}_{l=2}^{L-1}$ is stationary with respect to the activation variables of the Fenchel-Young lifted optimization problem in 10.

Remark 2. When $G_l = 0$ and each activation matrix X_l has full column rank, every stationary point of 10 satisfies the constraints $\{X_l = \sigma(W_{l-1}X_{l-1})\}_{l=2}^{L-1}$, and the equivalence holds.

Remark 3. The premises of such objective function can be found in Wang & Benning (2023) for some $\Psi = F + \frac{1}{2}\|\cdot\|^2$ without explicit connection with regularized predictors and Legendre functions. In addition, their algorithmic solution cannot handle regularizations G_l since it would require computing the proximal operator of $F + G_l$ which does not have a closed-form expression.

In what follows, we detail our algorithmic solution for solving equation 10 leveraging to the fullest the properties of Ψ and Ψ^*. Its convergence properties will be studied in Section 5 and evaluated empirically in 6. For convenience, we remove the subscripts l in G_l.

4 PROPOSED BACKPROPAGATION-FREE ALGORITHMIC SOLUTION

The proposed solution relies on the observation that the objective in equation 10 is *multiconvex*, i.e., it is convex in each block $(\boldsymbol{W}_l, \boldsymbol{X}_l)$ separately when all others are fixed. This structure suggests a natural *block coordinate descent* (BCD) strategy: alternately minimize over weights and activations, one layer at a time. Hereafter, we consider an *inexact* variant where each update performs a single step to decrease the overall objective. The resulting algorithmic scheme is sketched in Figure 1.

4.1 UPDATING THE WEIGHTS

For some fixed activations, updating each $\boldsymbol{W}_l$ via an *exact* BCD algorithm amounts in minimizing $\mu_l B_\psi(\boldsymbol{W}_l\boldsymbol{X}_l, \boldsymbol{X}_{l+1}) + G(\boldsymbol{W}_l)$, which is equivalent to

$$\underset{\boldsymbol{W}_l}{\text{minimize}}\ \ \Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l) - \langle \boldsymbol{W}_l\boldsymbol{X}_l, \boldsymbol{X}_{l+1}\rangle + \mu_l^{-1}G(\boldsymbol{W}_l). \tag{11}$$

Since Ψ^* is L_σ-gradient Lipschitz, we propose to apply a forward-backward algorithm.

Proposition 2 ($\boldsymbol{W}_l$-update)**.** Under Assumption 1, for any step-size $0 < \gamma \leq 1/(L_\sigma\|\boldsymbol{X}_l\|^2)$, the iterates defined by

$$\boldsymbol{W}_l^{(k+1)} \leftarrow \text{prox}_{(\gamma/\mu_l)G}\Big(\boldsymbol{W}_l^{(k)} - \gamma\big[\sigma(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l) - \boldsymbol{X}_{l+1}\big]\boldsymbol{X}_l^\top\Big) \tag{12}$$

converge to the minimizer of equation 11.

Intuitively, the gradient term measures the residual between the current post-activations $\sigma(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l)$ and the target activations $\boldsymbol{X}_{l+1}$, and propagates this error to update the weights through the input $\boldsymbol{X}_l$.

For common regularizers, $\mathrm{prox}_{(\gamma/\mu_l)G}$ is available in closed form, keeping the update equation 20 computationally cheap. We refer to Parikh & Boyd (2014); Combettes & Pesquet (2011) for a comprehensive treatment of proximal operators.

Remark 4. In Section B.2, we also propose a precondioned variant, analogous to using only the damped input-side factor of K-FAC Martens & Grosse (2015), to whiten the input directions, hence removing the $\|\boldsymbol{X}_l\|^2$ dependence from the step-size bound.

Remark 5. For the output layer $l = L$, the weight subproblem is analogous to equation 11, with the hidden-layer potential Ψ replaced by the output potential Φ. Hence, Proposition 2 extends to the output layer by replacing $\sigma = \nabla\Psi^*$ with $\nabla\Phi^*$ and L_σ with $L_{\nabla\Phi^*}$. The corresponding block smoothness constant is $L_{\nabla\Phi^*}\|\boldsymbol{X}_L\|_2^2$.

4.1.1 Updating the activations

We now turn to the update of the activations for some fixed weights. To define the update rule, we again study the optimization problem over a single $\boldsymbol{X}_l$ with $l \neq L$, that is

$$\underset{\boldsymbol{X}_l}{\text{minimize}}\ \mu_{l-1}(\Psi(\boldsymbol{X}_l) - \langle \boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}, \boldsymbol{X}_l\rangle) + \mu_l(\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l) - \langle \boldsymbol{W}_l\boldsymbol{X}_l, \boldsymbol{X}_{l+1}\rangle). \tag{13}$$

Unlike the weight update subproblem, this objective involves two Fenchel-Young gaps: one in which $\boldsymbol{X}_l$ acts as a preactivation and one in which it operates as a postactivation.

By leveraging that (i) Ψ^* is L_σ-gradient Lipschitz and (ii) Ψ is $(1/L_\sigma)$-strongly convex so its associated Bregman divergence dominates Euclidean distances, we devise the following update rule.

Proposition 3 ($\boldsymbol{X}_l$-Update)**.** Under Assumption 1, the iterates $\{\boldsymbol{X}_l^{(k)}\}_{k\in\mathbb{N}}$ converge to the minimizer of equation 13

$$\boldsymbol{X}_l^{(k+1)} \leftarrow \sigma\left(\frac{\mu_{l-1}\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1} + \mu_l L_\sigma^2\|\boldsymbol{W}_l\|^2\sigma^{-1}(\boldsymbol{X}_l^{(k)}) - \mu_l\boldsymbol{W}_l^\top(\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}) - \boldsymbol{X}_{l+1})}{\mu_{l-1} + \mu_l L_\sigma^2\|\boldsymbol{W}_l\|^2}\right). \tag{14}$$

The reader is invited to refer to Section C.3 of the appendix for a detailed discussion of the rationale leading to this update choice. Two keys points emerged. First, only equation 14 exploits the geometry induced by Ψ (and indirectly by σ), which leads to a closed-form expression together with favorable descent properties. Second, it converges regardless of the parameters $\{\mu_l\}$ and $\|\boldsymbol{W}_l\|^2$, thereby enabling the coupling strategy discussed later in Remark 9 to provide provable convergence guarantees.

For the case $l = L$, we report a general update rule in Section C.3.2 to handle the fact that the potentials Φ and Ψ may be different. Combined with the weight update equation 20, these updates yield a fully backpropagation-free layerwise training procedure. In the next section, we present the full algorithmic procedure.

4.2 From block-coordinate descent to a memory-efficient warm-start scheme

A well-known limitation of lifted training methods is the memory required to store and update auxiliary activations for every sample and layer. In our BCD scheme, activation updates depend on the previous iterate, requiring $\{\boldsymbol{X}_l\}$ to be retained across iterations. Unlike standard backpropagation, mini-batching alone therefore does not reduce this memory cost, limiting scalability. Building on (Li et al., 2020; Wang & Benning, 2023), we introduce LIFTY, a *memory-efficient warm-start strategy* tailored to our scheme (Algorithm 1), which reinitializes activations via a forward pass at each iteration instead of storing them. We now characterize the resulting activation updates under this warm-start regime.

Proposition 4. Let $l \in \{2, \dots, L-1\}$. Under Assumption 1, if the iterates $\{\boldsymbol{X}_j\}_{j=1,\dots,l}$ have been updated sequentially using the warm-start (i.e., forward-pass) and the update equation 14 for the l-th activation $\boldsymbol{X}_l$ boils down to

$$\boldsymbol{X}_l^{(1)} \leftarrow \sigma\left(\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1} - \frac{\mu_l}{\mu_{l-1} + \mu_l L_\sigma^2\|\boldsymbol{W}_l\|^2}\boldsymbol{W}_l^\top\left[\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(0)}) - \boldsymbol{X}_{l+1}\right]\right). \tag{15}$$

Algorithm 1 LIFTY: Lifted training via Fenchel-Young lifting

Require: Dataset $\mathcal{D}$ and network weights $\{\boldsymbol{W}_l\}_{l=1}^L$.
1: **repeat**
2: GET ACTIVATIONS FOR THIS BATCH
3: Sample $(\boldsymbol{X}_1, \boldsymbol{X}_{L+1}) := (\boldsymbol{X}, \boldsymbol{y}) \sim \mathcal{D}$
4: Get the activations by forward pass $\boldsymbol{X}_l \leftarrow \sigma(\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1})$
5: UPDATE PARAMETERS
6: Update activations $\{\boldsymbol{X}_l\}_{l=2}^L$ via equation 15 and equation 34
7: Update weights $\{\boldsymbol{W}_l\}_{l=1}^L$ via equation 20
8: **until** convergence

The update equation 15 allows the algorithm to be σ^{-1} free and admits an interpretation similar to Proposition 2. We discuss the memory efficiency in Appendix F.

While this modification breaks the global convergence guarantees of BCD, a key contribution of this work is to show that it still admits a principled justification and convergence guarantees. Unlike (Li et al., 2020; Wang & Benning, 2023), however, our convergence results hold while performing only a single closed-form update per inner subproblem (see steps 5-6 in Algorithm 1), without requiring the subproblems to be solved to optimality. This is a key practical distinction from prior works, as it substantially reduces the computational cost of each outer iteration (e.g., speedups of approximately $15\times$ and $4.5\times$, as reported in Table 4, compared to (Bauschke et al., 2018) and (Wang & Benning, 2023), respectively) while still enjoying convergence guarantees.

5 CONVERGENCE GUARANTEES

We now present theoretical guarantees for Algorithm 1 in the case $G_l \equiv 0$. Our analysis proceeds in two steps: we first establish convergence to a stationary point of the lifted objective equation 10, and then show that such points correspond to stationary points of the original neural network training problem equation 1, thereby bridging the two formulations.

Our analysis requires the following assumption, which ensures that the lifted objective upper-bounds the original objective evaluated at the forward pass.

Assumption 2 (Forward pass decrease)**.** The iterates $(\{\boldsymbol{W}_l\}, \{\boldsymbol{X}_l\})$ of Algorithm 1 satisfy

$$B_\Phi(\boldsymbol{W}_L^{(k)}\boldsymbol{X}_L^{(k)}, \boldsymbol{X}_{L+1}^{(k)}) + \sum_{l=1}^{L-1} \mu_l B_\Psi(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l^{(k)}, \boldsymbol{X}_{l+1}^{(k)}) \geq B_\Phi(\hat{\boldsymbol{y}}^{(k)}, \boldsymbol{y}) - \varepsilon_k, \tag{16}$$

where $\hat{\boldsymbol{y}} = \boldsymbol{W}_L^{(k)}\sigma\big(\boldsymbol{W}_{L-1}^{(k)} \cdots \sigma(\boldsymbol{W}_1^{(k)}\boldsymbol{X})\big)$ and $\{\varepsilon_k\}_k$ is a nonnegative summable sequence measuring the discrepancy between the lifted and compositional neural network objectives.

While not provable in full generality, it holds in several important cases (see Remark 7)) and is naturally satisfied near convergence, where all gaps vanish simultaneously. Experiments conducted in Section 2 show that ε_k rapidly decays during training and is equal to zero for most iterations, providing empirical support.

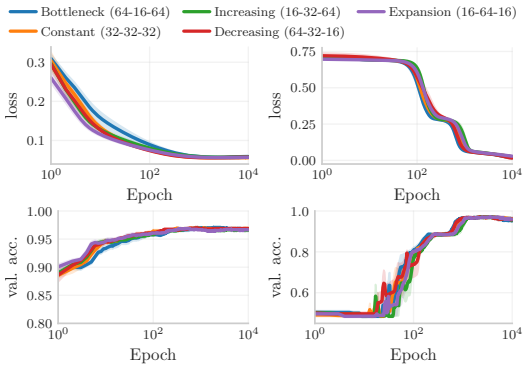
We are now equipped to provide our main convergence results[1].

Theorem 1. *Under Assumptions 1 and 2, with fixed $\mu_l > 0$ and $G_l \equiv 0$, suppose the sequence generated by Algorithm 1 is bounded. Assume that the weight step sizes satisfy $1/\gamma_l^{(k)} - (L_\sigma/2)\|\boldsymbol{X}_l^{(k+1)}\|_F^2 \geq c_w > 0$ for all l, k, and that $(\mu_{L-1}/L_\sigma) - L_{\nabla\Phi^*}\|\boldsymbol{W}_L^{(k)}\|_2^2 \geq 2c_x > 0$ for all k. Then every limit point is a stationary point of the lifted objective equation 10.*

We next connect these limit points to the original training problem.

