TRUST YOUR TARGETS: IS TARGET PROPAGATION BLOCK-DIAGONAL GAUSS-NEWTON?

Anonymous authors
Paper under double-blind review

ABSTRACT

Target propagation (TP) is a backpropagation alternative in which local targets carry the global learning signal. Prior works claim that TP performs Gauss-Newton (GN) optimisation on these targets, but the correspondence has only been established under unrealistic assumptions where hitting them exactly is trivial. Practical variants instead descend on a least-squares problem via a single gradient step. In this paper, we rederive TP from a trust region under an implicit, previously unexamined norm, and introduce CGTP (Conjugate Gradient Target Propagation), which uses conjugate gradients to solve the layer-local subproblems in parallel. A single adaptive parameter jointly controls this trust region and a matching bound on the weight update; KFAC emerges exactly as the special case where the per-sample Jacobians are approximated as identical, giving its factored damping a variational grounding. On MNIST and Fashion-MNIST MLPs, CGTP matches the accuracy of ADAM and KFAC, requiring 10x fewer backward-pass-equivalents than the former, while remaining stable across depth.

1 INTRODUCTION

End-to-end backpropagation paired with first-order optimizers (Kingma & Ba, 2014) is the de facto standard for training neural networks. Despite its remarkable effectiveness, backpropagation-based learning imposes a sequential dependency whereby each layer's gradient depends on every subsequent layer's. Consequently, error signals must propagate fully backward before updates can begin, which is known as the *update locking* problem (Jaderberg et al., 2016a). This dependency increasingly limits layer-parallel training (Wang et al., 2021), and has motivated a family of *local learning* methods that allow layers to perform optimization steps independently against local objectives. These methods span a spectrum: greedy supervised approaches (Belilovsky et al., 2018) achieve full layer independence but suffer from *information collapse* (Wang et al., 2021). Conversely, predictive coding methods (Millidge et al., 2021) simulate global feedback through iterative inference, justified by the free-energy principle Friston & Kiebel (2009). Target propagation (TP) (Lee et al., 2014) sits between these global communication extremes: local targets are inferred via a single pass through an approximate inverse, which in principle enables parallel layerwise optimization. However, to realize the benefits of this decoupled optimization, the local objectives must be theoretically sound. While greedy and predictive-coding methods can be derived from information-theoretic or variational principles (Tishby & Zaslavsky, 2015; Friston & Kiebel, 2009), TP's primary theoretical justification is through a loose analogy to Gauss-Newton (GN) optimization (Meulemans et al., 2020) that is built on assumptions that frequently break down in practice.

The existing proofs that connect TP to GN rely on several assumptions that diverge drastically from practice: they assume batch sizes of 1, single-layer updates, a regression objective, strictly invertible layers, and an *infinitesimal* step size β. Two of these are particularly damaging. First, the infinitesimal-β requirement defeats the purpose of a second-order method, which is sought *for* its faster convergence. Second, the assumption of strictly invertible layers forces practical TP variants to increasingly rely on inexact inverses, via autoencoders (Lee et al., 2014), or in the extreme case fixed random feedback weights Shibuya et al. (2022). This requires expensive feedback-weight training loops and more iterations to converge. Relying on many propagation phases especially runs counter to the goals of a local algorithm: since the targets are derived through a propagation phase, we should endeavour to minimize the number of propagation phases altogether.

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.

The result is a striking mismatch: TP is theoretically motivated as approximate GN, yet its dominant practical variants distance themselves from GN at every step. The local loss - the squared-error reconstruction objective each layer minimizes against its target - is treated as a means to an end: a single gradient step toward a target whose theoretical role is only loosely justified. We argue that this is a missed opportunity: if TP is genuinely a form of Gauss-Newton, then taking the local subproblem seriously - ie reaching the minimum - should yield the convergence advantages over first-order methods seen in modern BP-based second-order optimizers (e.g., K-FAC (Martens & Grosse, 2015), SHAMPOO (Gupta et al., 2018)), but crucially, without the update-locking bottleneck.

In this paper, we give TP a rigorous grounding as a sequence of local trust-region subproblems, which we then solve with a second-order method. Our specific contributions are as follows:

- **(C1) A rigorous derivation of TP under the trust-region framework.** Prior TP-GN equivalence proofs rely on a set of restrictive assumptions, typically left implicit or deferred to appendices. We collect these explicitly (Table 1), and relax each in turn. We derive TP from a linearly constrained least-squares subproblem via a linearized change of variables, accommodating classification objectives, multi-layer updates, finite step sizes, and non-invertible layers (Sections 3.1-3.2). Two decoupling theorems - one via norm choice, one via a simplicial representation - yield interpretable, local subproblemss whose unconstrained optima recover the regularized-inverse TP update.
- **(C2) A conjugate gradient solver, with KFAC as a special case.** KKT analysis of the constrained per-layer subproblem shows that the targets must be fit in the curvature-induced metric, not the Euclidean one, explaining why a naive second-order fit of the TP loss fails (Section 4.1). The null-space method reduces each layer to a positive definite system in the weights, solved in parallel by CG after a single backward pass (Section 4.3). Averaging the per-sample curvatures recovers KFAC with factored damping exactly (Corollary 12).
- **(C3) Foundational empirical validation.** On MNIST and Fashion-MNIST, we demonstrate proof-of-concept results that indicate our theoretically rigorous approach (TRTP) achieves order-of-magnitude reductions in outer iterations relative to BP+Adam, KFAC, and existing TP variants, while maintaining stability across depth (Section 5).

2 Identifying the Theoretical Foundations of Target Propagation

2.1 Preliminary: Classical Target Propagation

In this section, we briefly review the common presentation of TP. Consider a dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ of N samples. We fit an L-layer feedforward multilayer perceptron (MLP) computing:

$$u_n^{(0)} = x_n; \quad z_n^{(\ell+1)} = W^{(\ell)\top} u_n^{(\ell)} + b_n^{(\ell)}; \quad u^{(\ell+1)} = \sigma(z^{(\ell+1)}) \tag{1}$$

The output $z_n^{(\text{out})} = W^{(L)\top} u_n^{(L)} + b^{(L)}$ is trained to minimize the per-sample loss $\mathcal{L}_n(z_n^{(\text{out})}, y_n)$. To assign credit, TP begins by defining a target for the output via a gradient step on the loss:

$$\hat{z}_n^{(\text{out})}(\beta) = z_n^{(\text{out})} - \beta \frac{\partial \mathcal{L}_n}{\partial z^{(\text{out})}} := z_n^{(\text{out})} - \beta r_n, \tag{2}$$

where β is a step size and r_n denotes the (generalized) residual. The composite mapping $f^{(\ell)} : z^{(\ell)} \to z^{(\ell+1)}$ are then approximately inverted to compute intermediate targets $\hat{z}_n^{(\ell)}(\beta) = f^{(\ell)+}(\hat{z}_n^{(\ell+1)}(\beta))$. These targets define a layer-local regression problem, from which parameter gradients are derived:

$$L_{\text{TP}}^{(\ell)}(W^{(\ell)}, b^{(\ell)}) = \frac{1}{\beta} \sum_n ||W^{(\ell)\top} u_n^{(\ell)} + b_n^{(\ell)} - \hat{z}_n^{(\ell+1)}(\beta)||^2 \tag{3}$$

2.2 Classical Target Propagation as Hybrid Gauss Newton

The connection between the GN method and TP, as presented in such works as (Jaderberg et al., 2016b; Meulemans et al., 2020; Roulet & Harchaoui, 2021) requires several assumptions to draw formal equivalence, which we collect in Table 1. Let $F^{(\ell',\ell)}$ denote the composite map from $z^{(\ell)}$ to $z^{(\ell')}$, with Jacobian $J^{(\ell',\ell)}$. GN is a second-order method applied to least-squares problems (2\2).

Table 1: Assumptions invoked by prior proofs of the TP-GN equivalence

Assumption	Relaxed in
$\mathfrak{A}\mathbf{1}$: Infinitesimal step size β	Section 3: Trust-region radius
$\mathfrak{A}\mathbf{2}$: ℓ_2 regression loss	Section 3.1: Generalized surrogate loss
$\mathfrak{A}\mathbf{3}$: Single-layer updates	Section 3.2: Norm-penalties
$\mathfrak{A}\mathbf{4}$: Strictly invertible layers	Section 3.2: Norm-penalties, Section 4.2: Simplicial decoupling
$\mathfrak{A}\mathbf{5}$: Mini-batch of size 1	Section 4.1: Constrained activation space

Consider the following objective, for inferrring the optimal target of a single ($\mathfrak{A}3$) intermediate layer $z_n^{(l)}$ for a single datapoint ($\mathfrak{A}5$):

$$\mathcal{L}_n^{(\ell)} = \frac{1}{2}\left\|\hat{z}_n^{(\text{out})} - F^{(\text{out},\ell)}(z_n^{(\ell)})\right\|^2, \tag{4}$$

A GN step proceeds by linearizing $F^{(\text{out},\ell)}$ around the current $z_n^{(\ell)}$ and solving the resulting linear least-squares problem. Writing the residual as $r_n = F^{(\text{out},\ell)}(z_n^{(\ell)}) - \hat{z}_n^{(\text{out})}$, the GN update is

$$[\Delta z_{\text{GN}}^{(\ell)}]_n = \left(J^{(\text{out},\ell)\top} J^{(\text{out},\ell)}\right)^+ J^{(\text{out},\ell)\top} r. \tag{5}$$

If F is invertible ($\mathfrak{A}4$), then so is the Jacobian, and the above simplifies to:

$$[\Delta z_{\text{GN}}^{(\ell)}]_n = [J_n^{(\text{out},\ell)}]^{-1} r_n. \tag{6}$$

Let us return to the TP loss from Equation 3, considered as a function of the intermediate activation $z_n^{(\ell)}$. In the limit that the step-size $\beta \to 0$ ($\mathfrak{A}1$), a Taylor expansion on the TP gradient yields:

$$[\Delta z_n^{(\ell)}]_{\text{TP}} = z_n^{(\ell)} - [F^{(out,\ell)}]^{-1}(z_n^{(out)} - \beta r_n) \tag{7}$$

$$\approx z_n^{(\ell)} - [F^{(out,\ell)}]^{-1}(z_n^{(out)}) + \beta \frac{\partial [F^{(out,\ell)}]^{-1}(z_n^{(out)})}{\partial z_n^{(out)}} r_n = \beta \frac{\partial [F^{(out,\ell)}]^{-1}(z_n^{(out)})}{\partial z_n^{(out)}} r_n \tag{8}$$

By the inverse function theorem, the Jacobian of G is the inverse of the Jacobian of F. As such:

$$[\Delta z_n^{(\ell)}]_{\text{TP}} = \beta [J_n^{(out,\ell)}]^{-1} r_n = \beta [\Delta z_{\text{GN}}^{(\ell)}]_n, \tag{9}$$

which demonstrates the equivalence. Crucially, this is *not* a GN step on the weights. TP as presented in (Meulemans et al., 2020) is a hybrid algorithm: a GN step infers local targets, and a gradient descent step converts those targets into a weight update. The theory for a full TP-based GN update on the weights (Bengio, 2020; Meulemans et al., 2020) has been developed only in the single-datapoint case ($\mathfrak{A}5$), where any inferred target is exactly realizable and the TP step coincides with a full GN step up to a positive scalar. The remainder of this paper relaxes each of these assumptions in turn.

3 Generalising Target Propagation via Trust-Region Methods

The classical presentation of TP uses a single parameter (β) to control both the the approximation fidelity and step size. Trust-Regions (TR) methods separate these roles, permitting accelerated optimization even with poor approximations. One constructs a local quadratic model m of the objective, takes a step that approximately minimizes m within a radius Δ, and then accepts/rejects the step based on the ratio ρ between the predicted reduction (in m) and the reduction of the true loss. The radius adapts based on ρ: a successful step extends it, a failed step contracts it.