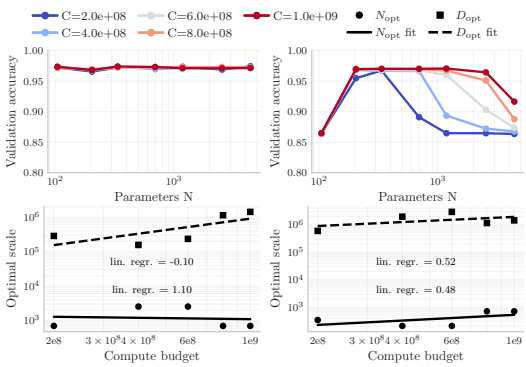
Theorem 2 (Equivalence with original problem)**.** *Under the assumptions of Theorem 1, every limit point generated by Algorithm 1 $(\{\boldsymbol{W}_l^*\}, \{\boldsymbol{X}_l^*\})$ satisfies $\boldsymbol{X}_{l+1}^* = \sigma(\boldsymbol{W}_l^*\boldsymbol{X}_l^*)$ for all $l \in \{1, \ldots, L-1\}$, and $\{\boldsymbol{W}_l^*\}$ is a stationary point of the original objective equation 1.*

[1]For the mini-batch setting, we give an expected finite-step bound under bounded-variance stochastic gradients; see Theorem 4.



(a) LIFTY (b) Adam

Figure 2: Training dynamics across MLP architectures. Top: training objective. Bottom: validation accuracy.



(a) LIFTY (b) Adam

Figure 3: Compute-optimal scaling under fixed compute budgets C. Top: validation accuracy. Bottom: empirical scaling laws.

The preceding results characterize every limit point. The following theorem strengthens this conclusion to convergence of the entire sequence.

Theorem 3 (Convergence of the iterates)**.** *Under the assumptions of Theorem 1, consider the full-batch algorithm with activation updates ordered as $L, \dots, 2$. If $\mathcal{F}$ satisfies the Kurdyka–Łojasiewicz property, $\inf_{l,k} \gamma_l^{(k)} > 0$, and $\varepsilon_k = 0$ for all sufficiently large k in Assumption 2, then the sequence has finite length and converges to a stationary point of equation 10, whose weights are stationary for equation 1.*

Note that the KL assumption holds whenever the task loss $\mathcal{L}(\cdot, \boldsymbol{y})$ and the activation potentials Ψ, Ψ^* are definable in a common o-minimal structure, provided the resulting lifted objective is proper and lower semicontinuous (Attouch et al., 2010). This includes squared and cross-entropy losses combined with the potentials associated with sigmoid, tanh, and leaky-ReLU activation.

Together, these results establish stationarity of every limit point for the original training problem and, under the additional assumptions of Theorem 3, convergence of the entire sequence to such a point, without backpropagation and with closed-form layerwise updates.

The algorithm itself permits nonzero G_l, but extending these guarantees and obtaining a nontrivial forward-consistent limit point in that setting would require a separate penalty-continuation analysis with iteration-dependent μ_l, which is outside the scope of this paper.

6 NUMERICAL EXPERIMENTS

We first study the optimization behavior of LIFTY on a controlled setting, before comparing it against lifting methods on standard datasets from the lifted literature. Details are reported in Section D.

6.1 A CONTROLLED STUDY OF TRAINING DYNAMICS

The Two Moons dataset provides a simple benchmark to study the optimization behavior of training methods. Since standard MLP architectures can easily fit the task, differences in training dynamics can be analyzed independently of capacity limitations.

Training dynamic. Figure 2 compares the training dynamics of LIFTY and Adam across several representative MLP architectures. Although all configurations eventually reach similar validation accuracy, the optimization trajectories differ significantly. In particular, LIFTY exhibits smoother and more homogeneous behavior across architectures, both in terms of objective decrease and validation accuracy. In contrast, Adam shows stronger architecture-dependent dynamics, with delayed transition phases and larger variability during training. Interestingly, some architectures remain close to random guessing for a substantial number of iterations before abruptly transitioning to a good solution, highlighting a stronger sensitivity to the network parameterization. These results suggest that the

Table 3: Test accuracy of lifting approaches and Back-propagation across benchmark datasets.

Method	MNIST	Fashion-MNIST	Sonar	Covertype
LIFTY (ours)	$\mathbf{97.2 \pm 0.1\%}$	$\mathbf{87.5 \pm 0.1\%}$	$\mathbf{82.4 \pm 4.9\%}$	$\mathbf{78.3 \pm 0.3\%}$
FY-Adam	$94.2 \pm 0.1\%$	$85.3 \pm 0.2\%$	$68.1 \pm 8.0\%$	$69.6 \pm 0.2\%$
FY-SGD	$94.0 \pm 0.2\%$	$84.6 \pm 0.3\%$	$71.9 \pm 6.4\%$	$71.0 \pm 0.2\%$
(Gu et al., 2020)	$82.6 \pm 0.4\%$	$75.8 \pm 0.4\%$	$67.6 \pm 3.1\%$	$55.3 \pm 0.8\%$
(Wang & Benning, 2023)	$91.8 \pm 0.6\%$	$82.6 \pm 0.4\%$	$67.6 \pm 2.7\%$	$71.2 \pm 0.1\%$
Backprop (Adam)	$98.6 \pm 0.0\%$	$89.6 \pm 0.1\%$	$81.0 \pm 2.4\%$	$93.9 \pm 0.2\%$

Fenchel-Young lifting induces a more stable optimization geometry that is less sensitive to the architectural shape of the network.

Compute-optimal scaling. Inspired by compute-optimal scaling analyses in large-scale language models (Hoffmann et al., 2022), Figure 3 studies compute allocation under the budget $C \simeq ND$, where N is the number of trainable parameters and D is the total number of training examples processed. For the fixed-depth one-hidden-layer MLP 2-w-1, the number of parameters is $N = 4w+1$, including biases. Thus, for a fixed compute budget, increasing the width reduces the available token budget $D \approx C/N$. The results show that LIFTY reaches high validation accuracy for essentially all tested widths and compute budgets, suggesting that performance is weakly affected by this width-budget trade-off. The fitted scaling further indicates that additional compute is mostly allocated to larger token budgets rather than wider models. In contrast, Adam displays a more visible dependence on the compute allocation, with estimated optima following $N_{\text{opt}} \sim C^{0.48}$ and $D_{\text{opt}} \sim C^{0.58}$, reminiscent of the balanced compute-optimal scaling observed in (Hoffmann et al., 2022).

6.2 BENCHMARK COMPARISONS ACROSS DATASETS

Finally, we compare the lifting approaches presented in Table 1 with backpropagation (Adam). We also include two natural approaches, denoted FY-Adam and FY-SGD, where the lifted objective equation 10 is optimized via a warm-start BCD scheme alternating between $\{\boldsymbol{X}\}$ and $\{\boldsymbol{W}\}$, using Adam or SGD updates, respectively. These baselines isolate the contribution of our proposed updates beyond the shared lifted formulation and warm-start scheme (see Remark 10).

Comparison against baselines. Mean results over five seeds, together with the corresponding standard deviations, are reported in Table 3. Among the backpropagation-free approaches considered, LIFTY shows consistently the strongest performance across the benchmark datasets. Moreover, since LIFTY overpass FY-Adam and FY-SGD, it highlights the benefit of the geometry-aware updates beyond the Fenchel-Young Training. Standard backpropagation (Adam) remains stronger on most of these benchmarks, that provides a reference for the trade-off between conventional training and backpropagation-free approaches.

7 CONCLUSION

We introduced LIFTY, a memory-efficient, backpropagation-free training method based on Legendre-induced Fenchel–Young gaps. This formulation connects forward-pass consistency to the geometry of the activations and yields explicit layerwise updates accommodating monotone activations and general convex weight regularizers. Our analysis connects the algorithm's limit points to stationary points of the original unregularized training objective. Our experiments demonstrate competitive performance among lifted approaches and reduced activation-storage requirements.
Beyond standard neural networks, this perspective motivates extensions to composite models for which end-to-end differentiation is unavailable or computationally demanding. Promising directions include hybrid solvers (Um et al., 2020; Mezidi et al., 2026) in physics-based machine learning and plug-and-play reconstruction methods in imaging (Hurault et al., 2023), where numerical solvers or denoisers may lack accessible derivatives. Extending our formulation and guarantees to such components is an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools for implementing part of the methods. We have not used generative AI tools for developing the theoretical models or conceptual frameworks or for any of the other tasks requiring disclosure. Additionally, we used generative AI tools for editing software code and improve readability. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI. .

ETHICS STATEMENT

This work develops an optimization method for training neural networks. We do not identify specific ethical concerns arising from the proposed method beyond those associated with the data and applications in which it may be used.

REPRODUCIBILITY STATEMENT

Algorithm 1 summarizes LIFTY, with update rules given in equation 20, equation 15, and equation 34. The theoretical assumptions are stated alongside the results, and their proofs are provided in the appendix. Section D documents the datasets, preprocessing, data splits, network architectures, training settings, baseline configurations, and random seeds used in the experiments. Results are aggregated over five runs to assess variability. Section F describes the setup used to compare memory consumption.

REFERENCES

Armin Askari, Geoffrey Negiar, Rajiv Sambharya, and Laurent El Ghaoui. Lifted neural networks. *arXiv preprint arXiv:1805.01532*, 2018.

Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka–łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010. doi: 10.1287/moor.1100.0449.

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer International Publishing, 2017. ISBN 9783319483115. doi: 10.1007/978-3-319-48311-5.

Heinz H Bauschke, Jonathan M Borwein, et al. Legendre functions and the method of random bregman projections. *Journal of convex analysis*, 4(1):27–67, 1997.

Heinz H Bauschke, Minh N Dao, and Scott B Lindstrom. Regularizing with bregman–moreau envelopes. *SIAM Journal on Optimization*, 28(4):3208–3228, 2018.

Mathieu Blondel, André FT Martins, and Vlad Niculae. Learning with fenchel-young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020.

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

Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220, 2010.

Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In Heinz H. Bauschke, Regina S. Burachik, Patrick L. Combettes, Veit Elser, D. Russell Luke, and Henry Wolkowicz (eds.), *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 185–212. Springer, New York, 2011.

Patrick L. Combettes and Jean-Christophe Pesquet. Deep neural network structures solving variational inequalities. *Set-Valued and Variational Analysis*, 28(3):491–518, February 2020. ISSN 1877-0541.

Werner Fenchel and Donald W Blackett. *Convex cones, sets, and functions*. Princeton University, Department of Mathematics, Logistics Research Project, 1953.

Jordan Frecon, Gilles Gasso, Massimiliano Pontil, and Saverio Salzo. Bregman neural networks. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 6779–6792. PMLR, 17–23 Jul 2022.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Samuel Hurault, Ulugbek Kamilov, Arthur Leclaire, and Nicolas Papadakis. Convergent bregman plug-and-play image restoration for poisson inverse problems. *Advances in Neural Information Processing Systems*, 36:27251–27280, 2023.

Dong-Hyun Lee, Saizheng Zhang, Asja Fischer, and Yoshua Bengio. Difference target propagation. In *Joint european conference on machine learning and knowledge discovery in databases*, pp. 498–515. Springer, 2015.

Jia Li, Mingqing Xiao, Cong Fang, Yue Dai, Chao Xu, and Zhouchen Lin. Training neural networks by lifted proximal operator machines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3334–3348, 2020.

Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1): 13276, 2016.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

Alexander Meulemans, Francesco Carzaniga, Johan Suykens, João Sacramento, and Benjamin F Grewe. A theoretical framework for target propagation. *Advances in Neural Information Processing Systems*, 33:20024–20036, 2020.

Abdel-Rahim Mezidi, Jordan Patracone, and Amaury Habrard. Proximal splitting methods for hybrid differentiable models. In *Forty-third International Conference on Machine Learning*, 2026.

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

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3): 127–239, 2014.

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

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation, 2017.

Sophia Sklaviadis, Thomas Moellenhoff, Andre Martins, and Mario Figueiredo. Fenchel-young variational learning. *arXiv preprint arXiv:2502.10295*, 2025.

Gavin Taylor, Ryan Burmeister, Zheng Xu, Bharat Singh, Ankit Patel, and Tom Goldstein. Training neural networks without gradients: A scalable admm approach. In *International conference on machine learning*, pp. 2722–2731. PMLR, 2016.

Kiwon Um, Robert Brand, Yun Raymond Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. *Advances in neural information processing systems*, 33:6111–6122, 2020.

University of California, Irvine. Uci machine learning repository. `http://archive.ics.uci.edu/ml`, 2018. School of Information and Computer Science, University of California, Irvine, CA, USA.