In what follows we generalise TP within this framework, and relax the remaining assumptions in the proof of Section 2.2. Section 3.1 identifies the loss that TP approximates, clarifying its role as a surrogate for general loss functions ($\mathfrak{A}2$), and how its fidelity is measured through the trust-region ratio ρ. Section 3.2 demonstrates that a principled choice of trust-region metric naturally accomodates the multi-layer setting ($\mathfrak{A}3$) and non-invertible layers ($\mathfrak{A}4$).

3.1 RELAXING 𝔄2 VIA CONVEX SURROGATES

Adopting the Trust Region framework for TP requires identifying *which* objective it is trying to approximate. We propose that the TP loss in Equation 3 can be viewed as a surrogate of the true objective $\mathcal{L}$, insofar as (see Supp B):

$$||\nabla_{z^{(\ell)}}\mathcal{L}_{\text{TP}}||^2 = 0 \implies ||\nabla_{z^{(\text{out})}}\mathcal{L}||^2 = 0 \tag{10}$$

The key observation is that the objectives used to train neural networks are commonly convex functions of the outputs. As such, the vanishing of $||\nabla_{z^{(out)}}\mathcal{L}||$ is a sufficient condition for optimality of the network, motivating the following general surrogate loss:

$$\mathcal{J}_n = ||\nabla_{z^{(out)}}\mathcal{L}(z^{(out)}(x_n, y_n; \theta)||^2_{M_n} \tag{11}$$

Here, $M_n \succeq 0$ denotes a choice of norm appropriate to the problem. Letting $r_n \coloneqq \nabla_{z^{(out)}}\mathcal{L}_n$ denote the generalized residual, we consider the result of applying a perturbation $\Delta z_n^{(\ell)}$ to the pre-activation of layer ℓ at the point n. Taylor expanding, we have:

$$r_n(z_n^{(\ell)} + \Delta z_n^{(\ell)}) \approx r_n + H_n J_n^{(out,\ell)} \Delta z_n^{(\ell)}; \quad H_n \coloneqq \nabla^2_{z_n^{(out)}}\mathcal{L}_n \tag{12}$$

We then obtain the following subproblem:

Definition 1 (Generalized Gauss-Newton Subproblem for the Pre-Activations)**.** For a general objective $\mathcal{L}$, the surrogate loss $\mathcal{J}_n$ may be approximated as a weighted least-squares subproblem:

$$\mathcal{J}_n \approx ||M_n^{1/2}(r_n + H_n J_n^{(out,\ell)} \Delta z_n^{(\ell)})||^2 \coloneqq m_n(\Delta z_n^{(\ell)}) \tag{13}$$

The improvement ratio ρ is then calculated by comparing $\Delta\mathcal{J}_n$ (measured) and Δm_n (predicted).

3.2 RELAXING 𝔄3 AND 𝔄4 VIA A PRINCIPLED CHOICE OF TRUST-REGION METRIC

Prior works generalize the single-layer result of Definition 1 to the multi-layer setting by appealing to the common empirical approximation of a block-diagonal Gauss-Newton matrix (Meulemans et al., 2020). In this section, we show that achieving layer decoupling requires no such approximation - only a principled choice of metric. The key observation is that the multi-layer problem is typically underdetermined. Let J_n denote the stacked Jacobian and Δz_n the stacked perturbation, as follows:

$$J_n \coloneqq \left[\, I \;\middle|\; J_n^{(\text{out},L)} \;\middle|\; \cdots \;\middle|\; J_n^{(\text{out},1)} \,\right]; \quad \Delta z_n \coloneqq \left[\, \Delta z_n^{(\text{out})\top} \;\middle|\; \Delta z_n^{(L)\top} \;\middle|\; \cdots \;\middle|\; \Delta z_n^{(1)\top} \,\right]^\top \tag{14}$$

Henceforth we drop the out, denoting $J_n^{(\ell)} \coloneqq J_n^{(\text{out},\ell)}$. The natural generalization of Definition 1 is:

$$m_n(\Delta z_n) = \left\| r_n + H_n J_n \Delta z_n \right\|^2_{M_n} \tag{15}$$

Note that, if each $\Delta z_n^{(\ell)} \in \mathbb{R}^K$, the joint perturbation across layers introduces $(L+1)K$ unknowns against only $C \ll K$ residuals, and so there are infinitely many solutions. In the trust region setting, a unique solution may be fixed via a norm constraint $||\Delta z_n||_{B_n} < \Delta$, for choice of metric B_n. The following theorem highlights a specific family of norms that natively decouples the layers:

Theorem 2 (Egalitarian Norm-Induced Layer Separation)**.** *Let λ be a Lagrange multiplier, and consider the penalty form of the norm constraint for metric B_n:*

$$m_n(\Delta z_n; \lambda) = \tfrac{1}{2}\left\| r_n + H_n J_n \Delta z_n \right\|^2_{M_n} + \tfrac{\lambda}{2} \Delta z_n^\top B_n \Delta z_n \qquad B_n = \bigoplus_{\ell=1}^{L+1} B_n^{(\ell)} \tag{16}$$

Then, if the metric is egalitarian (ie satisfies $\sum_\ell J_n^{(out,l)}[B_n^{(l)}]^{-1}[J^{(out,l)}]_n^\top = I$), the unconstrained solution for the layer perturbation is:

$$\Delta z_n^{(\ell)} = -\tfrac{1}{L} J_n^{(\ell)+} \tilde{r}_n(\lambda) = -\tfrac{1}{L}[B_n^{(\ell)}]^{-1} J_n^{(\ell)\top} \tilde{r}_n(\lambda) \tag{17}$$

Where $\tilde{r}_n(\lambda)$ denotes the regularized target, and is given by:

$$\tilde{r}_n(\lambda) = \left(H_n^\top M_n H_n + \lambda I\right)^{-1} H_n^\top M_n r_n \tag{18}$$

Proof. The identiy block in J_n makes it surjective (Lemma 14), so that for any B_n the the normal equations factor as $(\text{operator})J_n\Delta z_n = (\text{target})$. The egalitarian block requirement is then chosen to coincide with the classical TP update. Full details in Supp C.1. □

Remark 3 (Recovering Classical TP as the Low-Trust Limit in an Egalitarian Newton Norm). As $\lambda \to \infty$, choosing penalty norm $M_n = H_n^{-1}$, the regularized target of 17 approaches the Classical TP target of 9 with $\beta = \lambda^{-1}$.

Remark 4 (Interpretation of the Egalitarian Norm). Theorem 2 has a natural reading: no single layer should dominate the others in its contribution to the regularized objective. This is structurally similar to the scale-invariant diagonal damping used in classical Levenberg-Marquardt (LM) (Nocedal & Wright, 2006). Furthermore, the pseudo-inverse leaves unperturbed those direction of the pre-activations that the layer discards. Supp C.2 makes this construction explicit for the output layer.

4 Beyond Classical TP: Second-Order Methods on the Local Subproblem

Having relaxed Assumptions 𝔄1-𝔄4, we now address 𝔄5, and the prospect of deriving GN updates from the TP loss. The hybrid route of prior works (a single gradient step on the weights) is naturally read as the low-trust regime of our framework. Where previous approaches assess (and adapt) approximate inverses through reconstruction, the trust-region formulation calibrates trust through *utility*. A second-order step is licensed when reductions in the subproblem align with reduction in the true loss. It then seems a simple matter to apply a second-order optimiser to equation 3: the Hessian is block diagonal, and the Newton step yields the familiar least-squares fit. We will show in this section that doing so is not straightforward. The primary difficulty is one of feasibility: equation 3 assesses attainability of the local targets a Euclidean norm. When 𝔄5 holds, the targets are exactly realisable and so the choice of metric is irrelevant. For larger batch sizes, this choice of proves decisive.

To this end, in Section 4.1, we extend our trust-region subproblem to incorporate feasibility constraints. The KKT conditions reveal the necessity of measuring fit using the metric induced by curvature. The numerical cost of the KKT solution motivates a majorization-minimization decomposition of that metric (Section 4.2), yielding new local losses that generalize equation 3 and admit second-order optimizers (Section 4.3). We then damp the weight updates to combat overfitting (Section 4.4), recovering KFAC as a corrollary when samples are replaced by their batch average.

4.1 KKT Conditions for the Constrained Subproblem

The natural generalisation of the subproblem of Equation 15 is to simply sum m_n across datapoints. This constitutes N independent least-squares problems, whose minimisers are available in closed form via Theorem 2. However, such solutions ar not necessarily feasible: the model architecture constrains the space of pre-activations to satisfy $\Delta Z^{(\ell)} = \tilde{U}^{(\ell-1)}\Delta W^{(\ell-1)}$ for some $\Delta W^{(l-1)}$ (here, $\sim$ indicates augmentation by ones). Note, the constraint requires only that $\Delta W^{(\ell-1)}$ exists: it does not commit us to any particular choice. Hence, to retain the pre-activations as the principal optimisation variables, we must incorporate the following constraint:

Lemma 5 (Feasibility of perturbations (layer version)). *Let $\tilde{U}^{(\ell-1)} \in \mathbb{R}^{N\times d_{\ell-1}}$ and $\Delta Z^{(\ell)} \in \mathbb{R}^{N\times d_\ell}$. There exists $\Delta W^{(\ell-1)} \in \mathbb{R}^{d_{\ell-1}\times d_\ell}$ satisfying $\Delta Z^{(\ell)} = \tilde{U}^{(\ell-1)}\Delta W^{(\ell-1)}$ if and only if each column of $\Delta Z^{(\ell)}$ lies in $\mathrm{col}(\tilde{U}^{(\ell-1)})$. Equivalently,*

$$\Pi^{(\ell-1)}\,\Delta Z^{(\ell)} = 0, \tag{19}$$

where $\Pi^{(\ell-1)}$ is the orthogonal projector onto $\mathrm{col}(\tilde{U}^{(\ell-1)})^{\perp}$.

Proof. The equation $\Delta Z^{(\ell)} = \tilde{U}^{(\ell-1)}\Delta W^{(\ell-1)}$ holds column-wise: each column of $\Delta Z^{(\ell)}$ is a linear combination of the columns of $\tilde{U}^{(\ell-1)}$ with coefficients given by the corresponding column of $\Delta W^{(\ell-1)}$. Such a linear combination exists if and only if the column lies in $\mathrm{col}(\tilde{U}^{(\ell-1)})$, which is equivalent to vanishing under projection onto its orthogonal complement. □

Remark 6 (Classical TP as Projected Gradient Descent). The classical TP weight update, when applied to the inputs $U^{(\ell-1)}$, is effectively a projection of the targets into $\mathrm{col}(\tilde{U}^{(\ell-1)})$ via $\Delta\bar{Z}^{(\ell)}_{\text{PGD}} \propto \tilde{U}^{(\ell-1)}\tilde{U}^{(\ell-1)\top}\Delta Z^{(\ell)}$ (Supp C.3).

Lemma 5 holds at the level of a single layer. In what follows we require a constraint acting on all layers at once, after stacking across datapoints. This will keep the per-sample subproblems block-diagonal, but distributes the layer constraint across the stacked vector. This new constraint is expressed in the following Lemma (proof in Supp C.4):

Lemma 7 (Feasibility of perturbations (stacked version)). *Let the stacked pre-activations be denoted $\Delta \boldsymbol{z} = (\Delta z_1, \dots, \Delta z_N) \in \mathbb{R}^{ND}$, and let $E^{(\ell)}$ be defined such that $\Delta z_n^{(\ell)} = E^{(\ell)} \Delta z_n$. Then the layer feasibility operators of Lemma 5 may stated as $\boldsymbol{\Pi} \Delta \boldsymbol{z} = 0$, where*

$$\boldsymbol{\Pi} = \sum_{\ell} \Pi^{(\ell-1)} \otimes E^{(\ell)\top} E^{(\ell)}. \tag{20}$$

We are now ready to examine the constrained subproblem.