Xiaoyu Wang and Martin Benning. Lifted bregman training of neural networks. *Journal of Machine Learning Research*, 24(232):1–51, 2023.

James CR Whittington and Rafal Bogacz. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation*, 29(5): 1229–1262, 2017.

Jinshan Zeng, Tim Tsz-Kit Lau, Shaobo Lin, and Y. Yao. Global convergence of block coordinate descent in deep learning. In *International Conference on Machine Learning*, 2018.

Huaqing Zhang, Jian Wang, Zhanquan Sun, Jacek M Zurada, and Nikhil R Pal. Feature selection for neural networks using group lasso regularization. *IEEE Transactions on Knowledge and Data Engineering*, 32(4):659–673, 2019.

Ziming Zhang, Yuting Chen, and Venkatesh Saligrama. Efficient training of very deep neural networks for supervised hashing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1487–1495, 2016.

A Mathematical facts and reminders

We recall below the definition of a Legendre function Bauschke et al. (1997).

Definition 2 (Legendre function)**.** Let $\mathcal{X}$ be a Hilbert space and $\psi \in \Gamma_0(\mathcal{X})$. Then ψ is called

- *essentially smooth* if ψ is differentiable on $\operatorname{int}\operatorname{dom}\psi \neq \emptyset$ and $\|\nabla\psi(v)\| \to +\infty$ as $v \mapsto \operatorname{boundary}\operatorname{dom}\psi$,
- *essentially strictly convex* if ψ is strictly convex on every convex subset of $\operatorname{dom}\partial\psi$,
- *Legendre* if ψ is both essentially smooth and essentially strictly convex.

B Additional discussions

In this section, we provide several discussions worth sharing.

B.1 General remarks

Remark 6 (Exact $\boldsymbol{W}_l$ update)**.** When $G \equiv 0$, the problem in equation 11 is equivalent to a Bregman regression problem

$$\underset{\boldsymbol{W}_l}{\text{minimize}}\ D_{\Psi^*}(\boldsymbol{W}_l\boldsymbol{X}_l, \sigma^{-1}(\boldsymbol{X}_{l+1})), \tag{17}$$

where D_{ψ^*} is the Bregman divergence generated by ψ^*. It has a unique closed-form solution $\boldsymbol{W}_l^* = \sigma^{-1}(\boldsymbol{X}_{l+1})\boldsymbol{X}_l^\dagger$ where $\boldsymbol{X}_l^\dagger$ is the pseudo inverse of $\boldsymbol{X}_l$. However, this closed form is numerically impractical: for saturating activations such as sigmoid or softmax, σ^{-1} diverges as $\boldsymbol{X}_{l+1}$ approaches the boundary of its range, leading to numerically unstable targets. In practice, we therefore retain the proximal update equation 20, which avoids the numerical instability of the exact Bregman regression solution while preserving a simple layerwise interpretation.

Remark 7 (When the assumption is verified)**.** The assumption holds trivially for *single layer networks* ($L = 1$) with $\varepsilon_k = 0$, as both sides equal $B_\psi(\boldsymbol{W}_L\boldsymbol{X}_1, y)$. For *linear networks* ($\sigma = \mathrm{Id}$), the assumption holds under suitable Lipschitz control of the weight products and choice of μ_l. *Near convergence*, the activations approach their forward pass values and all gaps vanish simultaneously, so both sides tend to $B_\psi(\hat{\boldsymbol{y}}, y)$ and the assumption holds with approximate equality.

Remark 8 (Boundedness of iterates)**.** For the activations $\{\boldsymbol{X}_l\}$, boundedness follows automatically from the coercivity of Ψ: since the overall training objective $\mathcal{F} \geq 0$ and each update decreases $\mathcal{F}$, the sequence $\{\mathcal{F}^{(k)}\}$ is bounded, and coercivity of Ψ then prevents the activations from escaping to infinity. For the weights $\{\boldsymbol{W}_l\}$, the situation depends on G: if G is coercive (e.g. ℓ_1 or ℓ_2 regularization), the same argument applies and boundedness of the weights follows. If $G \equiv 0$, however, there is no such guarantee and boundedness of $\{\boldsymbol{W}_l^{(k)}\}$ must be assumed explicitly. Nonetheless, a ℓ_2-ball constraint for a sufficiently large radius could always be assumed without hampering the quality of the solution.

Remark 9 (Choosing values for μ_l)**.** If we define $\beta_l = \mu_l/\mu_{l-1}$, then update equation 15 is

$$\boldsymbol{X}_l^{(1)} \leftarrow \sigma\left(\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1} - \frac{\beta_l}{1+\beta_l L_\sigma^2\|\boldsymbol{W}_l\|^2}\boldsymbol{W}_l^\top\left[\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(0)}) - \boldsymbol{X}_{l+1}\right]\right), \tag{18}$$

then we take $\beta_l = \beta \in \mathbb{R}$. So, we only define two parameters for LIFTY, one is β and the other one is defined in a similar way for the update equation 34. In the practice we choose the value of β big (~ 100), but it is important to mention that LIFTY is not too sensitive to this parameter, and a value small for α (~ 0.01) to satisfy the hypothesis of Proposition 6. To recover each μ_l value we define $\nu_L = 1$ and then we calculate the values by definition of β and α.

Remark 10 (Comparison with FY-variants.)**.** Results of Table 3 show that the improvement cannot be attributed solely to the Fenchel-Young training formulation. Our proposed optimization algorithm (independently of the warm-start strategy) more effectively exploits the geometry induced by the lifting formulation than FY-SGD and FY-Adam. More specifically, while the (W)-update in equation 20 reduces to a standard gradient step when ($G = 0$) , the (X)-update in equation 14 does not. Instead, it performs a geometry-aware update that is tailored to the lifted optimization problem. These observations suggest that this additional refinement is the main reason for the observed performance gap between LIFTY and the baselines, despite all methods relying on the same underlying training formulation.

B.2 PRECONDITIONING

Instead a performing standard forward-backward steps for solving equation 11, one could instead consider a preconditioned variant as follows.

Proposition 5 (Preconditioned $\boldsymbol{W}_l$-update)**.** Let $H_l = \boldsymbol{X}_l\boldsymbol{X}_l^\top + \varepsilon I$ with $\varepsilon > 0$. Under Assumption 1, for any step-size $0 < \gamma \leq 1/L_\sigma$, the iterates defined by

$$\boldsymbol{W}_l^{(k+1)} \leftarrow \mathrm{prox}^{H_l}_{(\gamma/\mu_l)G}\Big(\boldsymbol{W}_l^{(k)} - \gamma\big[\sigma(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l) - \boldsymbol{X}_{l+1}\big]\boldsymbol{X}_l^\top H_l^{-1}\Big) \tag{19}$$

converge to the minimizer of equation 11.

Right-multiplying by H_l^{-1} whitens the input directions, which removes usual $\|\boldsymbol{X}_l\|^2$ dependence from the step-size bound. This is particularly useful in our setting, as the activations $\boldsymbol{X}_l$ are updated at each iteration, making a fixed step-size valid throughout training without requiring retuning. This preconditioning can be seen as the input factor of the Kronecker-factored Fisher approximation in KFAC Martens & Grosse (2015).

Note that, although the preconditioned scheme formally involves H_l^{-1}, we never explicitly need to compute this inverse in practice. Instead, the action of H_l^{-1} can be computed by solving the corresponding linear systems using a Cholesky factorization.

Here, the proximal operator $\mathrm{prox}^{H_l}_{(\gamma/\mu_l)G}$ is taken with respect to the geometry induced by H_l. In general, closed-form expressions are no longer available unless H_l has a simple structure. In particular, the ℓ_2^2 penalty still yields a closed-form linear rescaling. When $G = 0$, the update reduces to a plain preconditioned gradient step. In practice, diagonal or low-rank approximations of H_l can be used to recover simple closed-form expressions (e.g., soft-thresholding).

In practice, on the controlled experiments of Section 6.2, we have found that removing the preconditioning somewhat leads to a less stable optimization and slighlty lower final accuracy, while both variants eventually achieve similar performance on more challenging datasets, with preconditioning reaching high-performing regions substantially faster. These observations naturally raise whether adaptive preconditioning could combine the benefits of both variants.

C PROOFS

In this section, we detail the proofs of all statements made in the main paper.

C.1 FENCHEL-YOUNG TRAINING PROOFS

Lemma (Regularized prediction)**.** Under Assumption 1,

$$\boldsymbol{X}_{l+1} = \sigma(\boldsymbol{W}_l \boldsymbol{X}_l) = \arg\max_{\boldsymbol{X}} \langle \boldsymbol{X}, \boldsymbol{W}_l \boldsymbol{X}_l \rangle - \Psi(\boldsymbol{X}),$$

where Ψ is the coordinate-wise sum of ψ.

Proof. Direct from Assumption 1, expanding the squared norm and discarding constant terms. □

Lemma (Vanishing Fenchel-Young gap)**.** Let B_Ψ denotes Fenchel-Young gap associated to Ψ, i.e., $B_\Psi(\boldsymbol{U}, \boldsymbol{V}) \coloneqq \Psi^*(\boldsymbol{U}) + \Psi(\boldsymbol{V}) - \langle \boldsymbol{U}, \boldsymbol{V} \rangle \geq 0$. Then, under Assumption 1, we have

$$\boldsymbol{X}_{l+1} = \sigma(\boldsymbol{W}_l \boldsymbol{X}_l) \quad \Leftrightarrow \quad B_\Psi(\boldsymbol{W}_l \boldsymbol{X}_l, \boldsymbol{X}_{l+1}) = 0.$$

Proof. By the Fenchel–Young equality condition,

$$B_\Psi(\boldsymbol{U}, \boldsymbol{V}) = 0 \quad \Longleftrightarrow \quad \boldsymbol{V} \in \nabla\Psi^*(\boldsymbol{U}) \quad \Longleftrightarrow \quad \boldsymbol{U} \in \nabla\Psi(\boldsymbol{V}).$$

Moreover, the proximal representation in Assumption 1 is equivalent to $\sigma = \nabla\Psi^*$. Hence, with $\boldsymbol{U} = \boldsymbol{W}_l \boldsymbol{X}_l$ and $\boldsymbol{V} = \boldsymbol{X}_{l+1}$,

$$B_\Psi(\boldsymbol{W}_l \boldsymbol{X}_l, \boldsymbol{X}_{l+1}) = 0 \quad \Longleftrightarrow \quad \boldsymbol{X}_{l+1} = \nabla\Psi^*(\boldsymbol{W}_l \boldsymbol{X}_l) = \sigma(\boldsymbol{W}_l \boldsymbol{X}_l).$$

□

Proposition. Any collection of activation variables satisfying the forward-pass recursion constraints $\{X_l = \sigma(W_{l-1} X_{l-1})\}_{l=2}^{L-1}$ is stationary with respect to the activation variables of the Fenchel-Young lifted optimization problem in 10.

Proof. From Lemma 2, forward-pass is equivalent to $B_\Psi(W_{l-1} X_{l-1}, X_l) = 0$.

Consider now the lifted objective equation 10 and fix the weights. For any hidden activation X_l, $l = 2, \ldots, L-1$, the variable X_l appears only in the two consecutive Fenchel–Young gaps

$$B_\Psi(W_{l-1} X_{l-1}, X_l) \quad \text{and} \quad B_\Psi(W_l X_l, X_{l+1}).$$

Using

$$\nabla_2 B_\Psi(u, v) = \nabla\Psi(v) - u \quad \text{and} \quad \nabla_1 B_\Psi(u, v) = \nabla\Psi^*(u) - v,$$

we obtain

$$\nabla_{X_l} B_\Psi(W_{l-1} X_{l-1}, X_l) = \nabla\Psi(X_l) - W_{l-1} X_{l-1} = 0,$$

where the last equality follows from $X_l = \nabla\Psi^*(W_{l-1} X_{l-1})$, and similarly

$$\nabla_{X_l} B_\Psi(W_l X_l, X_{l+1}) = W_l^\top \big(\nabla\Psi^*(W_l X_l) - X_{l+1}\big) = 0,$$

because

$$X_{l+1} = \sigma(W_l X_l) = \nabla\Psi^*(W_l X_l).$$

Hence both contributions to the gradient of the lifted objective with respect to X_l vanish. Since the remaining terms of equation 10 do not depend on the hidden activations, we obtain

$$\nabla_{X_l} \mathcal{F} = 0, \qquad l = 2, \ldots, L.$$

Thus any collection of activation variables satisfying the forward-pass recursion constraints is stationary with respect to the activation variables of the Fenchel–Young lifted problem. □

C.2 UPDATING RULE OF WEIGHTS

Proposition ($\boldsymbol{W}_l$-update)**.** Under Assumption 1, for any step-size $0 < \gamma \leq 1/(L_\sigma \|\boldsymbol{X}_l\|^2)$, the iterates defined by

$$\boldsymbol{W}_l^{(k+1)} \leftarrow \mathrm{prox}_{(\gamma/\mu_l)G}\Big(\boldsymbol{W}_l^{(k)} - \gamma\big[\sigma(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l) - \boldsymbol{X}_{l+1}\big]\boldsymbol{X}_l^\top\Big) \tag{20}$$

converge to the minimizer of equation 11.

Proof. Let $\mathcal{F}_l(\boldsymbol{W}_l) = \mu_l B_\Psi(\boldsymbol{W}_l\boldsymbol{X}_l, \boldsymbol{X}_{l+1}) + G(\boldsymbol{W}_l)$. Since Ψ^* is L_σ-smooth and $\nabla\Psi^* = \sigma$, the function $\boldsymbol{W}_l \mapsto \Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l)$ has $L_\sigma\|\boldsymbol{X}_l\|^2$-Lipschitz gradient. Hence, for any $\boldsymbol{W}_l$,

$$\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l) \leq \Psi^*(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l) + \Big\langle \sigma(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l), (\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)})\boldsymbol{X}_l \Big\rangle + \frac{L_\sigma\|\boldsymbol{X}_l\|^2}{2}\|\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)}\|_F^2.$$

Discarding terms independent of $\boldsymbol{W}_l$ and defining $\mathbf{r}_l^{(k)} = \sigma(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l) - \boldsymbol{X}_{l+1}$, we obtain the majorization

$$\mathcal{F}_l(\boldsymbol{W}_l) \leq \mu_l \Big\langle \mathbf{r}_l^{(k)}, (\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)})\boldsymbol{X}_l \Big\rangle + G(\boldsymbol{W}_l) + \frac{\mu_l L_\sigma\|\boldsymbol{X}_l\|^2}{2}\|\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)}\|_F^2 + C.$$

Since $0 < \gamma \leq \frac{1}{L_\sigma\|\boldsymbol{X}_l\|^2}$, we have $\frac{L_\sigma\|\boldsymbol{X}_l\|^2}{2} \leq \frac{1}{2\gamma}$. Therefore, the right-hand side is majorized by

$$\mu_l \Big\langle \mathbf{r}_l^{(k)}\boldsymbol{X}_l^\top, \boldsymbol{W}_l - \boldsymbol{W}_l^{(k)} \Big\rangle + G(\boldsymbol{W}_l) + \frac{\mu_l}{2\gamma}\|\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)}\|_F^2 + C.$$

Completing the square shows that its minimizer is

$$\boldsymbol{W}_l^{(k+1)} = \mathrm{prox}_{(\gamma/\mu_l)G}\left(\boldsymbol{W}_l^{(k)} - \gamma\,\mathbf{r}_l^{(k)}\boldsymbol{X}_l^\top\right),$$

which is exactly equation 20.

Thus, equation 20 is the proximal-gradient step for the convex subproblem equation 11. Since the smooth part has $L_\sigma\|\boldsymbol{X}_l\|^2$-Lipschitz gradient and $\gamma \leq 1/(L_\sigma\|\boldsymbol{X}_l\|^2)$, the standard proximal-gradient convergence result guarantees that the iterates converge to a minimizer of equation 11. □

Proposition. Let $H_l = \boldsymbol{X}_l\boldsymbol{X}_l^\top + \varepsilon I$ with $\varepsilon > 0$. Under Assumption 1, for any step-size $0 < \gamma \leq 1/L_\sigma$, the iterates defined by

$$\boldsymbol{W}_l^{(k+1)} \leftarrow \mathrm{prox}^{H_l}_{(\gamma/\mu_l)G}\Big(\boldsymbol{W}_l^{(k)} - \gamma\big[\sigma(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l) - \boldsymbol{X}_{l+1}\big]\boldsymbol{X}_l^\top H_l^{-1}\Big) \tag{21}$$

converge to the minimizer of equation 11.

Proof. Let $H_l = \boldsymbol{X}_l\boldsymbol{X}_l^\top + \varepsilon I$. We equip the space of matrices with the metric

$$\langle A, B\rangle_{H_l} = \mathrm{tr}(AH_lB^\top), \qquad \|A\|_{H_l}^2 = \mathrm{tr}(AH_lA^\top) = \|A\boldsymbol{X}_l\|^2 + \varepsilon\|A\|_F^2. \tag{22}$$

since Ψ^* is L_σ-smooth and $\nabla\Psi^* = \sigma$, we have

$$\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l) \leq \Psi^*(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l) + \Big\langle \sigma(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l), (\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)})\boldsymbol{X}_l \Big\rangle + \frac{L_\sigma}{2}\|(\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)})\boldsymbol{X}_l\|^2. \tag{23}$$

Because $\|(\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)})\boldsymbol{X}_l\|^2 \leq \|\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)}\|_{H_l}^2$, we obtain the majorant

$$\mathcal{F}(\boldsymbol{W}_l) \leq \mu_l\Big\langle r_l^{(k)}, (\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)})\boldsymbol{X}_l \Big\rangle + G(\boldsymbol{W}_l) + \frac{\mu_l L_\sigma}{2}\|\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)}\|_{H_l}^2 + \text{const}, \tag{24}$$

with $r_l^{(k)} = \sigma(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l) - \boldsymbol{X}_{l+1}$. Using $\gamma \leq 1/L_\sigma$ we get $\frac{\mu_l L_\sigma}{2} \leq \frac{\mu_l}{2\gamma}$, hence

$$\mathcal{F}(\boldsymbol{W}_l) \leq \mu_l\Big\langle r_l^{(k)}, (\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)})\boldsymbol{X}_l \Big\rangle + G(\boldsymbol{W}_l) + \frac{\mu_l}{2\gamma}\|\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)}\|_{H_l}^2 + \text{const}. \tag{25}$$

Moreover,

$$\Big\langle r_l^{(k)}, (\boldsymbol{W}_l - \boldsymbol{W}_l^{(k)})\boldsymbol{X}_l \Big\rangle = \Big\langle r_l^{(k)}\boldsymbol{X}_l^\top H_l^{-1}, \boldsymbol{W}_l - \boldsymbol{W}_l^{(k)} \Big\rangle_{H_l}. \tag{26}$$

Thus $\boldsymbol{W}_l^{(k+1)}$ minimizes

$$G(\boldsymbol{W}_l) + \frac{\mu_l}{2\gamma}\left\|\boldsymbol{W}_l - \left(\boldsymbol{W}_l^{(k)} - \gamma r_l^{(k)} \boldsymbol{X}_l^\top H_l^{-1}\right)\right\|_{H_l}^2, \tag{27}$$

which is exactly the proximal update stated in the proposition. Since the update minimizes an upper bound of the objective, we have $\mathcal{F}(\boldsymbol{W}_l^{(k+1)}) \leq \mathcal{F}(\boldsymbol{W}_l^{(k)})$. Standard arguments then give convergence to the minimizer. □

C.3 Updating rule of activations

We present below several proofs concerning the update rule for $\boldsymbol{X}_l$.

C.3.1 Case $l \neq L$; same Fenchel-Young potential

In addition to the approach proposed in the main text, we first discuss alternative options, each with its own limitations, to clarify the rationale behind our final choice.

We recall the training objective

$$\underset{\{\boldsymbol{W}_l\}_{l=1}^L, \{\boldsymbol{X}_l\}_{l=2}^L}{\text{minimize}} \left[\mathcal{F}(\{\boldsymbol{W}_l\}, \{\boldsymbol{X}_l\}) \coloneqq \sum_{l=1}^{L} \left(\mu_l B_\Psi(\boldsymbol{W}_l \boldsymbol{X}_l, \boldsymbol{X}_{l+1}) + G_l(\boldsymbol{W}_l)\right)\right].$$

The terms in $\mathcal{F}$ involving $\boldsymbol{X}_l$ are

$$\mathcal{G}(\boldsymbol{X}_l) = \mu_{l-1}[\Psi(\boldsymbol{X}_l) - \langle \boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}, \boldsymbol{X}_l\rangle] + \mu_l[\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l) - \langle \boldsymbol{W}_l\boldsymbol{X}_l, \boldsymbol{X}_{l+1}\rangle]. \tag{28}$$

We start with the most naive, yet still interesting approach.

Proposition. Under Assumption 1, if $\mu_l/\mu_{l-1} < 1/(L_\sigma^2\|\boldsymbol{W}_l\|^2)$, the iterates defined by

$$\boldsymbol{X}_l^{(k+1)} \leftarrow \sigma\left(\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1} - \frac{\mu_l}{\mu_{l-1}}\boldsymbol{W}_l^\top\left(\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}) - \boldsymbol{X}_{l+1}\right)\right), \tag{29}$$

converge to the minimizer of equation 13.

Proof. Let us consider the first order stationary condition by differentiating $\mathcal{G}$ with respect to $\boldsymbol{X}_l$.

$$\begin{aligned} & 0 = \mu_{l-1}\left(\nabla\Psi(\boldsymbol{X}_l) - \boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}\right) + \mu_l \boldsymbol{W}_l^\top\left(\nabla\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l) - \boldsymbol{X}_{l+1}\right) \\ \Leftrightarrow \quad & 0 = \mu_{l-1}\left(\sigma^{-1}(\boldsymbol{X}_l) - \boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}\right) + \mu_l \boldsymbol{W}_l^\top\left(\sigma(\boldsymbol{W}_l\boldsymbol{X}_l) - \boldsymbol{X}_{l+1}\right) \end{aligned}$$

The update rule follows from a direct inversion of the last equation. To establish convergence, note that the update is a Picard fixed-point iteration whose Jacobian J satisfies

$$\|J\| < L_\sigma^2 \frac{\mu_l}{\mu_{l-1}} \cdot \|\boldsymbol{W}_l\|^2,$$

where L_σ is the Lipschitz constant of σ. By the Banach fixed-point theorem, the iteration contracts and converges to the unique fixed point whenever $\|J\| < 1$, which yields the stated condition. □

Lemma 3. Under the same assumptions of previous proposition the iterates define by equation 29 satisfies

$$\mathcal{G}(\boldsymbol{X}_l^{(k)}) - \mathcal{G}(\boldsymbol{X}_l^{(k+1)}) \geq \frac{1}{2}(\mu_{l-1} - \mu_l L_\sigma \|\boldsymbol{W}_l\|^2)\|\boldsymbol{X}_l^{(k+1)} - \boldsymbol{X}_l^{(k)}\|^2. \tag{30}$$

Proof. Let $g(\boldsymbol{X}_l) = \Psi(\boldsymbol{X}_l) - \frac{1}{2}\|\boldsymbol{X}_l\|^2$ and $f(\boldsymbol{X}_l) = \frac{1}{2}\|\boldsymbol{X}_l\|^2 - \langle \boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}, \boldsymbol{X}_l\rangle + \frac{\mu_l}{\mu_{l-1}}[\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l) - \langle \boldsymbol{W}_l\boldsymbol{X}_l, \boldsymbol{X}_{l+1}\rangle]$, then $\frac{1}{\mu_{l-1}}\mathcal{G}(\boldsymbol{X}_l) = f(x) + g(x)$ and update equation 29 is a Forward Backward step with stepsize 1. Moreover μ's condition of the proposition is sufficient to assure the convergence and a sufficient decrease property

$$\mathcal{G}(\boldsymbol{X}_l^{(k)}) - \mathcal{G}(\boldsymbol{X}_l^{(k+1)}) \geq \frac{1}{2}(\mu_{l-1} - \mu_l L_\sigma \|\boldsymbol{W}_l\|^2)\|\boldsymbol{X}_l^{(k+1)} - \boldsymbol{X}_l^{(k)}\|^2.$$

□

The update equation 29 combines a *forward* signal and a *backward* signal. The term $\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}$ carries the *forward* signal from layer $l-1$, recovering the standard forward pass once σ is applied. The correction $-\frac{\mu_l}{\mu_{l-1}}\boldsymbol{W}_l^\top(\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}) - \boldsymbol{X}_{l+1})$ carries the *backward* signal from layer $l+1$, propagating the local residual $\sigma(\boldsymbol{W}_l\boldsymbol{X}_l) - \boldsymbol{X}_{l+1}$. At a fixed point of equation 29, $\boldsymbol{X}_l$ satisfies both $\boldsymbol{X}_l = \sigma(\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1})$ and $\boldsymbol{X}_{l+1} = \sigma(\boldsymbol{W}_l\boldsymbol{X}_l)$, i.e., it is consistent with the global forward pass.