Theorem 8 (Solution of the Joint Constrained Subproblem). *Let $G_n \coloneqq J_n^\top H_n^\top M H_n J_n$ denote the Hessian of the per-datapoint subproblem, and $\tilde{G}_n(\lambda) \coloneqq G_n + \lambda B_n$ its regularisation. The unconstrained problems are uncoupled across datapoints, so we define the stacked subproblem Hessian $\tilde{\boldsymbol{G}}(\lambda) = \bigoplus_n \tilde{G}_n(\lambda)$. Finally, let $\Delta \boldsymbol{z}^\star(\lambda)$ denote the stacked unconstrained minimiser of equation 17. Then, up to an additive constant in $\Delta \boldsymbol{z}$, the constrained joint subproblem is:*

$$m(\Delta \boldsymbol{z}; \lambda) = \big(\Delta \boldsymbol{z} - \Delta \boldsymbol{z}^\star(\lambda)\big)^\top \tilde{\boldsymbol{G}} \big(\Delta \boldsymbol{z} - \Delta \boldsymbol{z}^\star(\lambda)\big) \quad \textit{subject to} \quad \boldsymbol{\Pi} \Delta \boldsymbol{z} = 0 \tag{21}$$

The optimal perturbations are then given by projecting $\Delta \boldsymbol{z}^\star(\lambda)$ onto $\ker \boldsymbol{\Pi}$ in the metric induced by $\tilde{\boldsymbol{G}}(\lambda)$:

$$\Delta \boldsymbol{z}^\star_{\boldsymbol{\Pi}}(\lambda) = \Delta \boldsymbol{z}^\star(\lambda) - \tilde{\boldsymbol{G}}(\lambda)^{-1} \boldsymbol{\Pi}^\top \big(\boldsymbol{\Pi} \tilde{\boldsymbol{G}}(\lambda)^{-1} \boldsymbol{\Pi}^\top\big)^{-1} \boldsymbol{\Pi} \Delta \boldsymbol{z}^\star(\lambda). \tag{22}$$

The proof follows from KKT analysis (Supp C.5). The naive Newton step on Equation 3, by contrast, is a *Euclidean* projection: it treats all directions in ΔZ space on equal footing. When G_n is singular, directions in its null space leave the local loss exactly unchanged, so the constraint can be satisfied by routing the correction through this null space at zero cost; damping replaces "zero cost" with "low cost," graded by proximity to the null space, but the cheap subspace persists. The Euclidean projection has no access to this structure and generically pays for directions the curvature metric would have obtained for free.

Unfortunately, computing equation 22 directly is not feasible. The Schur complement $\boldsymbol{\Pi} \tilde{\boldsymbol{G}}(\lambda)^{-1} \boldsymbol{\Pi}^\top$ is dense and of dimension $N(L+1)K \times N(L+1)K$. The product inherits neither the block diagonal structure of $G(\lambda)$ nor the Kronecker structure of C, making the cross-layer couplings unavoidable and thus prohibitive at practical scale.

4.2 Local Surrogates Through Simplicial Relaxations

Section 3 recovered the TP update by choosing a block-diagonal norm, which resolves the under-determination of the multi-layer subproblem and yields layer-separable targets. The curvature, however, remains coupled across layers. This suffices for inferring local targets, but not hitting them with a second-order method. In this section we separate the curvature itself, replacing the subproblem with a layer-local surrogate that majorises it.

Consider, for each datapoint n, a set of L scalars $\alpha_n^{(\ell)} > 0$ with $\sum_\ell \alpha_n^{(\ell)} = 1$, and the family of perturbations satisfying $J_n^{(\ell)} \Delta z_n^{(\ell)} = -\alpha_n^{(\ell)} r_n$. By inspection, this family minimises the subproblem in the absence of constraints: the final layer residual is simply split across the L-layers, and the local residual may be fit exactly. This perspective enables us to identify a principled decomposition which generalises to the constrained setting. It is encapsulated in the following theorem:

Theorem 9 (Layer-separable majorization). *Let $M \succeq 0$ and let α_n lie in the interior of the simplex. Then the following is a majorizer of m_n at every value of α_n*

$$m_n(\Delta z) \;\le\; \frac{1}{2} \sum_{\ell} \Big\| \sqrt{\alpha_n^{(\ell)}}\, r_n + \tfrac{1}{\sqrt{\alpha_n^{(\ell)}}} H_n J_n^{(\ell)} \Delta z_n^{(\ell)} \Big\|_{M_n}^2 \;\equiv\; \widehat{m}_n(\Delta z; \alpha_n), \tag{23}$$

Furthermore, in the presence of a regulariser $\lambda B_n^{(\ell)}$, the following sequence converges to the minimum of the local subproblem:

$$\textit{(A)} \quad \Delta z_n^{(\ell)} \leftarrow -\alpha_n^{(\ell)} \big(G_n^{(\ell)} + \alpha_n^{(\ell)} \lambda B_n^{(\ell)}\big)^{-1} g_n^{(\ell)}, \qquad \textit{(B)} \quad \alpha_n^{(\ell)} \leftarrow \frac{\|\Delta z_n^{(\ell)}\|_{G_n^{(\ell)}}}{\sum_k \|\Delta z_n^{(k)}\|_{G_n^{(k)}}}, \tag{24}$$

where $G_n^{(\ell)} = J_n^{(\ell)\top} H_n^\top M_n H_n J_n^{(\ell)}$ *and* $g_n^{(\ell)} = J_n^{(\ell)\top} H_n^\top M_n r_n$.

Proof. Write $r_n = (\sum_\ell \alpha_n^{(\ell)}) r_n$ and apply Jensen's inequality to $\|\cdot\|_{M_n}^2$. The main result is then alternating minimisation after proving that $\widehat{m}_n(\Delta z_n; \alpha_n) + \frac{1}{2}\lambda ||\Delta z_n||_{B_n}^2$ is convex (Supp C.6). □

Note that *any* α on the simplex yields a valid subproblem (equation 23). We list several approximations in Supp D.3, all of which. empriically, converge faster than the uniform baseline $\alpha_n^{(\ell)} = L^{-1}$.

4.3 Null-Space Methods on the Local Surrogate

The per-layer surrogate of Theorem 9 is significantly smaller than the full multi-layer subproblem of Theorem 8, but its $NK \times NK$ Hessian still lacks the structure that would admit the closed-form KKT solution of Supp C.5. While the Schur complement is expensive to form, its components - block-diagonal Hessian $\tilde{\boldsymbol{G}}(\lambda)$ and the Kronecker-structured projector $\boldsymbol{\Pi}$ - admit cheap matrix-vector products. Hence, methods like conjugate gradient (CG) are a natural choice of solver, as they only require the action of each component (Nocedal & Wright, 2006). Though one may use projected CG to optimise ΔZ directly, the bi-level approach of the null-space method is more memory efficient. As in Classical TP, we identify local targets $\Delta Z^{(\ell)\star}(\alpha^{(\ell)})$ via the inversion in Theorem 9. The null-space of the constraint is then simply $\Delta Z^{(\ell)}(\Delta W^{(\ell-1)}) = U^{(\ell-1)} \Delta W^{(\ell-1)}$, and we minimise the reduced system, which is a weighted least-squares fit:

$$\min_{\Delta W^{(\ell-1)}} \frac{1}{2} \left\| \tilde{U}^{(\ell-1)} \Delta W^{(\ell-1)} - \Delta Z^{(\ell)\star}(\alpha^{(\ell)}) \right\|_{\tilde{\boldsymbol{G}}^{(\ell)}}^2, \qquad \tilde{\boldsymbol{G}}^{(\ell)} = \bigoplus_n \tilde{G}_n^{(\ell)}(\lambda; \alpha_n^{(\ell)}). \tag{25}$$

The gradient is $\sum_n u_n \left(\Delta W^{(\ell-1)\top} u_n - \Delta z_n^{(\ell)\star}\right)^\top \tilde{G}_n^{(\ell)}$. To proceed, we define the linear operator $\mathcal{G}_\lambda^{(\ell)}(V) \coloneqq \sum_n u_n u_n^\top V \tilde{G}_n^{(\ell)}(\lambda; \alpha)$, and use CG to solve:

$$\mathcal{G}_\lambda^{(\ell)}\left(\Delta W^{(\ell-1)}\right) = -\sum_n u_n g_n^{(\ell)\top}. \tag{26}$$

4.4 Preventing Overfitting with Adaptive Weight Decay

Solving the local subproblem to high accuracy is not uniformly beneficial. Though CGTP reduces training loss far faster than first-order baselines, its validation error begins to rise after a small number of outer iterations, leading to a performance gap when compared to BP+ADAM. It is easy to see the cause: while the trust region bounds the magnitude permissible $\Delta Z^{(\ell)}$, it does nothing to bound $\Delta W^{(\ell-1)}$, which can be arbitrarily large whenever $U^{(\ell-1)}$ is poorly conditioned. In this section, we suggest an adaptive strategy for controlling the conditioning.

Choice of norm. Phase one fixes a metric on pre-activation space: solving the damped subproblem declares that the effective local geometry of $\Delta z_n^{(\ell)}$ is $\tilde{\boldsymbol{G}}^{(\ell)})$ rather than the Euclidean one. Any quantity that maps *into* pre-activation space should be measured accordingly. A weight perturbation $I \otimes \Delta W^{(\ell-1)}$ is a linear map from the metric space $(\mathbb{R}^{N \times K_{\text{in}}}, I)$ to $(\mathbb{R}^{N \times K_{\text{out}}}, \tilde{\boldsymbol{G}}^{(\ell)})$, and its size is its operator (Hilbert-Schmidt) norm between those two inner-product spaces:

$$\|I \otimes \Delta W^{(\ell-1)}\|_{I \to \tilde{\boldsymbol{G}}^{(\ell)}}^2 = \sum_n \operatorname{trace}(\Delta W^{(\ell-1)\top} \Delta W^{(\ell-1)} \tilde{G}_n^{(\ell)}(\lambda; \alpha_n^{(\ell)})) \tag{27}$$

$$\implies \|\Delta W^{(\ell-1)}\|_{\text{HS}} = \|\Delta W^{(\ell-1)} \bar{G}^{1/2}\|_F^2 + \lambda \|\Delta W^{(\ell-1)} \bar{B}^{1/2}\|_F^2. \tag{28}$$

When $B_n^{(\ell)} = I$ second term is the ordinary weight decay, while the first term controls a local Lipschitz constant that penalizes overall model sensitivity.

Regularization Strength. Each phase now has its own constraint: during inference, $\|\Delta Z^{(\ell)}\|_F \leq \delta_z$ is controlled by an LM rule on the reduction ratio, whereas $\|\Delta W^{(\ell-1)}\|_{\text{HS}} \leq \delta_w$ is to be enforced during the second weight-fitting phase. Independently adapting a second radius would leave us with a two-trust-region problem, which, while being well-studied, introduces additional complexity (Ai &

Zhang, 2009; Bienstock, 2014). We instead observe that our constraint sets are both convex and have non-zero intersection ($\Delta W = 0$ lies in the interior of both), and may thus use a single trust-region parameter to limit the weight perturbation to the intersection. This leads us to the following defintions, and the resulting CG system (Proof in Supp C.7, full listing in Algorithm 1).

Theorem 10 (Single-parameter coupling of the two trust regions)**.** *Fix a layer ℓ and suppress the layer index. Let $\delta_z, \delta_w > 0$ be fixed radii and $\lambda > 0$ a trust parameter, and set*

$$\mu_z = \frac{\lambda}{\delta_z^2}, \qquad \mu_J = \frac{\lambda}{\delta_w^2}, \qquad \tilde{G}_n(\mu_z;\alpha) = \frac{1}{\alpha_n} G_n + \mu_z I, \qquad \bar{\tilde{G}} = \bar{G} + \mu_z I, \tag{29}$$

with $\bar{G} = \frac{1}{N}\sum_n \frac{1}{\alpha_n} G_n$, so that the damping of the pre-activation bound is also the metric of the weight bound. With

$$q_z(\Delta W) = \frac{1}{\delta_z^2}\|\tilde{U}\Delta W\|_F^2, \qquad q_w(\Delta W) = \frac{1}{\delta_w^2}\|\Delta W \bar{\tilde{G}}^{1/2}\|_F^2, \tag{30}$$

the ellipsoid $\{q_z + q_w \le 1\}$ is inscribed in $\{q_z \le 1\} \cap \{q_w \le 1\}$, and penalising it with the single multiplier λ gives

$$L_\lambda(\Delta W) = \sum_n \frac{1}{2}\Big\|\sqrt{\alpha_n}\, r_n + \frac{1}{\sqrt{\alpha_n}} J_n \Delta W^\top u_n\Big\|_{M_n}^2 + \frac{\mu_z}{2}\|\tilde{U}\Delta W\|_F^2 + \frac{\mu_J}{2}\|\Delta W \bar{\tilde{G}}^{1/2}\|_F^2 \tag{31}$$

$$= \text{const} + \frac{1}{2}\|\tilde{U}\Delta W - \Delta Z^\star\|_{\tilde{\boldsymbol{G}}}^2 + \frac{\mu_J}{2}\|\Delta W \bar{\tilde{G}}^{1/2}\|_F^2. \tag{32}$$

Its unique minimiser solves

$$\mathcal{G}_\lambda(\Delta W) = -\sum_n u_n g_n^\top, \qquad \mathcal{G}_\lambda(V) \coloneqq \sum_n u_n u_n^\top V \tilde{G}_n(\mu_z;\alpha) + \mu_J V \bar{\tilde{G}}. \tag{33}$$

Corollary 11 (General pre-activation norm)**.** *Theorem 10 holds with the pre-activation bound measured in per-sample norms $B_n \succ 0$, upon replacing $\mu_z I$ by $\mu_z B_n$ in $\tilde{G}_n$, $\|\tilde{U}\Delta W\|_F^2$ by $\sum_n \|\Delta W^\top u_n\|_{B_n}^2$ in q_z, and $\bar{\tilde{G}}$ by $\bar{G} + \mu_z \bar{B}$ with $\bar{B} = \frac{1}{N}\sum_n B_n$. The CG system becomes*

$$\mathcal{G}_\lambda(\Delta W) = -\sum_n u_n g_n^\top, \qquad \mathcal{G}_\lambda(V) \coloneqq \sum_n u_n u_n^\top V\Big(\frac{1}{\alpha_n} G_n + \mu_z B_n\Big) + \mu_J V\big(\bar{G} + \mu_z \bar{B}\big). \tag{34}$$

Proof. Identical to that of Theorem 10, replacing I by B_n in the pre-activation penalty and by $\bar{B}$ in the weight metric. Positive definiteness of $\mathcal{G}_\lambda$ follows from $\bar{G} + \mu_z \bar{B} \succeq \mu_z \bar{B} \succ 0$. □

Corollary 12 (Exact recovery of KFAC)**.** *Suppose the per-sample metrics are replaced by their batch average, $\tilde{G}_n(\mu_z;\alpha) \equiv \bar{\tilde{G}} = \bar{G} + \mu_z I$ for all n. Then $\mathcal{G}_\lambda$ is Kronecker-structured,*

$$\mathcal{G}_\lambda(V) = \big(A + \mu_J I\big)\, V\, \big(\bar{G} + \mu_z I\big), \qquad A = \tilde{U}^\top \tilde{U}, \tag{35}$$

and the solution of equation 33 is

$$\Delta W = -\big(A + \mu_J I\big)^{-1} \nabla_W L\, \big(\bar{G} + \mu_z I\big)^{-1}, \qquad \nabla_W L = \sum_n u_n g_n^\top, \tag{36}$$

5 Experiments

Datasets and architecture. We evaluate on two low-dimensional classification benchmarks: MNIST (LeCun & Cortes, 2010) and Fashion-MNIST (Xiao et al., 2017). All models are fully connected networks with $\tanh$ nonlinearities, a softmax output layer, and cross-entropy loss. Unless otherwise specified, networks have depth 5 with 256 hidden units. As an ablation, we consider four variants of trust-region target propagation:

- **GDTP**: Gradient descent step on the Euclidan subproblem (equation 58). We tune the learning rate, batch size, and Tikhonov dampening for the inverse of J_n.
- **EucGTP**: Naive fit of the targets (no curvature). Same parameters as GDTP, minus learning rate.

Table 2: Error and convergence cost, in backward-pass equivalents. Outer iterations are counted to the error threshold noted in each column header. $\times$ denotes failure to reach threshold. N/A indicates result is not reported by source.

	F-MNIST				MNIST			
alg	final err	outer @ 0.13	/it	total	final err	outer @ 0.05	/it	total
ADAM	$.111 \pm .000$	$25{,}667 \pm 4509$	1	25.7k	$.019 \pm .000$	7333 ± 577	1	7.3k
KFAC	$.108 \pm .002$	2700 ± 500	1	2.7k	$.018 \pm .001$	467 ± 58	1	467
DRL-lin.	$.111 \pm .004$	N/A	N/A	N/A	$.020 \pm .008$	N/A	N/A	N/A
TJML	$.114 \pm .001$	1468 ± 724	1	1.5k	$.018 \pm .001$	1377 ± 756	1	1.4k
TPRI	$.152 \pm .004$	$\times$	$\times$	$\times$	$.061 \pm .003$	$\times$	$\times$	$\times$
GDTP	$.147 \pm .000$	$\times$	$\times$	$\times$	$.092 \pm .000$	$\times$	$\times$	$\times$
EucTP	$.146 \pm .002$	$\times$	$\times$	$\times$	$.054 \pm .002$	$\times$	$\times$	$\times$
CGTP (no WD)	$.113 \pm .000$	15.0 ± 0.0	201	3.0k	$.037 \pm .001$	23.3 ± 2.9	201	4.7k
CGTP	$.109 \pm .002$	6.67 ± 0.58	201	1.3k	$.019 \pm .001$	3.67 ± 0.58	201	738

- **CGTP (no WD)**(Section 4.3): CG without weight decay. An outer iteration costs one backward pass (to form J_n, g_n) and up to `max_iter`$= 100$ CG iterations (one JVP + one VJP).
- **CGTP** (Section 4.4): automatically adjusted weight decay.

Additionally, we benchmark against five baselines: two general optimisers and three TP variants.

- **BP+Adam**: standard end-to-end training. We tune the learning rate and batch size.
- **KFAC**: (Martens & Grosse, 2015) an approximation of Natural Gradients using Kronecker-Factored approximations. We tune the batch size, the exponential decay weight, and the initial value of the trust-region penalty parameter λ.
- **DRL** (Meulemans et al., 2020): difference reconstruction loss. A TP variant that learns autoencoder-style inverses with a reconstruction loss specifically to make DTP recover Gauss-Newton targets. We use the same architecture as in (Meulemans et al., 2020) and report the published error.
- **TPRI** (Roulet & Harchaoui, 2021): compute targets using a product of layer-wise Tikhonov-regularized Jacobian inversesl weights are updated by SGD on the standard local reconstruction loss. We tune the regularization strength, batch size, learning rate, and momentum.
- **TJML** (Bao et al., 2024): a Jacobian matching loss yields closed-form approximate inverses (transposed weights for linear layers, analytically derived dual activation-function). Closed-form inverse introduces no hyperparameters; we tune the batch size, learning rate, and momentum,

Table 2 reports final test error and iteration to milestone for each run, where each method is compared at its own tuned batch size, aggregated over three runs. Note that KFAC and ADAM tend to favour smaller batch sizes, whereas CGTP favours larger ones (as per-sample curvature is only informative when the batch size is large enough to estimate it). Figure 1 shows BP, KFAC, and CGTP at matched batch size, where the order is preserved but the milestone margins change. More results (such as depth and parameter sensitivity analysis) can be found in Supp D, and more detailed discussions of Related Works can be found in Supp A.

6 Conclusion

We show that the correspondence between target propagation and Gauss–Newton optimization, previously established only where hitting the targets is trivial, extends to the general case once realization is treated as a trust-region problem in the targets' own metric. Classical TP is a single gradient step under this metric; solving it by conjugate gradient yields CGTP, and sharing curvature across samples recovers KFAC's factored damping exactly. Layers are solved independently and in parallel after one backward pass. On MNIST and Fashion-MNIST MLPs, CGTP matches Adam and KFAC in accuracy with an order of magnitude fewer outer iterations. Our experiments are proof-of-concept, restricted to small MLPs; extending to the stochastic setting, structured architectures, and principled preconditioning for ill-conditioned problems such as PINNs are natural next steps.

AI TOOL DECLARATION

In this work, we used generative AI tools primarily for fine-tuning and validation:

- Derivations were submitted to Claude to check for errors, and fixes devised where they were raised. Examples abound: the indexing scheme of Lemma 5 was previously incorrect (the correction verified through a simple numpy script), and the previous definition of B_n in Section 3.2 only held for invertible layers, leading to the correction term in equation 52).
- Our code was profiled and optimised using Claude code tools: for example, the push through identity was adopted in several places, in addition to certain refactors that flagged bottlenecks introduce by logging, and permitted acceleration through use of pytorch compilation tools.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Wenbao Ai and Shuzhong Zhang. Strong duality for the cdt subproblem: A necessary and sufficient condition. *SIAM Journal on Optimization*, 19(4):1735–1756, 2009. doi: 10.1137/07070601X. URL https://doi.org/10.1137/07070601X.

Yanhao Bao, Tatsukichi Shibuya, Ikuro Sato, Rei Kawakami, and Nakamasa Inoue. Efficient target propagation by deriving analytical solution. In *AAAI Conference on Artificial Intelligence*, 2024. URL https://api.semanticscholar.org/CorpusID:268692707.

Sergey Bartunov, Adam Santoro, Blake Aaron Richards, Geoffrey E. Hinton, and Timothy P. Lillicrap. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. In *Neural Information Processing Systems*, 2018. URL https://api.semanticscholar.org/CorpusID:49670549.

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to imagenet. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:57189514.

Yoshua Bengio. Deriving differential target propagation from iterating approximate inverses. *CoRR*, abs/2007.15139, 2020. URL https://arxiv.org/abs/2007.15139.

Frederik Benzing. Gradient Descent on Neurons and its Link to Approximate Second-Order Optimization. *arXiv e-prints*, art. arXiv:2201.12250, January 2022. doi: 10.48550/arXiv.2201.12250.

Daniel Bienstock. A note on polynomial solvability of the CDT problem. *arXiv e-prints*, art. arXiv:1406.6429, June 2014. doi: 10.48550/arXiv.1406.6429.

Sheng-Wei Chen, Chun-Nan Chou, and Edward Chang. Bda-pch: Block-diagonal approximation of positive-curvature hessian for training neural networks. 02 2018. doi: 10.48550/arXiv.1802.06502.

Maxence Ernoult, Fabrice Normandin, A. Moudgil, Sean Spinney, Eugene Belilovsky, Irina Rish, Blake Aaron Richards, and Yoshua Bengio. Towards scaling difference target propagation by learning backprop targets. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:246430965.

Karl J. Friston and Stefan J. Kiebel. Predictive coding under the free-energy principle. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364:1211 – 1221, 2009. URL https://api.semanticscholar.org/CorpusID:13950127.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1842–1850. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/gupta18a.html.