The ratio μ_l/μ_{l-1} controls the relative weight of these two signals. The convergence condition $\mu_l/\mu_{l-1} < 1/(L_\sigma^2\|\boldsymbol{W}_l\|^2)$ is restrictive in practice: it couples the penalty schedule to the spectral norm of the weights, which varies during training and is not controlled a priori. For wide or poorly normalized networks, $\|\boldsymbol{W}_l\|^2$ can be large, forcing $\mu_l \ll \mu_{l-1}$ and effectively suppressing the signal from the layer above.

In order to obtain more permissive convergence conditions, we now turn to majorization-minimization approaches. These methods construct surrogates that are tight at the current iterate $\boldsymbol{X}_l^{(k)}$ and upper-bound $\mathcal{G}$. By defining $\boldsymbol{X}_l^{(k+1)}$ as the minimizer of such a surrogate, we obtain the descent property $\mathcal{G}(\boldsymbol{X}_l^{(k+1)}) \leq \mathcal{G}(\boldsymbol{X}_l^{(k)})$.

The first approach we consider to build such a surrogate relies on the fact that Ψ^* has an L_σ-Lipschitz gradient. As a result, the term $\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l)$ admits the standard quadratic upper bound in Euclidean geometry.

Proposition. Under Assumption 1, let $\tau_l > 0$ satisfy $\tau_l \leq 1/(\mu_l L_\sigma\|\boldsymbol{W}_l\|^2)$. Then the iterates

$$\boldsymbol{X}_l^{(k+1)} \leftarrow \operatorname{prox}_{\tau_l\mu_{l-1}\Psi}\left(\boldsymbol{X}_l^{(k)} - \tau_l\left[\mu_l\boldsymbol{W}_l^\top(\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}) - \boldsymbol{X}_{l+1}) - \mu_{l-1}\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}\right]\right) \quad (31)$$

converge to the minimizer of equation 13.

Proof. Since Ψ^* is L_σ-smooth, we have

$$\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l) \leq \Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}) + \langle\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}), \boldsymbol{W}_l(\boldsymbol{X}_l - \boldsymbol{X}_l^{(k)})\rangle + \tfrac{L_\sigma}{2}\|\boldsymbol{W}_l(\boldsymbol{X}_l - \boldsymbol{X}_l^{(k)})\|^2,$$

where we have used the fact that $\nabla\Psi^* = \sigma$. Using $\|\boldsymbol{W}_l(\boldsymbol{X}_l - \boldsymbol{X}_l^{(k)})\|^2 \leq \|\boldsymbol{W}_l\|^2\|\boldsymbol{X}_l - \boldsymbol{X}_l^{(k)}\|^2$ and $\tau_l \leq 1/(\mu_l L_\sigma\|\boldsymbol{W}_l\|^2)$, we obtain the majorant

$$\begin{aligned}\mathcal{G}(\boldsymbol{X}_l) \leq\ & \mu_{l-1}\big[\Psi(\boldsymbol{X}_l) - \langle\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}, \boldsymbol{X}_l\rangle\big] \\ & + \mu_l\langle\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}), \boldsymbol{W}_l(\boldsymbol{X}_l - \boldsymbol{X}_l^{(k)})\rangle - \mu_l\langle\boldsymbol{X}_{l+1}, \boldsymbol{W}_l\boldsymbol{X}_l\rangle \\ & + \tfrac{1}{2\tau_l}\|\boldsymbol{X}_l - \boldsymbol{X}_l^{(k)}\|^2 + \text{const}.\end{aligned}$$

Using $\langle a, \boldsymbol{W}_l x\rangle = \langle\boldsymbol{W}_l^\top a, x\rangle$, the linear terms in $\boldsymbol{X}_l$ become

$$\left\langle\mu_l\boldsymbol{W}_l^\top(\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}) - \boldsymbol{X}_{l+1}) - \mu_{l-1}\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}, \boldsymbol{X}_l\right\rangle.$$

Thus $\boldsymbol{X}_l^{(k+1)}$ minimizes

$$\mu_{l-1}\Psi(\boldsymbol{X}_l) + \tfrac{1}{2\tau_l}\|\boldsymbol{X}_l - \boldsymbol{X}_l^{(k)}\|^2 + \langle c_l^{(k)}, \boldsymbol{X}_l\rangle,$$

with $c_l^{(k)} = \mu_l\boldsymbol{W}_l^\top(\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}) - \boldsymbol{X}_{l+1}) - \mu_{l-1}\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}$. Completing the square yields

$$\boldsymbol{X}_l^{(k+1)} = \arg\min_{\boldsymbol{X}_l}\left\{\mu_{l-1}\Psi(\boldsymbol{X}_l) + \tfrac{1}{2\tau_l}\left\|\boldsymbol{X}_l - \left(\boldsymbol{X}_l^{(k)} - \tau_l c_l^{(k)}\right)\right\|^2\right\},$$

which gives the proximal update. □

Unfortunately, the proximal operator $\operatorname{prox}_{\tau_l\mu l-1\Psi}$ does not admit a closed-form expression in general. To overcome this limitation, we replace the Euclidean quadratic majorant with a Bregman majorant induced by Ψ, which is better aligned with the geometry of the problem. More precisely, since Ψ is $(1/L_\sigma)$-strongly convex, its Bregman divergence dominates the Euclidean quadratic term.

Proposition. Under Assumption 1, the iterates $\{\boldsymbol{X}_l^{(k)}\}_{k\in\mathbb{N}}$ converge to the minimizer of equation 13

$$\boldsymbol{X}_l^{(k+1)} \leftarrow \sigma\left(\frac{\mu_{l-1}\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1} + \mu_l L_\sigma^2\|\boldsymbol{W}_l\|^2\sigma^{-1}(\boldsymbol{X}_l^{(k)}) - \mu_l\boldsymbol{W}_l^\top(\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}) - \boldsymbol{X}_{l+1})}{\mu_{l-1} + \mu_l L_\sigma^2\|\boldsymbol{W}_l\|^2}\right).$$

Proof. We start again using the fact Ψ^* is L_σ-smooth to build same majorizing upper bound as before, for a current iteration $\boldsymbol{X}_l^{(k)}$:

$$\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l) \leq \Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}) + \langle\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}), \boldsymbol{W}_l(\boldsymbol{X}_l - \boldsymbol{X}_l^{(k)})\rangle + \frac{L_\sigma\|\boldsymbol{W}_l\|^2}{2}\|\boldsymbol{X}_l - \boldsymbol{X}_l^{(k)}\|^2.$$

Since Ψ^* is L_σ-smooth, Ψ is $\frac{1}{L_\sigma}$-strongly convex, so

$$D_\Psi(\boldsymbol{X}_l, \boldsymbol{X}_l^{(k)}) \geq \frac{1}{2L_\sigma}\|\boldsymbol{X}_l - \boldsymbol{X}_l^{(k)}\|^2,$$

which gives $\frac{L_\sigma\|\boldsymbol{W}_l\|^2}{2}\|\boldsymbol{X}_l - \boldsymbol{X}_l^{(k)}\|^2 \leq L_\sigma^2\|\boldsymbol{W}_l\|^2 D_\Psi(\boldsymbol{X}_l, \boldsymbol{X}_l^{(k)})$. Replacing the Euclidean term by this Bregman term, which is geometrically consistent with the first term in $\mathcal{G}$, the majorizing upper bound becomes

$$\mathcal{G}(\boldsymbol{X}_l) \leq \mu_{l-1}[\Psi(\boldsymbol{X}_l) - \langle\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}, \boldsymbol{X}_l\rangle] + \mu_l\langle\boldsymbol{W}_l^\top(\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^{(k)}) - \boldsymbol{X}_{l+1}), \boldsymbol{X}_l\rangle \\ + \mu_l L_\sigma^2\|\boldsymbol{W}_l\|^2 D_\psi(\boldsymbol{X}_l, \boldsymbol{X}_l^{(k)}) + \text{const.}$$

Let define $u_k(\boldsymbol{X}_l)$ as the r.h.s., it is easy to prove that 1) $\mathcal{G}(\boldsymbol{X}_l^{(k)}) = u_k(\boldsymbol{X}_l^{(k)})$ and 2) $\nabla\mathcal{G}(\boldsymbol{X}_l^{(k)}) = \nabla u_k(\boldsymbol{X}_l^{(k)})$. Setting to zero the gradient with respect to $\boldsymbol{X}_l$ and using $\nabla_{\boldsymbol{X}_l} D_\Psi(\boldsymbol{X}_l, \boldsymbol{X}_l^t) = \nabla\Psi(\boldsymbol{X}_l) - \nabla\Psi(\boldsymbol{X}_l^t)$:

$$\mu_{l-1}(\nabla\Psi(\boldsymbol{X}_l) - \boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}) + \mu_l\boldsymbol{W}_l^\top(\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^t) - \boldsymbol{X}_{l+1}) \\ + \mu_l L_\sigma^2\|\boldsymbol{W}_l\|^2(\nabla\Psi(\boldsymbol{X}_l) - \nabla\Psi(\boldsymbol{X}_l^t)) = 0.$$

Passing to dual coordinates $\boldsymbol{S}_l = \nabla\Psi(\boldsymbol{X}_l)$, this is linear in $\boldsymbol{S}_l$:

$$\left(\mu_{l-1} + \mu_l L_\sigma^2\|\boldsymbol{W}_l\|^2\right)\boldsymbol{S}_l = \mu_{l-1}\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1} + \mu_l L_\sigma^2\|\boldsymbol{W}_l\|^2\nabla\Psi(\boldsymbol{X}_l^t) \\ - \mu_l\boldsymbol{W}_l^\top(\sigma(\boldsymbol{W}_l\boldsymbol{X}_l^t) - \boldsymbol{X}_{l+1}),$$

which yields the proposed update equation 14 after dividing by $(\mu_{l-1} + \mu_l L_\sigma^2\|\boldsymbol{W}_l\|^2)$ and applying the inverse map $\boldsymbol{X}_l = (\nabla\Psi)^{-1}(\boldsymbol{S}_l) = \sigma(\boldsymbol{S}_l)$.

Since u_k is strongly convex, $X_l^{(k+1)}$ is the exact minimizer for it. So $\mathcal{G}(\boldsymbol{X}_l^{(k+1)}) \leq \mathcal{G}(\boldsymbol{X}_l^{(k)})$, moreover

$$\mathcal{G}(\boldsymbol{X}_l^{(k)}) - \mathcal{G}(\boldsymbol{X}_l^{(k+1)}) \geq \frac{\mu_{l-1} + \mu_l L_\sigma^2\|\boldsymbol{W}_l\|^2}{2L_\sigma}\|\boldsymbol{X}_l^{(k+1)} - \boldsymbol{X}_l^{(k)}\|^2. \tag{32}$$

For the Majorizing Minimizing argument the iteration converges to the unique minimizer of $\mathcal{G}$. □

Compared to equation 29, the update equation 14 preserves the same structure while removing the restrictive condition on the ratio μ_l/μ_{l-1}. The additional term involving $\sigma^{-1}(\boldsymbol{X}_l^{(k)}) \equiv \nabla\Psi(\boldsymbol{X}_l^{(k)})$ acts as a mirror proximal correction: instead of controlling the smooth term $\Psi^*(\boldsymbol{W}_l\boldsymbol{X}_l)$ through an explicit Euclidean step-size restriction, the update stabilizes the iteration directly in the geometry induced by Ψ.

As a result we have two interesting properties. First the influence of the upper layer is no longer limited by the spectral norm of $\boldsymbol{W}_l$, and the penalty parameters μ_l can be chosen based on modeling considerations rather than convergence constraints. In particular, this allows to implement the coupling strategy discussed in Remark 9 to yield provable convergence guarantees. Second, the update is interpretable and explicitly involves the activation function itself.

C.3.2 Case $l = L$; different Fenchel-Young potentials

For the case $l = L$ we study the optimization problem

$$\underset{\boldsymbol{X}_L}{\text{minimize}}\ \mu_{L-1}\left(\Psi(\boldsymbol{X}_L) - \langle\boldsymbol{W}_{L-1}\boldsymbol{X}_{L-1}, \boldsymbol{X}_L\rangle\right) + \left(\Phi^*(\boldsymbol{W}_L\boldsymbol{X}_L) - \langle\boldsymbol{W}_L\boldsymbol{X}_L, \boldsymbol{X}_{L+1}\rangle\right). \tag{33}$$

The fact of mix functions Ψ and Φ does not allow us to extend in a easy way Proposition 3, devoted to solve 33 we devise the following update rule.