Max Jaderberg, Wojciech M. Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. In *International Conference on Machine Learning*, 2016a. URL https://api.semanticscholar.org/CorpusID:5615093.

Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. *CoRR*, abs/1608.05343, 2016b. URL http://arxiv.org/abs/1608.05343.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Dong-Hyun Lee, Saizheng Zhang, Asja Fischer, and Yoshua Bengio. Difference target propagation. In *ECML/PKDD*, 2014. URL https://api.semanticscholar.org/CorpusID:14218948.

James Martens and Roger Baker Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, 2015. URL https://api.semanticscholar.org/CorpusID:11480464.

Alexander Meulemans, Francesco S. Carzaniga, Johan A. K. Suykens, João Sacramento, and Benjamin F. Grewe. A theoretical framework for target propagation. *ArXiv*, abs/2006.14331, 2020. URL https://api.semanticscholar.org/CorpusID:220055614.

Beren Millidge, Anil. K. Seth, and Christopher L. Buckley. Predictive coding: a theoretical and experimental review. *ArXiv*, abs/2107.12979, 2021. URL https://api.semanticscholar.org/CorpusID:236447532.

Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, New York, NY, 2. ed. edition, 2006. ISBN 978-0-387-30303-1. URL http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+502988711&sourceid=fbw_bibsonomy.

Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. In *Neural Information Processing Systems*, 2016. URL https://api.semanticscholar.org/CorpusID:2843914.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Vincent Roulet and Zaïd Harchaoui. Target propagation via regularized inversion. *Trans. Mach. Learn. Res.*, 2023, 2021. URL https://api.semanticscholar.org/CorpusID:244798709.

Tatsukichi Shibuya, Nakamasa Inoue, Rei Kawakami, and Ikuro Sato. Fixed-weight difference target propagation. *ArXiv*, abs/2212.10352, 2022. URL https://api.semanticscholar.org/CorpusID:254877538.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5, 2015. URL https://api.semanticscholar.org/CorpusID:5541663.

Yulin Wang, Zanlin Ni, Shiji Song, Le Yang, and Gao Huang. Revisiting locally supervised learning: an alternative to end-to-end training. *ArXiv*, abs/2101.10832, 2021. URL https://api.semanticscholar.org/CorpusID:231709464.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

SUPPLEMENTARY MATERIALS

A Related Work **12**

B What do the TP losses mean? **13**

C Proofs and Extended Remarks **13**

C.1 Proof of Theorem 2: Norm-Induced Layer Separation 13

C.2 Remark: Handling Non-Invertible Output Layers 15

C.3 Classical TP as Projected Gradient Descent 15

C.4 Proof of Lemma 7: Feasibility of stacked perturbations 16

C.5 Proof of Theorem 8: KKT Conditions for the Constrained Subproblem 16

C.6 Proof of Theorem 9: Simplicial Decomposition of the Global Subproblem 17

C.7 Proof of Theorem 10: Single-parameter coupling of the two trust regions 20

C.8 Proof of Corrollary 12: Exact recovery of KFAC 20

D Additional Experimental Details **21**

D.1 Settings 21

D.2 Compute Requirements 21

D.3 Simplex Schemes 22

D.4 Ablation Studies 23

E Algorithm Listings **25**

A RELATED WORK

Practical TP variants. Among backprop alternatives, the TP literature is comparatively sparse, in part due to the overhead historically associated with the method. The requirement of approximate inverses is restrictive: the canonical approach of training layerwise autoencoders introduces many iterations of feedback-weight training between forward updates, additional per-layer learning rates, and instability that worsens with depth (Bartunov et al., 2018). What's more, these approaches have weaker theoretical grounding, as it is difficult to characterize how close feedback functions are to the true Jacobian inverse. Much subsequent work has focused on eliminating or simplifying these feedback weights. (Meulemans et al., 2020) examines direct feedback pathways with similarities to DFA (Nøkland, 2016); FW-DTP (Shibuya et al., 2022) pushes this further, arguing that DTP regularizes strongly enough that even a fixed random feedback matrix suffices. Our work is closer in spirit to (Roulet & Harchaoui, 2021), which leverages autograd to compute the true Jacobian and employ a linearized version of Target Propagation. We extend this direction by embedding the inversion in a trust-region framework, separating the inverse's fidelity from its utility as a step-direction estimator. The inner-loop iterations our method incurs are spent solving a subproblem that approximates the true objective, rather than training feedback parameters that are discarded after training.

TP and Gauss-Newton. The connection between TP and Gauss-Newton was first made by (Bengio, 2020) and given more rigorous treatment by (Meulemans et al., 2020), whose proof we recount in Section 2.2. Subsequent work has largely retreated from the GN framing. (Roulet & Harchaoui, 2021) attempt to leverage the differential prescription but raise several objections to the proof in their appendix, ultimately focusing on RNN training and concluding that GN is unlikely to explain TP's empirical success. Other recent work abandons the GN connection entirely: LDRL (Ernoult

et al., 2022) aims to reproduce BP gradients directly, achieving competitive results on CIFAR-style benchmarks, while (Bao et al., 2024) derive closed-form feedback functions via a Jacobian-matching loss. Both rely on alignment with BP gradients - positive but imperfect - which limits convergence speed by construction. We take the GN connection more seriously: solving the local subproblem properly turns TP from a method that can approximate BP into one that can potentially outpace it.

Second-order optimization. A range of works have applied second-order machinery to training deep networks. Block-diagonal approximations of the curvature matrix are common, supported by empirical evidence (Martens & Grosse, 2015; Gupta et al., 2018; Chen et al., 2018), and our work is closest in spirit to KFAC (Martens & Grosse, 2015), which targets the Gauss-Newton approximation and uses Levenberg-Marquardt-style damping. Our derivation differs in its treatment of this block-diagonal structure: rather than assume it, we obtain it as an upper bound on the Gauss-Newton subproblem via Theorem 9. KFAC is itself derived as a special case of our setup in Corollary 12. A connection between TP and KFAC was previously suggested in (Benzing, 2022) based on superficial similarity in their appearance. Our work rigorously establishes this connection.

B What do the TP losses mean?

The conventional presentation of Section 2.1 is operationally complete: it specifies how to compute targets and how to update weights. This subsection examines several choices that the presentation leaves implicit, and that bear on the theoretical status of the local subproblem. We surface them because they motivate the framing adopted in subsequent sections (and in particular, the surrogate loss of Section 3.1).

First, we consider why Equation equation 2 makes for a sensible target. In backpropagation, the weights are the principal optimization variable. Under backpropagation, thee gradients are computed via the chain rule, and the weight update aggregates the per-sample loss gradients:

$$[\Delta W_{\text{BP}}^{(l)}]_{ij} = \sum_n \frac{\partial \mathcal{L}_n}{\partial W_{ij}^{(\ell)}}; \quad \frac{\partial \mathcal{L}_n}{\partial W_{ij}^{(\ell)}} = \Big(\sum_k \frac{\partial \mathcal{L}_n}{\partial z_k^{(\text{out})}} J_{ki}^{(\text{out},\ell+1)}(x_n) \Big) u_j^{(\ell)}(x_n). \tag{37}$$

Target propagation takes a different approach, promoting the intermediate pre-activations $z_n^{(\ell)}$ as principal. To this end, note that the per-sample gradient vanishes as $\|\partial \mathcal{L}_n / \partial z^{(\text{out})}\| \to 0$. A gradient descent step on the outputs $z^{(\text{out})}$, as in Equation 2, is one way to drive this.

With the output targets specified, we turn towards the hidden layers. As these are connected to the output by an invertible map, the natural choice for an intermediate target is the inverse image. We can make this intuition more rigorous as follows: with the downstream targets $z_n^{(out)}$ fixed, a natural loss function for $z_n^{(\ell)}$ is the following:

$$\mathcal{L}_n^{(\ell)} = ||F^{(out,\ell)}(z_n^{(\ell)}) - \hat{z}_n^{(out)}||^2 \tag{38}$$

Note, by comparison with equation 37, that $\nabla_{z^{(\ell)}} \mathcal{L}_n^{(l)} = \nabla_{z^{(\ell)}} \mathcal{L}_n$. If $F^{(out,\ell)}$ has a corresponding inverse $[F^{(out,\ell)}]^{-1}$, the above may be re-written:

$$\mathcal{L}_n^{(\ell)} = ||F^{(out,\ell)}(z_n^{(\ell)}) - F^{(out,\ell)} \circ [F^{(out,\ell)}]^{-1}(\hat{z}_n^{(out)})||^2 \tag{39}$$

If the composite function $F^{(L,\ell)}$ is $C^{(L,\ell)}$-Lipschitz, then we may bound the above loss as follows:

$$\mathcal{L}_n^{(\ell)} \leq C^{(L,\ell+1)} ||z_n^{(\ell)} - [F^{(out,\ell)}]^{-1}(\hat{z}_n^{(out)})||^2 \tag{40}$$

We recognize the norm on the right-hand side as $\beta \mathcal{L}_{\text{TP}}$ - implying that the local TP loss, for $\beta = 1$ is an upper bound on the BP loss for the targets.

C Proofs and Extended Remarks

C.1 Proof of Theorem 2: Norm-Induced Layer Separation

Recall the setup of the multi-layer subproblem from Section 3.2: let J_n denote the stacked Jacobian and Δz_n the stacked perturbation, as follows:

$$J_n \coloneqq \big[\, I \;\big|\; J_n^{(\text{out},L)} \;\big|\; \cdots \;\big|\; J_n^{(\text{out},1)} \,\big]; \quad \Delta z_n \coloneqq \big[\, \Delta z_n^{(\text{out})\top} \;\big|\; \Delta z_n^{(L)\top} \;\big|\; \cdots \;\big|\; \Delta z_n^{(1)\top} \,\big]^\top \tag{41}$$

The regularised, multi-layer subproblem is given by:

$$m_n(\Delta z_n;\lambda) = \left\|r_n + H_n J_n \Delta z_n\right\|^2_{M_n} + \tfrac{1}{2}\lambda \sum_{\ell=1}^{L+1} \left(\Delta z_n^{(\ell)}\right)^\top B_n^{(\ell)} \Delta z_n^{(\ell)} \tag{42}$$

In the derivation to come, we will require the following definition and lemma:

Definition 13 (S-Regularised Residual)**.** Given the residual $r_n \in \mathbb{R}^C$, output Hessian $H_n \in \mathbb{R}^{C\times C}$, output metric $M_n \in \mathbb{R}^{C\times C}$, Lagrange multiplier λ, and a choice of PD matrix $S_n \in \mathbb{R}^{C\times C}$, the S-regularised residual is:

$$\tilde{r}_n(\lambda; S_n) = \left(H_n^\top M_n H_n + \lambda S_n^{-1}\right)^{-1} H_n^\top M_n r \tag{43}$$

Lemma 14 (Properties of the full Jacobian)**.** *J_n is surjective; furthermore, for any $B_n \succ 0$, the matrix $S_n = J_n B_n^{-1} J_n^\top \in \mathbb{R}^{K\times K}$ is positive definite.*

Proof. For any $\Delta z_n \neq 0$, the first block of $J_n^\top \Delta z$ is simply $\Delta z_n^{(\text{out})}$, due to the identity matrix in 14. Hence, $J_n^\top \Delta z_n \neq 0$, and thus $\Delta z_n^\top S_n \Delta z_n = \|B_n^{-1/2} J_n^\top \Delta z_n\|^2 > 0$. □

We are now in a position to prove the theorem. Taking the gradient of 16 with respect to Δz_n, we obtain the normal equations:

$$H_n^\top M_n H_n J_n \Delta z_n^\star + \lambda B_n \Delta z_n^\star = -H_n^\top M_n r_n \tag{44}$$

Left multiplying by $J_n B_n^{-1}$:

$$S_n H_n^\top M_n H_n (J\Delta z_n^\star) + \lambda (J\Delta z_n^\star) = -S_n H_n^\top M_n r \tag{45}$$

Then, by Lemma14, S_n^{-1} exists, and we have (using Definition 13):

$$J_n \Delta z_n^\star = \tilde{r}_n(\lambda; S_n) \tag{46}$$

Accordingly, any choice of metric B_n is sufficient to separate the layers. However, in what follows, we will see that a diagonal B_n holds additional advantages. To this end, re-examine the normal equations in 44, and consider the ansatz $\Delta z_n^\star = -B_n^{-1} J_n^\top w$ for some $w \in \mathbb{R}^C$. Substituting into the left-hand-side of 44:

$$\left(J_n^\top H_n^\top M_n H_n J_n + \lambda B_n\right)\left(-B_n^{-1} J_n^\top w\right) = -J_n^\top H_n^\top M_n H_n S_n w - \lambda J_n^\top w \tag{47}$$

$$= -J_n^\top \left(H_n^\top M_n H_n + \lambda S_n^{-1}\right) S_n w \tag{48}$$

This equals the right side of 44 when$S_n w = \tilde{r}_n(\lambda; S_n)$. Therefore, an alternative (and by uniqueness, equivalent) expression for the minimiser is:

$$\Delta z_n^\star = -B_n^{-1} J_n^\top S_n^{-1} \tilde{r}_n(\lambda; S_n) \tag{49}$$

$J_n^\top S_n^{-1} \tilde{r}_n(\lambda; S_n)$ is the backpropagated regularised target, and B_n^{-1} acts as a preconditioner. Block diagonality of B_n ensures this feasible to compute. The computation of S_n also simplifies:

$$S = [B^{(\text{out})}]^{-1} + \sum_\ell C^{(\ell)} \qquad C^{(\ell)} = J^{(\ell)} [B^{(\ell)}]^{-1} J^{(\ell)\top} \succeq 0 \tag{50}$$

To proceed, observe that, if we consider perturbations $\Delta z_n \sim \mathcal{N}(0, B^{-1})$, the matrix $S_n = J_n B_n^{-1} J_n^\top$ represents the covariance of $J_n \Delta z_n$ - ie the pushforward of the prior covariance at the output. Thus, beyond numerical convenience, the choice $S = I$ suggests every target-space direction equally likely a priori, hence we refer to it as the **egalitarian prior**.

In the invertible case, it is clear that the following choice achieves $S = I$:

$$B_n^{(\ell)} = \frac{1}{\alpha_n^{(\ell)}} J_n^{(\ell)\top} J_n^{(\ell)}; \quad \alpha_n^{(\ell)} > 0,\ \sum_\ell \alpha_n^{(\ell)} = 1 \tag{51}$$

Any choice of $\alpha_n^{(\ell)}$ on the simplex will work. The TP literature commonly makes the uniform choice $\alpha_n^{(\ell)} = (L+1)^{-1}$. When the layers are not invertible, we include a small regularisation term as follows:

$$\tilde{B}^{(\ell)} = \frac{1}{\alpha_n^{(\ell)}} \left(J_n^{(\ell)\top} J_n^{(\ell)} + \epsilon I\right) \tag{52}$$

C.2 Remark: Handling Non-Invertible Output Layers

A standard deep neural network has a characteristic shape: let D denote the input dimension, K the dimension of the intermediate activations, and C the output dimension, with the common ordering $D > K \gg C$. The first layer is non-invertible (it maps D-dimensional inputs to K-dimensional activations), but this poses no difficulty for our derivation, since we never propagate corrective targets to the input. The output layer is the genuine concern: $W^{(\mathrm{out})} \in \mathbb{R}^{K\times C}$ collapses the K-dimensional activation space onto C outputs, with $K \gg C$ implying a large null space. The output map deletes substantial information about the activations, and we must specify how to handle this asymmetry.

Our strategy is to make the deleted information manifest. Decompose the output weight as $W^{(\mathrm{out})} = USV^\top$ via the singular value decomposition, with $S = [S_\parallel, 0]$ partitioned into the active singular values and the trailing zero block. The columns of V split correspondingly into $V = [V_\parallel, V_\perp]$: components of the input lying in $\mathrm{col}(V_\parallel)$ are mapped to the output, while components in $\mathrm{col}(V_\perp)$ are mapped to zero. The latter are precisely the directions in activation space that the output map deletes.

We make these directions explicit by augmenting the forward map:

$$\begin{pmatrix} z_n^{(\mathrm{out})} \\ c_n \end{pmatrix} = \begin{pmatrix} US_\parallel & 0 \\ 0 & I \end{pmatrix} V^\top u_n^{(L)} + ob^\top, \qquad o = \begin{pmatrix} 1_\parallel \\ 0 \end{pmatrix}, \tag{53}$$

where c_n records the components that the original output map discards, and o is a mask that suppresses the bias on the augmented coordinates. The augmented map is invertible by construction: V is orthogonal, the upper block $US_\parallel$ is the original output map, and the lower-block identity preserves the previously-deleted directions.

The corresponding inverse problem for the activation perturbation is then

$$\begin{pmatrix} \Delta z_n^{(\mathrm{out})} \\ \Delta c_n^{(\mathrm{out})} \end{pmatrix} = \begin{pmatrix} US_\parallel & 0 \\ 0 & I \end{pmatrix} V^\top \Delta u_n^{(L)} + ob^\top. \tag{54}$$

The solution to this inverse problem is

$$\Delta u_n^{(L)} = W^+\big(\Delta z_n^{(\mathrm{out})} - b\big) + V_\perp \Delta c_n^{(\mathrm{out})}, \tag{55}$$

where W^+ denotes the pseudoinverse of $W^{(\mathrm{out})}$. Accommodating non-invertible layers thus reduces to specifying $\Delta c_n^{(\mathrm{out})}$ - a prescription for how to update the components of the activation that the loss does not directly constrain. We adopt the simplest choice, $\Delta c_n^{(\mathrm{out})} = 0$, equivalent to using the pseudoinverse of the original (unaugmented) forward map: in directions where the loss offers no corrective feedback, leave the activation unchanged.

C.3 Classical TP as Projected Gradient Descent

Differentiating the per-datapoint reconstruction loss with respect to $W^{(\ell)}$ gives

$$\frac{\partial}{\partial W^{(\ell)}} \big\| W^{(\ell)\top} u_n^{(\ell)} + b^{(\ell)} - \hat{z}_n^{(\ell+1)} \big\|^2 \;=\; -2\, u_n^{(\ell)} \big(\hat{z}_n^{(\ell+1)} - W^{(\ell)\top} u_n^{(\ell)} - b^{(\ell)}\big)^\top. \tag{56}$$

Since $z_n^{(\ell+1)} = W^{(\ell)\top} u_n^{(\ell)} + b^{(\ell)}$ and $\hat{z}_n^{(\ell+1)} = z_n^{(\ell+1)} + \Delta z_n^{(\ell+1)}$, the residual collapses to $\Delta z_n^{(\ell+1)}$. Summing over n and folding the bias into $\tilde{u}$ yields, in matrix form,

$$\Delta W_{\mathrm{TP}}^{(\ell)} = 2\, \tilde{U}^{(\ell)\top} \Delta Z^{(\ell+1)}. \tag{57}$$

The induced pre-activation perturbation is

$$\Delta \bar{Z}_{\mathrm{PGD}}^{(\ell+1)} = \eta\, \tilde{U}^{(\ell)} \Delta W_{\mathrm{TP}}^{(\ell)} = 2\eta\, \tilde{U}^{(\ell)} \tilde{U}^{(\ell)\top} \Delta Z^{(\ell+1)}, \tag{58}$$

which lies in $\mathrm{col}(\tilde{U}^{(\ell)})$ and therefore satisfies $\Pi^{(\ell)} \Delta \bar{Z}_{\mathrm{PGD}}^{(\ell+1)} = 0$ by Lemma 5.

C.4 Proof of Lemma 7: Feasibility of stacked perturbations

For each datapoint, stack the perturbations of all layers,

$$\Delta z_n = \big(\Delta z_n^{(L+1)}, \Delta z_n^{(L)}, \dots, \Delta z_n^{(1)}\big) \in \mathbb{R}^D, \qquad D = \sum_\ell K_\ell, \tag{59}$$

and stack these across the batch, $\Delta \boldsymbol{z} = (\Delta z_1, \dots, \Delta z_N) \in \mathbb{R}^{ND}$. In this ordering the curvature is block-diagonal, $\boldsymbol{G} = \bigoplus_n G_n$ with $G_n = J_n^\top H_n^\top M_n H_n J_n$. Let $E_\ell \in \mathbb{R}^{K_\ell \times D}$ select the block of layer ℓ, so that $\Delta z_n^{(\ell)} = E_\ell \Delta z_n$ and $\mathrm{vec}_r(\Delta Z^{(\ell)}) = (I_N \otimes E_\ell)\Delta \boldsymbol{z}$. The constraint of Lemma 5 for layer ℓ then reads

$$\big(\Pi^{(\ell-1)} \otimes I_{K_\ell}\big)\big(I_N \otimes E_\ell\big)\Delta \boldsymbol{z} = \big(\Pi^{(\ell-1)} \otimes E_\ell\big)\Delta \boldsymbol{z} = 0. \tag{60}$$

Since the E_ℓ select disjoint blocks, $E_\ell E_{\ell'}^\top = 0$ for $\ell \neq \ell'$ and $\sum_\ell E_\ell^\top E_\ell = I_D$, the operators $\Pi^{(\ell-1)} \otimes E_\ell^\top E_\ell$ are mutually orthogonal projectors, and their sum

$$\boldsymbol{\Pi} = \sum_\ell \Pi^{(\ell-1)} \otimes E_\ell^\top E_\ell \tag{61}$$

is itself an orthogonal projector on $\mathbb{R}^{ND}$. The constraints for all layers are therefore equivalent to the single condition $\boldsymbol{\Pi}\,\Delta \boldsymbol{z} = 0$.

C.5 Proof of Theorem 8: KKT Conditions for the Constrained Subproblem

This appendix collects the KKT analysis of constrained quadratic minimization with linear equality constraints, and applies it to the multi-layer subproblem of Theorem 8. The generic result is standard (Nocedal & Wright, 2006); we state and prove it here for completeness, since it is invoked at several points in the main text.

Lemma 15 (KKT solution for equality-constrained quadratics)**.** *Let $H \in \mathbb{R}^{m\times m}$ be symmetric positive definite, $D \in \mathbb{R}^{q\times m}$ have full row rank, and $x^\star \in \mathbb{R}^m$ be given. The unique minimizer of*

$$\min_{x\in\mathbb{R}^m} (x - x^\star)^\top H (x - x^\star) \quad \textit{subject to} \quad Dx = 0 \tag{62}$$

is

$$x = (I - H^{-1}D^\top \big(DH^{-1}D^\top\big)^{-1} D)x^\star. \tag{63}$$

The matrix $DH^{-1}D^\top$ is the Schur complement of the KKT system; it is invertible under the stated assumptions.

Proof. The Lagrangian of equation 62 is

$$\mathcal{L}(x, \nu) = (x - x^\star)^\top H (x - x^\star) + \nu^\top Dx.$$

Stationarity with respect to x gives

$$2H(x - x^\star) + D^\top \nu = 0 \implies x = x^\star - \tfrac{1}{2} H^{-1} D^\top \nu.$$

Substituting into the feasibility constraint $Dx = 0$:

$$Dx^\star - \tfrac{1}{2} DH^{-1}D^\top \nu = 0 \implies \nu = 2\big(DH^{-1}D^\top\big)^{-1} Dx^\star,$$

where invertibility of $DH^{-1}D^\top$ follows from $H \succ 0$ and D having full row rank. Substituting back yields equation 63. Uniqueness follows from strict convexity of the objective on the affine feasible set. □

Remark 16 (Geometric interpretation). The right-hand side of equation 63 is the projection of $x^\star$ onto $\ker D$ in the inner product $\langle u, v\rangle_H = u^\top H v$. When $H = I$, this reduces to the standard Euclidean projection $x = x^\star - D^\top (DD^\top)^{-1} Dx^\star$. For $H \neq I$, the projection is in the metric induced by H, and is the unique minimizer of the H-weighted distance to $x^\star$ over the feasible set.