Proposition 6. Under Assumption 1, if $1/\mu_{L-1} < 1/\left(L_\sigma L_{\nabla\Phi^*}\|\boldsymbol{W}_L\|^2\right)$, the iterates defined by

$$\boldsymbol{X}_L^{(k+1)} \leftarrow \sigma\left(\boldsymbol{W}_{L-1}\boldsymbol{X}_{L-1} - \frac{1}{\mu_{L-1}}\boldsymbol{W}_L^\top\left(\nabla\Phi^*(\boldsymbol{W}_L\boldsymbol{X}_L^{(k)}) - \boldsymbol{X}_{L+1}\right)\right), \tag{34}$$

converge to the minimizer of equation 33. Moreover, writing $\mathcal{G}_L$ for the objective in equation 33, the same mirror-descent estimate gives

$$\mathcal{G}_L(\boldsymbol{X}_L^{(k)}) - \mathcal{G}_L(\boldsymbol{X}_L^{(k+1)}) \geq \frac{1}{2}\left(\frac{\mu_{L-1}}{L_\sigma} - L_{\nabla\Phi^*}\|\boldsymbol{W}_L\|_2^2\right)\|\boldsymbol{X}_L^{(k+1)} - \boldsymbol{X}_L^{(k)}\|_F^2.$$

Indeed, $\Phi^*(\boldsymbol{W}_L\cdot)$ has gradient-Lipschitz constant $L_{\nabla\Phi^*}\|\boldsymbol{W}_L\|_2^2$, while the Ψ term is $1/L_\sigma$-strongly convex; applying the standard three-point inequality for the mirror step yields the displayed bound.

C.4 Full-batch convergence

Lemma 4. Let $\{a_k\}$ and $\{\varepsilon_k\}$ be non-negative sequences, such that $\sum_{k=0}^{\infty}\varepsilon_k < \infty$ and $a_{k+1} \leq a_k + \varepsilon_k$. Then $\{a_k\}$ converges.

Proof. Let S_k be the partial sum for ε_k and $b_k = a_k - S_{k-1}$. Notice that b_k is bounded from below. Now we show that is a non-increasing sequence

$$\begin{aligned} b_{k+1} &= a_{k+1} - S_k \\ &\leq (a_k + \varepsilon_k) - (S_{k-1} + \varepsilon_k) \\ &= a_k - S_{k-1} = b_k. \end{aligned}$$

In consequence, b_k converges. Finally, from $a_k = b_k + S_{k-1}$, the fact that b_k converges, and S_k converges, we conclude the result. □

We are now ready to prove Theorem 1 that we recall below.

Theorem. *Under Assumptions 1 and 2, with fixed $\mu_l > 0$ and $G_l \equiv 0$, suppose the sequence generated by Algorithm 1 is bounded. Assume that the weight step sizes satisfy $1/\gamma_l^{(k)} - (L_\sigma/2)\|\boldsymbol{X}_l^{(k+1)}\|_F^2 \geq c_w > 0$ for all l, k, and that $(\mu_{L-1}/L_\sigma) - L_{\nabla\Phi^*}\|\boldsymbol{W}_L^{(k)}\|_2^2 \geq 2c_x > 0$ for all k. Then every limit point is a stationary point of the lifted objective equation 10.*

Proof. Since the sequence is bounded, there exist finite constants $M_{X,l}$ and $M_{W,L}$ such that $\|\boldsymbol{X}_l^{(k)}\|_F \leq M_{X,l}$ and $\|\boldsymbol{W}_L^{(k)}\|_2 \leq M_{W,L}$ for all k. We also set $\lambda_l := \mu_l$ for $l < L$ and $\lambda_L := 1$. These bounds ensure that all block-gradient Lipschitz constants remain finite.

Step 1 – Forward-pass initialization. For fixed weights $\{\boldsymbol{W}_l^{(k)}\}$, the warm-start initializes activations sequentially as $\boldsymbol{X}_l^{(k+1/2)} := \sigma(\boldsymbol{W}_{l-1}^{(k)}\boldsymbol{X}_{l-1}^{(k+1/2)})$ for $l = 2, \ldots, L$, with $\boldsymbol{X}_1^{(k+1/2)} := X$. By construction all intermediate Fenchel-Young gaps vanish, so

$$\mathcal{F}(\{\boldsymbol{W}_l^{(k)}\}, \{\boldsymbol{X}_l^{(k+1/2)}\}) = B_\Phi(\boldsymbol{W}_L^{(k)}\boldsymbol{X}_L^{(k+1/2)}, \boldsymbol{X}_{L+1}), \qquad \text{because } G_l \equiv 0.$$

By Assumption 2 applied to $(\{\boldsymbol{W}_l^{(k)}\}, \{\boldsymbol{X}_l^{(k)}\})$, the remaining gap is bounded by the full sum of gaps at the previous iterate, giving

$$\mathcal{F}(\{\boldsymbol{W}_l^{(k)}\}, \{\boldsymbol{X}_l^{(k+1/2)}\}) \leq \mathcal{F}^{(k)} + \varepsilon_k, \tag{i}$$

where $\mathcal{F}^{(k)} \coloneqq \mathcal{F}(\{\boldsymbol{W}_l^{(k)}\}, \{\boldsymbol{X}_l^{(k)}\})$.

Step 2 – Correction step decreases the objective. Let $\boldsymbol{X}_l^{(k+1)}$ be the result after applying the activation updates sequentially. For the hidden activations, equation equation 32 in the proof of Proposition 3 gives the decrease; for the last activation, we use the mirror-descent estimate in Proposition 6. Thus,

$$\mathcal{F}(\{\boldsymbol{W}_l^{(k)}\}, \{\boldsymbol{X}_l^{(k+1)}\}) \leq \mathcal{F}(\{\boldsymbol{W}_l^{(k)}\}, \{\boldsymbol{X}_l^{(k+1/2)}\}) - C_{\boldsymbol{X}_l}^{(k)}\sum_l \|\boldsymbol{X}_l^{(k+1)} - \boldsymbol{X}_l^{(k+1/2)}\|^2, \tag{ii}$$

where

$$C_{\boldsymbol{X}_l}^{(k)} = \begin{cases} \frac{1}{2L_\sigma}(\mu_{l-1} + \mu_l L_\sigma^2 \|\boldsymbol{W}_l^{(k)}\|^2), & 2 \le l \le L-1 \\ \frac{1}{2}\left(\frac{\mu_{L-1}}{L_\sigma} - L_{\nabla\Phi^*}\|\boldsymbol{W}_L^{(k)}\|_2^2\right), & l = L. \end{cases}$$

Under the bounds in Theorem 1, these coefficients satisfy $C_{\boldsymbol{X}_l}^{(k)} \ge c_x$ for every activation block and every k.

Step 3 – Weight update gives sufficient decrease. By Proposition 2, the composition for block l has Lipschitz gradient constant $L_\sigma \|\boldsymbol{X}_l^{(k+1)}\|_F^2$. Define $\lambda_l := \mu_l$ for $l < L$ and $\lambda_L := 1$. Since the lifted objective weights the hidden gaps by μ_l, each weight update satisfies

$$\mathcal{F}(\{\boldsymbol{W}_l^{(k+1)}\}, \{\boldsymbol{X}_l^{(k+1)}\}) \le \mathcal{F}(\{\boldsymbol{W}_l^{(k)}\}, \{\boldsymbol{X}_l^{(k+1)}\}) - \sum_l a_{w,l}^{(k)} \|\boldsymbol{W}_l^{(k+1)} - \boldsymbol{W}_l^{(k)}\|^2, \qquad \text{(iii)}$$

where

$$a_{w,l}^{(k)} := \lambda_l \left(\frac{1}{\gamma_l^{(k)}} - \frac{L_\sigma}{2}\|\boldsymbol{X}_l^{(k+1)}\|_F^2\right).$$

Chaining inequalities. Combining (i), (ii), (iii), and denoting $\mathcal{F}^{(k+1)} := \mathcal{F}(\{\boldsymbol{W}_l^{(k+1)}\}, \{\boldsymbol{X}_l^{(k+1)}\})$, we obtain

$$\mathcal{F}^{(k+1)} \overset{\text{(iii)}}{\le} \mathcal{F}(\{\boldsymbol{W}_l^{(k)}\}, \{\boldsymbol{X}_l^{(k+1)}\}) \overset{\text{(ii)}}{\le} \mathcal{F}(\{\boldsymbol{W}_l^{(k)}\}, \{\boldsymbol{X}_l^{(k+1/2)}\}) \overset{\text{(i)}}{\le} \mathcal{F}^{(k)} + \varepsilon_k. \qquad (35)$$

where the first inequality follows from the sufficient decrease of the weight update, the second from lifting (ii) from $\mathcal{G}$ to $\mathcal{F}$ as the remaining terms are unchanged by the activation update, and the third from Assumption 2.

Convergence of the objective. Applying Lemma 4 to equation 35 we have that $\mathcal{F}^{(k)}$ convergences. Moreover,

$$\mathcal{F}^{(k+1)} \le \mathcal{F}^{(k)} + \varepsilon_k - \sum_l a_{w,l}^{(k)} \|\boldsymbol{W}_l^{(k+1)} - \boldsymbol{W}_l^{(k)}\|^2 - C_{\boldsymbol{X}_l}^{(k)} \sum_l \|\boldsymbol{X}_l^{(k+1)} - \boldsymbol{X}_l^{(k+1/2)}\|^2. \qquad \text{(iv)}$$

Summing (iv) over k yields

$$\sum_{k=0}^{\infty} a_{w,l}^{(k)} \|\boldsymbol{W}_l^{(k+1)} - \boldsymbol{W}_l^{(k)}\|^2 < \infty \quad \text{and} \quad \sum_{k=0}^{\infty} C_{\boldsymbol{X}_l}^{(k)} \|\boldsymbol{X}_l^{(k+1)} - \boldsymbol{X}_l^{(k+1/2)}\|^2 < \infty,$$

The theorem's weight-step condition gives $a_{w,l}^{(k)} \ge \lambda_l c_w$. For the activation blocks, define $\underline{c}_x := \min\{c_x, \min_{2\le l\le L-1} \mu_{l-1}/(2L_\sigma)\} > 0$; then $C_{\boldsymbol{X}_l}^{(k)} \ge \underline{c}_x$. Therefore the corresponding unweighted sums are finite, and hence $\|\boldsymbol{W}_l^{(k+1)} - \boldsymbol{W}_l^{(k)}\| \to 0$ and $\|\boldsymbol{X}_l^{(k+1)} - \boldsymbol{X}_l^{(k+1/2)}\| \to 0$. Along every convergent subsequence, continuity of the forward pass then gives $\boldsymbol{X}_{l+1}^* = \sigma(\boldsymbol{W}_l^* \boldsymbol{X}_l^*)$ for $l = 1, \dots, L-1$.

Stationarity of limit points. Let $(\{\boldsymbol{W}_l^*\}, \{\boldsymbol{X}_l^*\})$ be any limit point. By the forward-consistency conclusion above,

$$\boldsymbol{X}_{l+1}^* = \sigma(\boldsymbol{W}_l^* \boldsymbol{X}_l^*) \qquad (l = 1, \dots, L-1).$$

For $G_l \equiv 0$, the vanishing weight updates give the fixed-point conditions

$$0 = \big[\sigma(\boldsymbol{W}_l^* \boldsymbol{X}_l^*) - \boldsymbol{X}_{l+1}^*\big](\boldsymbol{X}_l^*)^\top \quad (l = 1, \dots, L-1),$$

which hold by forward consistency, and

$$0 = \nabla_{\boldsymbol{W}_L} B_\Phi(\boldsymbol{W}_L^* \boldsymbol{X}_L^*, \boldsymbol{X}_{L+1}).$$

For the activation variables $\boldsymbol{X}_l$ with $2 \le l \le L-1$, forward consistency makes both adjacent Fenchel–Young residuals zero, so their activation stationarity conditions hold. The vanishing correction of the last activation update gives

$$0 = \mu_{L-1}\big(\nabla\Psi(\boldsymbol{X}_L^*) - \boldsymbol{W}_{L-1}^* \boldsymbol{X}_{L-1}^*\big) + (\boldsymbol{W}_L^*)^\top\big(\nabla\Phi^*(\boldsymbol{W}_L^* \boldsymbol{X}_L^*) - \boldsymbol{X}_{L+1}\big).$$

The first term is zero by forward consistency, hence $(\boldsymbol{W}_L^*)^\top(\nabla\Phi^*(\boldsymbol{W}_L^* \boldsymbol{X}_L^*) - \boldsymbol{X}_{L+1}) = 0$. Together with the weight condition for $\boldsymbol{W}_L$, these are the remaining stationarity conditions of equation 10.