C.6 Proof of Theorem 9: Simplicial Decomposition of the Global Subproblem

Motivation: Recall the unregularised multi-layer subproblem from 15:

$$m_n(\Delta z_n) = \left\| r_n + H_n J_n \Delta z_n \right\|_{M_n}^2 \tag{64}$$

Let α_n be a set of layer-weight which lie in the interior of an $(L+1)$ simplex, such that:

$$\forall \ell \qquad \alpha_n^{(\ell)} > 0, \qquad \sum_{\ell=1}^{L+1} \alpha_n^{(\ell)} = 1 \tag{65}$$

It's clear from substitution that any layer perturbations $\Delta z_n^{(\ell)}$ satisfying

$$H_n J_n^{(\ell)} \Delta z_n^{(\ell)}(\alpha_n^{(\ell)}) = -\alpha_n^{(\ell)} r_n, \qquad \ell = 1, \dots, L+1, \tag{66}$$

will reduce m_n to 0. The difficulty lies when we have non-zero λ and we minimize the $m_n(\Delta z_n; \lambda)$ of 2. As we will see, this will complicate the relationship between $\alpha_n^{(\ell)}$ and $\Delta z_n^{(\ell)}$.

Lemma 17 (A Majoriser on the GN Subproblem)**.** *Let $\alpha_n^{(\ell)}$ lie on the simplex. Then the following is a majoriser of the un-regularised subproblem in Theorem 2:*

$$\hat{m}_n(\Delta z_n; \alpha_n) = \sum_{\ell} \left\| \sqrt{\alpha_n^{(\ell)}}\, r_n + \tfrac{1}{\sqrt{\alpha_n^{(\ell)}}} H_n J_n^{(\ell)} \Delta z_n^{(\ell)} \right\|_{M_n}^2 \tag{67}$$

Bounded above Because $\sum_\ell \alpha_n^{(\ell)} = 1$, we may write $r_n = \sum_\ell \alpha^{(\ell)} r_n$ and regroup the argument of the norm as a convex combination:

$$r_n + \sum_{\ell=1}^{L+1} H_n J_n^{(\ell)} \Delta z_n^{(\ell)} = \sum_{\ell=1}^{L+1} \alpha_n^{(\ell)} \left(r_n + (\alpha_n^{(\ell)})^{-1} H_n J_n^{(\ell)} \Delta z_n^{(\ell)} \right) \tag{68}$$

The map $v \mapsto \|v\|_{M_n}^2$ is convex (and strictly so when $M \succ 0$) so we may apply Jensen's inequality as follows:

$$m(\Delta z) \le \frac{1}{2} \sum_{\ell=1}^{L+1} \alpha_n^{(\ell)} \left\| r_n + (\alpha_n^{(\ell)})^{-1} H_n J_n^{(\ell)} \Delta z_n^{(\ell)} \right\|_{M_n}^2 \coloneqq \frac{1}{2} \sum_{\ell=1}^{L+1} \hat{m}_n^{(\ell)}(\Delta z_n^{(\ell)}; \alpha_n^{(\ell)}) \tag{69}$$

Thus, the coupling is removed. Expanding each term:

$$\frac{\alpha_n^{(\ell)}}{2} \left\| r_n + (\alpha_n^{(\ell)})^{-1} H_n J_n^{(\ell)} \Delta z_n^{(\ell)} \right\|_{M_n}^2 \equiv \frac{\alpha_n^{(\ell)}}{2} \|r\|_{M_n}^2 + g_n^{(\ell)\top} \Delta z_n^{(\ell)} + \frac{1}{2\alpha_n^{(\ell)}} \Delta z_n^{(\ell)\top} G_n^{(\ell)} \Delta z_n^{(\ell)} \tag{70}$$

Then, summing over ℓ, using the fact that $\sum_\ell \alpha^{(\ell)} = 1$:

$$\hat{m}_n(\Delta z_n; \alpha_n) = \frac{1}{2}\|r\|_{M_n}^2 + \sum_{\ell=1}^{L+1} \left[g_n^{(\ell)\top} \Delta z_n^{(\ell)} + \frac{1}{2\alpha^{(\ell)}} \Delta z_n^{(\ell)\top} G_n^{(\ell)}(\lambda) \Delta z_n^{(\ell)} \right] \tag{71}$$

$$\ge m(\Delta z) \tag{72}$$

Tangency in value

$$\hat{m}_n(0; \alpha_n) = \frac{1}{2} \sum_{\ell=1}^{L+1} \alpha_n^{(\ell)} \|r\|_{M_n}^2 = \frac{1}{2}\|r\|_{M_n}^2 = m(0) \tag{73}$$

Tangency in gradient Differentiating the ℓ-th term of $\hat{m}_n(\Delta z_n; \alpha_n)$:

$$\nabla_{z_n^{(\ell)}} \hat{m}_n(\Delta z_n; \alpha_n) = \alpha_n^{(\ell)} \cdot (\alpha_n^{(\ell)})^{-1} J_n^{(\ell)\top} H_n^\top M_n \left(r_n + (\alpha_n^{(\ell)})^{-1} H_n J_n^{(\ell)} \Delta z_n^{(\ell)} \right) \tag{74}$$

$$= g_n^{(\ell)} + \frac{1}{\alpha_n^{(\ell)}} G_n^{(\ell)} \Delta z_n^{(\ell)} \tag{75}$$

At the origin, this is simply:

$$\nabla_{z_n^{(\ell)}} \hat{m}_n(0; \alpha_n) = g_n^{(\ell)} = \nabla_{z_n^{(\ell)}} m_n(0) \tag{76}$$

□

Now that the layers may be treated separately, we introduce the trust-region framework through a Levenberg-Marquardt penalty. This leads to the following definition:

Definition 18 (Regularised Layer-Local Subproblem)**.** Given the layer weight $\alpha_n^{(\ell)}$, Lagrange multiplier λ, and the block-diagonal metric B_n of Theorem 2, the layer-local subproblem is:

$$Q_n^{(\ell)}(\Delta z_n^{(\ell)}; \lambda, \alpha_n^{(\ell)}) = \hat{m}_n^{(\ell)}(\Delta z_n^{(\ell)}; \alpha_n^{(\ell)}) + \tfrac{\lambda}{2} \Delta z_n^{(\ell)\top} B_n^{(\ell)} \Delta z_n^{(\ell)} \tag{77}$$

By completing the square in the above definition, we may read off additional objects of note:

$$Q_n^{(\ell)}(\Delta z_n^{(\ell)}; \lambda, \alpha_n^{(\ell)}) = \Phi_n^{(\ell)}(\lambda, \alpha_n^{(\ell)}) + \tfrac{1}{2} ||\Delta z_n^{(\ell)} + \Delta z_n^{(\ell)\star}||^2_{\tilde{G}_n^{(\ell)}(\lambda, \alpha_n^{(\ell)})} \tag{78}$$

$$\tilde{G}_n^{(\ell)}(\lambda, \alpha_n^{(\ell)}) \coloneqq \frac{1}{\alpha_n^{(\ell)}} G_n^{(\ell)} + \lambda B_n^{(\ell)} \qquad \text{Curvature} \tag{79}$$

$$\Delta z_n^{(\ell)\star}(\lambda, \alpha_n^{(\ell)}) \coloneqq -[\tilde{G}_n^{(\ell)}(\lambda, \alpha_n^{(\ell)})]^{-1} g_n^{(\ell)} \qquad \text{Minimiser} \tag{80}$$

$$\Phi_n^{(\ell)}(\lambda, \alpha_n^{(\ell)}) \coloneqq \alpha_n^{(\ell)} ||r||^2_{M_n} - g_n^{(\ell)\top} [\tilde{G}_n^{(\ell)}(\lambda, \alpha_n^{(\ell)})]^{-1} g_n^{(\ell)} \qquad \text{Residual} \tag{81}$$

Remark 19 (Interpretation of the $\alpha_n^{(\ell)}$). : The layer weights enable us to distribute a single trust-region parameter λ across the different layers and datapoints.

The residual $\Phi_n^{(\ell)}(\lambda, \alpha_n^{(\ell)})$ holds particular relevance: it is error that remains unaccounted for by the minimiser $\Delta z_n^{(\ell)\star}(\lambda, \alpha_n^{(\ell)})$, and so the optimal perturbations will be found by choosing $\alpha_n^{(\ell)}$ in the simplex that minimise $\Phi(\lambda, \alpha) = \sum_\ell \Phi_n^{(\ell)}(\lambda, \alpha_n^{(\ell)})$. Unfortunately, the relationship between $\alpha_n^{(\ell)}$ and $\Delta z_n^{(\ell)\star}(\lambda, \alpha_n^{(\ell)}$ is too complex to find a closed-form solution, and so we must resort to iterative methods. To this end, we prove the following lemma:

Lemma 20 (Convexity of the Regularised Layer-Local Subproblem)**.** *For $\lambda > 0$, $B_n^{(\ell)} \succ 0$, and $G_n^{(\ell)} \succeq 0$, $Q_n^{(\ell)}(\Delta z_n^{(\ell)}; \lambda, \alpha_n^{(\ell)})$ is jointly convex in $(\Delta z_n^{(\ell)}, \alpha_n^{(\ell)})$ on $\mathbb{R}^K \times (0, \infty)$.*

Proof. Proof. From 70, Q is a sum over three terms:

$$\text{(i)} \quad : \quad \frac{\alpha_n^{(\ell)}}{2} \|r\|^2_{M_n} + g_n^{(\ell)\top} \Delta z_n^{(\ell)} \tag{82}$$

$$\text{(ii)} \quad : \quad \frac{1}{2\alpha_n^{(\ell)}} \Delta z_n^{(\ell)\top} G_n^{(\ell)} \Delta z_n^{(\ell)} \tag{83}$$

$$\text{(iii)} \quad : \quad \frac{\lambda}{2} \Delta z_n^{(\ell)\top} B_n^{(\ell)} \Delta z_n^{(\ell)} \tag{84}$$

Each term is jointly convex in $(\Delta z_n^{(\ell)}, \alpha_n^{(\ell)})$:

- **(i)** is affine.
- **(ii)** For $\alpha_n^{(\ell)} > 0$,

$$\frac{1}{2\alpha_n^{(\ell)}} \Delta z_n^{(\ell)\top} G_n^{(\ell)} \Delta z_n^{(\ell)} = \sup_y \Big(y^\top (G_n^{(\ell)})^{1/2} \Delta z_n^{(\ell)} - \tfrac{1}{2} \alpha_n^{(\ell)}\, y^\top y \Big), \tag{85}$$

  which is a pointwise supremum of functions affine in $(\Delta z_n^{(\ell)}, \alpha_n^{(\ell)})$.

- **(iii)** is a convex quadratic in $\Delta z_n^{(\ell)}$ alone, since $B_n^{(\ell)} \succ 0$.