Full sequence convergence. We do not claim whole-sequence convergence here; proving it would require a relative-error estimate for the warm-start reset in addition to the sufficient-decrease estimates above. □

Corollary 1 (Vanishing activation residual)**.** Under Assumption 1, with fixed $\mu_l > 0$, $G_l \equiv 0$, the forward-pass assumption, and the uniform sufficient-decrease bounds of Theorem 1, and boundedness of the iterates $\{(\{\boldsymbol{W}_l^{(k)}\}, \{\boldsymbol{X}_l^{(k)}\})\}_k$, the sequence of activation corrections generated by Algorithm 1 satisfies

$$\sum_{k=0}^{\infty} \|\boldsymbol{X}_l^{(k+1)} - \boldsymbol{X}_l^{(k+1/2)}\|^2 < \infty,$$

and therefore $\|\boldsymbol{X}_l^{(k+1)} - \boldsymbol{X}_l^{(k+1/2)}\| \to 0$ and $B_\Psi(\boldsymbol{W}_l^{(k)}\boldsymbol{X}_l^{(k)}, \boldsymbol{X}_{l+1}^{(k)}) \to 0$ for $l = 1, \dots, L-1$.

Proof. Direct from previous Theorem's proof. □

The equivalence between the stationary points of the lifted and original problem is proved below.

Theorem (Equivalence with original problem)**.** *Under the assumptions of Theorem 1, every limit point generated by Algorithm 1 $(\{\boldsymbol{W}_l^*\}, \{\boldsymbol{X}_l^*\})$ satisfies $\boldsymbol{X}_{l+1}^* = \sigma(\boldsymbol{W}_l^*\boldsymbol{X}_l^*)$ for all $l \in \{1, \dots, L-1\}$, and $\{\boldsymbol{W}_l^*\}$ is a stationary point of the original unregularized objective equation 1.*

Proof. Let $(\{\boldsymbol{W}_l^*\}, \{\boldsymbol{X}_l^*\})$ be a limit point generated by Algorithm 1. By the vanishing activation corrections proved above and continuity of σ,

$$\boldsymbol{X}_{l+1}^* = \sigma(\boldsymbol{W}_l^*\boldsymbol{X}_l^*) \qquad (l = 1, \dots, L-1).$$

Thus the hidden Fenchel–Young gaps vanish at the limit point.

For the hidden weight blocks, $G_l \equiv 0$ and the weight fixed-point conditions are

$$0 = \mu_l\big(\sigma(\boldsymbol{W}_l^*\boldsymbol{X}_l^*) - \boldsymbol{X}_{l+1}^*\big)(\boldsymbol{X}_l^*)^\top, \qquad l = 1, \dots, L-1,$$

which hold by forward consistency. For the last weight block, the fixed-point condition is

$$0 = \nabla_{\boldsymbol{W}_L} B_\Phi(\boldsymbol{W}_L^*\boldsymbol{X}_L^*, \boldsymbol{X}_{L+1}).$$

Writing $r_L^* := \nabla\Phi^*(\boldsymbol{W}_L^*\boldsymbol{X}_L^*) - \boldsymbol{X}_{L+1}$, this condition is $r_L^*(\boldsymbol{X}_L^*)^\top = 0$.

The activation stationarity conditions for $\boldsymbol{X}_l$, $2 \le l \le L-1$, hold because both adjacent hidden residuals vanish. The fixed-point condition of the last activation update is

$$0 = \mu_{L-1}\big(\nabla\Psi(\boldsymbol{X}_L^*) - \boldsymbol{W}_{L-1}^*\boldsymbol{X}_{L-1}^*\big) + (\boldsymbol{W}_L^*)^\top r_L^*.$$

The first term is zero by forward consistency, so $(\boldsymbol{W}_L^*)^\top r_L^* = 0$. This is precisely the first back-propagated stationarity condition at the output. Consequently, the chain-rule gradients (or the corresponding generalized chain-rule conditions for nonsmooth monotone σ) of the original objective with respect to all hidden weights vanish as well, since every such condition contains the factor $(\boldsymbol{W}_L^*)^\top r_L^*$. Together with $r_L^*(\boldsymbol{X}_L^*)^\top = 0$, this proves stationarity of the original unregularized objective equation 1.

□

Proof of Theorem 3. Write $W^k = (\boldsymbol{W}_l^{(k)})_{l=1}^L$, $X^k = (\boldsymbol{X}_l^{(k)})_{l=2}^L$, and $z^k = (W^k, X^k)$. Let $p^k = P(W^k)$ denote the forward-pass activations, with $p_1^k = X_1$ and $p_{l+1}^k = \sigma(W_l^k p_l^k)$ for $l < L$. All norms on collections of blocks are product Frobenius norms. Set

$$q^k = X^{k+1} - p^k, \qquad w^k = W^{k+1} - W^k, \qquad d_k^2 = \|q^k\|^2 + \|w^k\|^2,$$

and put $q_1^k = 0$ for the fixed input.

Sufficient decrease. The block estimates in the proof of Theorem 1 give a constant $a > 0$ such that

$$\mathcal{F}(z^k) + \varepsilon_k - \mathcal{F}(z^{k+1}) \ge a d_k^2. \tag{36}$$

For the weight blocks, their coefficients are $\lambda_l(1/\gamma_l^{(k)} - (\ell_l/2)\|X_l^{k+1}\|_F^2)$, where $\lambda_l = \mu_l$, $\ell_l = L_\sigma$ for $l < L$, and $\lambda_L = 1$, $\ell_L = L_{\nabla\Phi^*}$. The activation coefficients are bounded below by $\mu_{l-1}/(2L_\sigma)$ for $l < L$ and by c_x for $l = L$. Thus the assumed uniform margins give the required $a > 0$. Since ε_k eventually vanishes and $\mathcal{F} \ge 0$, $\mathcal{F}(z^k)$ eventually decreases to a finite value f_*, and $\sum_k d_k^2 < \infty$.

A gradient bound from the updates. Put $\beta_l^k = \mu_l L_\sigma^2 \|W_l^k\|_2^2$. In the prescribed reverse sweep, the lower neighbor of block l is still p_{l-1}^k, whereas its upper neighbor has already been updated. For $2 \le l < L$, the activation update and the forward-pass identities therefore give

$$\nabla\Psi(X_l^{k+1}) - \nabla\Psi(p_l^k) = \frac{\mu_l}{\mu_{l-1} + \beta_l^k}(W_l^k)^\top q_{l+1}^k. \tag{37}$$

Here $\nabla\Psi(p_l^k) = W_{l-1}^k p_{l-1}^k$. The inverse-free formula agrees with the Bregman block update because this lower neighbor has not yet changed.

At (W^k, X^{k+1}), the hidden activation gradient is

$$\begin{aligned}\nabla_{X_l}\mathcal{F} = \mu_{l-1}\big[\nabla\Psi(X_l^{k+1}) - W_{l-1}^k X_{l-1}^{k+1}\big] \\ + \mu_l (W_l^k)^\top\big[\sigma(W_l^k X_l^{k+1}) - X_{l+1}^{k+1}\big].\end{aligned}$$

Using equation 37, forward consistency of p^k, boundedness of the weights, and Lipschitz continuity of σ bounds its norm by $C\|q^k\|$, uniformly in k. For the last activation, the update gives instead

$$\begin{aligned}\nabla_{X_L}\mathcal{F}(W^k, X^{k+1}) = -\mu_{L-1} W_{L-1}^k q_{L-1}^k \\ + (W_L^k)^\top\big[\nabla\Phi^*(W_L^k X_L^{k+1}) - \nabla\Phi^*(W_L^k p_L^k)\big],\end{aligned}$$

which has the same bound. At fixed X^{k+1}, the activation gradients depend Lipschitz continuously on the weights over the bounded sets under consideration. Hence

$$\|\nabla_X \mathcal{F}(z^{k+1})\| \le C\big(\|q^k\| + \|w^k\|\big).$$

For the weights, since $G_l \equiv 0$, their updates imply

$$\nabla_{W_l}\mathcal{F}(W^k, X^{k+1}) = -\frac{\lambda_l}{\gamma_l^{(k)}} w_l^k.$$

Each weight-block gradient is Lipschitz with constant at most $\lambda_l \ell_l \|X_l^{k+1}\|_F^2$. Boundedness and $\inf_{l,k} \gamma_l^{(k)} > 0$ therefore yield $\|\nabla_W \mathcal{F}(z^{k+1})\| \le C\|w^k\|$. Combining these bounds gives a uniform $b > 0$ such that

$$\|\nabla\mathcal{F}(z^{k+1})\| \le b d_k. \tag{38}$$

The gradients used here are continuous near the cluster set. Indeed, boundedness and the explicit update formulas bound the arguments passed to $\sigma = \nabla\Psi^*$, both in the forward pass and in the activation updates. Their images lie in a compact subset of $\operatorname{int}\operatorname{dom}\Psi$ by the Legendre property. Thus $\mathcal{F}$ is continuously differentiable there. No Lipschitz assumption on $\nabla\Psi$ is required.

The KL argument. Let Ω be the cluster set of z^k. It is nonempty and compact, $\operatorname{dist}(z^k, \Omega) \to 0$, and continuity gives $\mathcal{F} = f_*$ on Ω. Moreover, equation 38 and $d_k \to 0$ imply stationarity on Ω. Uniformizing the KL inequality on this compact set gives a concave desingularizing function φ, with $\varphi(0) = 0$, such that, for all sufficiently large k with $r_k = \mathcal{F}(z^k) - f_* > 0$,

$$\varphi'(r_k)\|\nabla\mathcal{F}(z^k)\| \ge 1.$$

Take these indices also large enough that $\varepsilon_k = 0$. If $r_k = 0$ at any such index, equation 36 forces all subsequent d_k to vanish. Otherwise, concavity and equation 36–equation 38 give

$$\varphi(r_k) - \varphi(r_{k+1}) \ge \varphi'(r_k)(r_k - r_{k+1}) \ge \frac{a d_k^2}{b d_{k-1}}.$$

The denominator is positive by the KL inequality and equation 38. Young's inequality consequently yields

$$2d_k \le d_{k-1} + \frac{b}{a}\big[\varphi(r_k) - \varphi(r_{k+1})\big].$$

Summing from a sufficiently large K to N gives $\sum_{k=K}^N d_k \le d_{K-1} + (b/a)\varphi(r_K)$. Thus $\sum_k d_k < \infty$.

Finite length and convergence. The forward map P is Lipschitz on bounded weight sets. For $k \geq 1$,

$$X^{k+1} - X^k = P(W^k) - P(W^{k-1}) + q^k - q^{k-1}.$$

It follows that $\|z^{k+1} - z^k\| \leq C(d_k + d_{k-1})$ and hence

$$\sum_{k=0}^{\infty} \|z^{k+1} - z^k\| < \infty.$$

Therefore z^k converges to a single point z^*. Equation equation 38 and continuity imply $\nabla \mathcal{F}(z^*) = 0$. Moreover, $q^k \to 0$ gives $X^* = P(W^*)$, and Theorem 2 yields stationarity of W^* for equation 1. □

C.5 MINI-BATCH CONVERGENCE

For the stochastic mini-batch variant, we use a conditional stochastic-gradient assumption.

Assumption 3. Let $\mathbb{E}_k[\cdot]$ denote conditional expectation with respect to the mini-batch sampled at iteration k. For each weight block, let $g_l^{(k)}$ be the stochastic gradient used in the weight update and let $\nabla_l^{(k)}$ denote the corresponding full-batch gradient. Assume

$$\mathbb{E}_k[g_l^{(k)}] = \nabla_l^{(k)}, \qquad \mathbb{E}_k\left[\|g_l^{(k)} - \nabla_l^{(k)}\|_F^2\right] \leq \nu_l^2.$$

Assume also that the block gradients are uniformly $\bar{L}_w$-Lipschitz along the iterates, and that the conditional analogue of Assumption 2 holds with a nonnegative summable error sequence $\{\varepsilon_k\}$.

Theorem 4 (Mini-batch convergence)**.** *Under Assumptions 1 and 3, with fixed $\mu_l > 0$ and $G_l \equiv 0$, suppose the iterates generated by the mini-batch variant of Algorithm 1 are bounded. Let $\bar{L}_w$ be the uniform block-gradient Lipschitz constant and choose a constant step size $0 < \gamma < 2/\bar{L}_w$. Define $\lambda_l := \mu_l$ for $l < L$ and $\lambda_L := 1$, and set $\nu_\lambda^2 := \sum_l \lambda_l \nu_l^2$. Then, with $E_\varepsilon := \sum_{k=0}^{\infty} \varepsilon_k$, the iterates satisfy*

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\left[\sum_l \lambda_l \|\nabla_l^{(k)}\|_F^2\right] \leq \frac{\mathcal{F}^{(0)} + E_\varepsilon}{K\gamma(1 - \gamma\bar{L}_w/2)} + \frac{\gamma \bar{L}_w \nu_\lambda^2}{2(1 - \gamma\bar{L}_w/2)}.$$

Thus, for fixed γ, the result is an expected finite-step bound with a stochastic noise floor of order $\mathcal{O}(\gamma\nu_\lambda^2)$. It does not assert almost-sure convergence to a stationary point.

Proof. By the standard smoothness estimate for each weight block and the conditional unbiasedness and variance bounds in Assumption 3, the stochastic weight update gives

$$\mathbb{E}_k[\mathcal{F}^{(k+1)}] \leq \mathcal{F}^{(k)} - \gamma\left(1 - \frac{\gamma\bar{L}_w}{2}\right)\sum_l \lambda_l \|\nabla_l^{(k)}\|_F^2 + \frac{\gamma^2 \bar{L}_w}{2}\nu_\lambda^2 + \varepsilon_k. \tag{iii′}$$

The conditional forward-pass and activation-update estimates are included in the error term ε_k. Taking total expectations, summing over $k = 0, \ldots, K-1$, and using $\mathcal{F} \geq 0$ yields

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\left[\sum_l \lambda_l \|\nabla_l^{(k)}\|_F^2\right] \leq \frac{\mathcal{F}^{(0)} + E_\varepsilon}{K\gamma(1 - \gamma\bar{L}_w/2)} + \frac{\gamma \bar{L}_w \nu_\lambda^2}{2(1 - \gamma\bar{L}_w/2)}.$$

No almost-sure stationarity conclusion follows from this finite-step estimate. □

D EXPERIMENTAL SETTINGS

This section provides the implementation and training details required to reproduce the experiments presented in Section 6.2. All reported results are averaged over five runs using random seeds ranging from 0 to 4. Variability across runs is quantified using the standard error of the mean, reported either as shaded confidence envelopes in the figures or after the symbol $\pm$ in the tables. If not mentioned otherwise, we use the parameter $\mu_l = 1$ for every $l \in \{1, \ldots, L\}$. Test results are reported with an early stopping done on the validation accuracy. Experiments are conducted on a 12th Gen Intel(R) Core(TM) i9-12900H CPU. Runtimes are reported in Table 4 for the Covertype dataset. The same trends are observed over all settings.

Table 4: Average runtime per iteration on a 12th Gen Intel(R) Core(TM) i9-12900H CPU.

Method	Time per iteration (ms)
LIFTY (ours)	2.01 ± 0.00
FY-Adam	2.48 ± 0.05
FY-SGD	$\mathbf{1.93 \pm 0.12}$
(Gu et al., 2020)	29.95 ± 0.12
(Wang & Benning, 2023)	9.04 ± 0.02

We begin by describing the setups used in the controlled synthetic setting based on the two-moons classification problem

Two-moons architecture study. We first evaluate the sensitivity of the training dynamics to the hidden-layer geometry of the architecture. We generate 600 samples with noise level 0.2 and keep the generated dataset fixed across runs. For each seed, the data are randomly split into equally sized training and validation sets. Inputs are standardized globally, and results are averaged over five seeds. We compare five three-hidden-layer sigmoid MLP architectures: bottleneck 2-64-16-64-1, constant-width 2-32-32-32-1, increasing-width 2-16-32-64-1, decreasing-width 2-64-32-16-1, and expansion 2-16-64-16-1. All models are trained for $10{,}000$ epochs with batch size 300.
Depending on the experiment, the networks are trained either with LIFTY or with Adam. For the lifted solver, we use sigmoid Fenchel-Young blocks at all layers, a single inner activation-update step per epoch, and parameter $\varepsilon = 10^{-3}$. For the Adam baseline, we use binary cross-entropy and learning rate 10^{-3}.

Two-moons scaling experiment. This experiment is designed to assess how model size and training budget interact under a fixed compute proxy. Following the spirit of compute-optimal scaling analyses, we generate $12{,}000$ samples with noise level 0.2 and use a stratified $80\%/20\%$ train-validation split. Inputs are standardized using training-set statistics only. We consider a family of sigmoid MLPs of the form 2-w-w-1, with widths $w \in \{8, 12, 16, 24, 32, 48, 64\}$.
For each width and each compute budget $C \in \{2, 4, 6, 8, 10\} \times 10^8$, we allocate a token budget according to the proxy $C \simeq N_{\mathrm{params}} D$, where N_{params} is the number of trainable parameters and D is the number of training examples consumed. Models are trained with mini-batches of size 128, and the number of optimization steps is chosen so that the total number of consumed examples approximately matches D. For each compute budget, we select the configuration with the lowest validation loss and fit power laws in log-log space for N_{opt} as a function of C, D_{opt} as a function of C, and D_{opt} as a function of N_{opt}.

We now report the settings for the two tabular classification benchmarks obtained from the UCI Machine Learning Repository (University of California, Irvine, 2018).

Sonar dataset. We follow a setting similar to that of Zhang et al. (2019) and consider a binary classification task on the Sonar dataset from the UCI repository, consisting of distinguishing sonar signals reflected by metal cylinders from those reflected by rocks. The dataset contains 208 samples with 60 input features. We use a fixed one-hidden-layer MLP of architecture 60-10-1 and report results averaged over five seeds. The dataset is split into training, validation, and test sets with proportions 60%, 20%, and 20%, respectively, using stratified sampling. Input features are standardized using statistics computed on the training set only. All models are trained for 2000 epochs with batch size 8. For LIFTY, FY-Adam, and FY-SGD, we use sigmoid hidden activations, while for baselines restricted to piecewise-linear activations we use ReLU. FY-Adam and FY-SGD use a learning rate of 10^{-3}. For the lifted-network baseline of Gu et al. (2020), we use the hyperparameters $\lambda = 1.0$, $\rho = 0.0$, $\rho_d = 10.0$, and a single block-coordinate alternation per iteration. For the lifted Bregman network baseline of Wang & Benning (2023), we use 30 inner Bregman iterations per batch together with proximal regularization parameters $\delta_{W_1} = \delta_{W_2} = 0.5$.

Covertype dataset. We next consider a large-scale multiclass classification setting using the UCI Covertype dataset, where the goal is to predict the dominant forest cover type from cartographic descriptors. The dataset contains $581{,}012$ samples described by 54 input features and spans 7 classes. In contrast to the small-scale Sonar experiment, we employ a wider network with architecture 54-128-7 in order to handle the increased complexity and scale of the task. Results are averaged over five random seeds. As before, we use stratified splits with proportions 60% for training, 20% for

validation, and 20% for testing, and standardize the inputs using statistics computed on the training set only. All methods are trained for 30 epochs with batch size 128.
For our lifted approaches (LIFTY, FY-Adam, and FY-SGD), we use sigmoid hidden activations and a Shannon-entropy output geometry associated with multiclass cross-entropy losses. FY-Adam and FY-SGD are both trained with learning rate 10^{-3}. For the lifted-network baseline of Gu et al. (2020), we use the recommended configuration $\lambda = 1.0$, $\rho = 0.0$, and $\rho_d = 10.0$, with a single block-coordinate alternation at each iteration. Finally, for the lifted Bregman network baseline of Wang & Benning (2023), computational and memory considerations on this large-scale dataset require reducing the number of inner Bregman iterations to one per batch; we use proximal regularization parameters $\delta_{W_1} = \delta_{W_2} = 0.5$ throughout.

Finally, settings regarding classification tasks on image datasets discussed below.

MNIST dataset. We also evaluate our approach on the MNIST handwritten-digit classification benchmark, consisting of $60{,}000$ grayscale images of size 28×28 distributed across 10 classes. Images are normalized using the standard MNIST statistics and flattened into vectors of dimension 784. We consider a fully connected MLP of architecture 784-1000-500-10 with two hidden layers following Cireşan et al. (2010). Results are averaged over seeds $[0, 37, 42, 73, 123]$. The dataset is randomly partitioned into training and validation subsets using proportions 50% and 30%, respectively, while the remaining samples are used for testing. All models are trained for 30 epochs with batch size 64.
For the lifted formulation, we use ReLU hidden activations together with a cross-entropy loss. We follow the approach explained in Remark 9 with $\beta = 100$, $\alpha = 0.01$, and regularization parameter $\varepsilon = 1$.

Fashion-MNIST dataset. We consider the Fashion-MNIST image classification benchmark, restricted to the original training split, and use it as a medium-scale multiclass experiment. The dataset consists of $60{,}000$ grayscale images of size 28×28, distributed across 10 classes. Images are rescaled to $[0, 1]$ and flattened into vectors of dimension 784. For each seed, we split this original training set into training, validation, and test subsets with proportions 50%, 30%, and 20%, respectively, using stratified sampling. We use a fully connected network of architecture 784-256-1024-10 and report the test accuracy obtained at the epoch with best validation accuracy. All models are trained for 30 epochs with batch size 64, and results are averaged over five seeds.
All methods are run with the same hyperpameters used for the MNIST experiment.

E VALIDATION OF ASSUMPTION 2

Empirical validation of the relaxed forward-pass decrease condition. To assess the practical validity of Assumption 2, we monitor the violation term ε_k during training, where ε_k is defined as the smallest nonnegative value required for the relaxed inequality to hold at iteration k. Figure 4 reports its mean value across training epochs for MNIST. It shows that ε_k decreases rapidly during the first epochs and becomes negligible thereafter. These observations indicate that the discrepancy between the lifted and compositional objectives is rapidly controlled in practice, providing empirical support for the relaxed forward-pass decrease condition. Similar observations can be drawn for other architectures and datasets.

F MEMORY USAGE EXPERIMENTS

We measured the peak GPU memory allocation (in MB) on MNIST and FashionMNIST while varying the network depth L. We used the same batch size (256) and fully-connected architectures of the form $784 - (512 \times (L-1)) - 10$ and $784 - (1024 \times (L-1)) - 10$ reported in Tables 5 and 6, respectively.

The results show that LIFTY consistently requires substantially less allocated GPU memory than standard backpropagation; requires less GPU memory than the naive solvers FY-SGD and FY-Adam, despite solving the same lifted objective through automatic differentiation; consistently requires less GPU memory than the lifted baseline of (Wang & Benning, 2023). We omit (Gu et al., 2020) because our available PyTorch implementation is not memory-optimized, and the resulting measurements would therefore not provide a fair comparison.

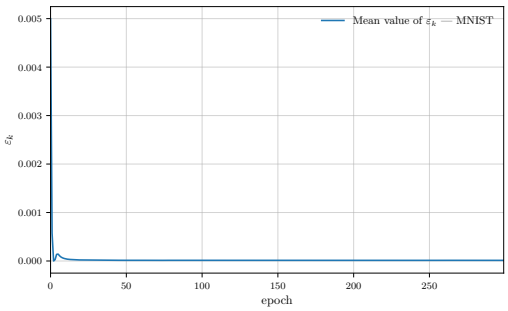


Figure 4: Mean value of ε_k across epochs for LIFTY on MNIST.

Table 5: Peak GPU memory allocation (MB) for the $784 - (512 \times (L-1)) - 10$ architecture on FashionMNIST.

Method / Depth	**2**	**5**	**8**	**10**	**12**	**15**
Backprop (Adam)	25.5	40.5	55.6	65.6	75.6	90.6
LIFTY	28.2	37.4	**46.5**	**52.8**	**58.4**	**70.8**
Wang & Benning (2023)	21.7	**35.5**	49.3	60.8	69.8	85.9
FY-SGD	**20.0**	38.3	56.3	69.8	80.8	99.9
FY-Adam	27.6	55.2	86.5	107.4	127.5	156.6

Moreover, the relative memory advantage of LIFTY becomes more pronounced as the network depth or width increases. These measurements, together with the corresponding discussion, have been included in the revised supplementary material.

Table 6: Peak GPU memory allocation (MB) for the $784 - (1024 \times (L-1)) - 10$ architecture on MNIST.

Method / Depth	**2**	**5**	**8**	**10**	**12**	**15**
Backprop (Adam)	33.3	94.3	157.1	195.3	237.2	295.4
LIFTY	39.0	**79.8**	**112.0**	**140.0**	**168.0**	**210.1**
Wang & Benning (2023)	33.0	86.0	140.1	176.1	212.2	266.1
FY-SGD	**30.4**	94.0	154.1	194.1	234.2	294.1
FY-Adam	45.5	135.3	224.2	284.3	344.3	434.4