Thus, since the sum of jointly convex functions is jointly convex, Q is jointly convex. □

The joint convexity of $Q_n^{(\ell)}(\Delta z_n^{(\ell)}; \lambda, \alpha_n^{(\ell)})$ suggests that a simple alternating minimisation scheme can be used to find the optimal α_n and $\Delta z_n^{(\ell)}$. The optimal $\Delta z_n^{(\ell)}$ for fixed $\alpha_n^{(\ell)}$ may be read off from the completed-square in 80. For fixed $\Delta z_n^{(\ell)}$, we make use of the following Lemma:

Lemma 21 (Optimal layer weights given fixed pre-activation perturbations)**.** *The optimal $\alpha_n^{(\ell)}$ as a function of $\Delta z_n^{(\ell)}$ is given by:*

$$\alpha_n^{(\ell)\star}(\Delta z_n) = \frac{\|\Delta z_n^{(\ell)}\|_{G_n^{(\ell)}}}{\sum_k \|\Delta z_n^k\|_{G_n^{(k)}}} \tag{86}$$

Proof. Let μ be a Lagrange multiplier enforcing $\sum_\ell \alpha_n^{(\ell)} = 1$, and consider the Lagrangian:

$$L_\alpha = Q_n^{(\ell)}(\Delta z_n^{(\ell)}; \lambda, \alpha_n^{(\ell)}) + \mu\left(\sum_\ell \alpha_n^{(\ell)} - 1\right) \tag{87}$$

Taking the derivative with respect to $\alpha_n^{(\ell)}$, we have:

$$\frac{\partial L_\alpha}{\partial \alpha_n^{(\ell)}} = -\frac{1}{2(\alpha_n^{(\ell)})^2} \Delta z_n^{(\ell)\top} G_n^{(\ell)} \Delta z_n^{(\ell)} + \mu = 0 \tag{88}$$

Thus:

$$\alpha_n^{(\ell)} = \frac{1}{\sqrt{2\mu}} ||\Delta z_n^{(\ell)}||_{G_n^{(\ell)}} \tag{89}$$

Now, performing the sum over ℓ:

$$\frac{1}{\sqrt{2\mu}} \sum_\ell ||\Delta z_n^{(\ell)}||_{G_n^{(\ell)}} = 1 \tag{90}$$

$$\implies \sqrt{2\mu} = \sum_\ell ||\Delta z_n^{(\ell)}||_{G_n^{(\ell)}} \tag{91}$$

Hence, finally:

$$\alpha_n^{(\ell)}(\Delta z) = \frac{||\Delta z_n^{(\ell)}||_{G_n^{(\ell)}}}{\sum_k ||\Delta z_n^{(k)}||_{G_n^{(k)}}} \tag{92}$$

□

We now prove that alternating minimisation can solve the subproblem:

Lemma 22 (Convergence of Alternating Minimisation)**.** *The sequence below converges to the minimum of $Q_n(\Delta z_n; \lambda, \alpha_n)$:*

$$\textit{(A)} \quad \Delta z_n^{(\ell)} \leftarrow -\alpha_n^{(\ell)} \big(G_n^{(\ell)} + \alpha_n^{(\ell)} \lambda B^{(\ell)}\big)^{-1} g_n^{(\ell)}, \qquad \textit{(B)} \quad \alpha_n^{(\ell)} \leftarrow \frac{\|\Delta z_n^{(\ell)}\|_{G_n^{(\ell)}}}{\sum_k \|\Delta z_n^{(k)}\|_{G_n^{(k)}}} \tag{93}$$

Proof. Step (A) is the exact minimiser of $Q_n(\Delta z_n; \lambda, \alpha_n)$: over Δz_n for fixed α_n . Step (B) is the exact minimizer of $Q_n(\Delta z_n; \lambda, \alpha_n)$: over α_n in the simplex for fixed Δz_n. Denoting $\Delta z_n(t), \alpha_n(t)$ the t-th iterates, the algorithm is monotonically non-increasing.

$$Q_n(\Delta z_n(t+1); \lambda, \alpha_n(t+1)) \le Q_n(\Delta z_n(t+1); \lambda, \alpha_n(t)) \le Q_n(\Delta z_n(t); \lambda, \alpha_n(t)) \tag{94}$$

The sequence is bounded below by $\min_{\Delta z_n} m(\Delta z_n) \ge 0$, and so it converges. □

C.7 Proof of Theorem 10: Single-parameter coupling of the two trust regions

Throughout, $\langle X, Y\rangle = \operatorname{tr}(X^\top Y)$, $u_n^\top$ are the rows of $\tilde{U}$, and we write $\Delta z_n = \Delta W^\top u_n$ for the pre-activation change induced on datapoint n.

Inscription. Both q_z and q_w are nonnegative. If $q_z(\Delta W) + q_w(\Delta W) \le 1$, then $q_z(\Delta W) \le 1 - q_w(\Delta W) \le 1$, and likewise $q_w(\Delta W) \le 1$, so $\{q_z + q_w \le 1\} \subseteq \{q_z \le 1\} \cap \{q_w \le 1\}$. Minimising the local objective over the inscribed ellipsoid is a convex problem in which $\Delta W = 0$ is strictly feasible; by Slater's condition strong duality holds, and its Lagrangian with multiplier $\lambda/2$ is, up to a constant, the penalised objective L_λ.

The penalty. By the definitions in equation 29 and equation 30,

$$\frac{\lambda}{2}\big(q_z + q_w\big) = \frac{\mu_z}{2}\|\tilde{U}\Delta W\|_F^2 + \frac{\mu_J}{2}\|\Delta W \bar{\tilde{G}}^{1/2}\|_F^2, \tag{95}$$

which gives the first line of equation 31.

Completing the square. Let $g_n = J_n^\top H_n^\top M_n r_n$. Expanding the n-th summand and adding its share $\frac{\mu_z}{2}\|\Delta z_n\|^2$ of the pre-activation penalty,

$$\frac{1}{2}\Big\|\sqrt{\alpha_n}\, r_n + \frac{1}{\sqrt{\alpha_n}} J_n \Delta z_n\Big\|_{M_n}^2 + \frac{\mu_z}{2}\|\Delta z_n\|^2 = \frac{\alpha_n}{2} r_n^\top M_n r_n + g_n^\top \Delta z_n + \frac{1}{2}\Delta z_n^\top \tilde{G}_n \Delta z_n, \tag{96}$$

since the cross term carries the factor $2\sqrt{\alpha_n}\alpha_n^{-1/2} = 2$. This quadratic is minimised at $\Delta z_n^\star = -\tilde{G}_n^{-1} g_n$, and because $\tilde{G}_n \Delta z_n^\star = -g_n$,

$$g_n^\top \Delta z_n + \frac{1}{2}\Delta z_n^\top \tilde{G}_n \Delta z_n = \frac{1}{2}\big(\Delta z_n - \Delta z_n^\star\big)^\top \tilde{G}_n \big(\Delta z_n - \Delta z_n^\star\big) - \frac{1}{2}\Delta z_n^{\star\top} \tilde{G}_n \Delta z_n^\star. \tag{97}$$

Summing over n and collecting the terms independent of ΔW into the constant gives the second line of equation 31.

Stationarity. Write $e_n = \Delta W^\top u_n - \Delta z_n^\star$. Perturbing $\Delta W \to \Delta W + V$ changes e_n by $V^\top u_n$, so the first-order change in $\frac{1}{2}\sum_n e_n^\top \tilde{G}_n e_n$ is $\langle V, \sum_n u_n e_n^\top \tilde{G}_n\rangle$. By symmetry of $\bar{\tilde{G}}$, the first-order change in $\frac{\mu_J}{2}\operatorname{tr}(\Delta W^\top \Delta W \bar{\tilde{G}})$ is $\langle V, \mu_J \Delta W \bar{\tilde{G}}\rangle$. Setting the gradient to zero,

$$\sum_n u_n u_n^\top \Delta W \tilde{G}_n + \mu_J \Delta W \bar{\tilde{G}} = \sum_n u_n \Delta z_n^{\star\top} \tilde{G}_n = -\sum_n u_n g_n^\top, \tag{98}$$

where the last equality uses $\tilde{G}_n \Delta z_n^\star = -g_n$ and the symmetry of $\tilde{G}_n$. This is equation 33.

Uniqueness. For any V,

$$\langle V, \mathcal{G}_\lambda(V)\rangle = \sum_n (V^\top u_n)^\top \tilde{G}_n (V^\top u_n) + \mu_J \operatorname{tr}(V^\top V \bar{\tilde{G}}) \ge \mu_J \mu_z \|V\|_F^2, \tag{99}$$

since $\tilde{G}_n \succeq 0$ and $\bar{\tilde{G}} = \bar{G} + \mu_z I \succeq \mu_z I$. The same expression is symmetric in its two arguments, so $\mathcal{G}_\lambda$ is symmetric positive definite in the Frobenius inner product. Hence L_λ is strictly convex, and its unique minimiser is the unique solution of equation 33.

C.8 Proof of Corrollary 12: Exact recovery of KFAC

Substituting $\tilde{G}_n = \bar{\tilde{G}}$ into equation 33,

$$\mathcal{G}_\lambda(V) = \Big(\sum_n u_n u_n^\top\Big) V \bar{\tilde{G}} + \mu_J V \bar{\tilde{G}} = \big(A + \mu_J I\big) V \bar{\tilde{G}}. \tag{100}$$

Both factors are symmetric positive definite, so the system $(A + \mu_J I)\Delta W \bar{\tilde{G}} = -\nabla_W L$ is solved by left-multiplying by $(A + \mu_J I)^{-1}$ and right-multiplying by $\bar{\tilde{G}}^{-1}$. The right-hand side is unchanged by the substitution, since it is the true gradient for any choice of per-sample metric.

D Additional Experimental Details

D.1 Settings

In this section, we recount additional experimental details that were not described in the main body of the paper. Our models are implemented using PyTorch (Paszke et al., 2019), and trained using NVIDIA RTX A6000 48GB GPUs.

- GDTP, EucTP, CGTP (no WD), and CGTP all make use of trust regions. To this end we use the common, default values: a stringency of 0.15, acceptance/success thresholds of 0.25 and 0.75, a growth factor of 4 and a shrink factor of 0.25. The regularization strength is clipped to lie within 1e-5 and 1024.
- Each of the trust regions methods take, as subproblem norm, $M_n = H_n^{-1}$. For softmax cross-entropy classification problems (like those considered in the experiments), H_n is not invertible. We combat this by adding a small, constant penalty on the magnitude of the model outputs, set to $\sigma = 0.001$, which amounts to Tikhonov regularisation $H_n \to H_n + \sigma I$.
- Choice of $B_n^{(\ell}$ for CGTP: motivated by the connection with KFAC, and the scale invariant diagonal damping of classical Levenberg-Marquardt (Nocedal & Wright, 2006), we choose $B_n^{(\ell)} = \mathrm{diag}(\mathbb{E}[J^{(\ell)\top} J^{(\ell)\top}])$.
- We perform an ablation study (Table 4) incorporating preconditioners into our CG algorithm. Preconditioners are desirable because they can enable faster convergence, enabling us to get competitive results with a smaller number in cg iterations. We precondition with the averaged operator $\mathcal{P} = (A + \mu_J I) \otimes \bar{\bar{G}}$ rather than a per-sample alternative. Beyond its Kronecker structure, this choice is the one consistent with the null-space elimination of Section 4.3 as it commutes with with the constraint operator. The general preconditioned CG algorithm is shared in Algorithm 1. Ultimately, however, we find that using preconditioners results in worse performance, and the gap is related to the choice of simplex scheme (Supp D.3).
- KFAC here is our own implementation. Per Corrollary 12, KFAC is exactly recovered from CGTP when the per-sample Jacobians are replaced by their mean. To ensure a fair comparison, we use the same trust-region adapation as CGTP rather than periodic-inversion schedule of (Martens & Grosse, 2015). Similarly, we invert every step. This is more expensive per-step than a scheduled inversion but negligible relative to a network backward pass.
- All methods were trained were given a maximum of 15 epochs to train. We reserve 10% of the training set as a validation set. The test errors are reported at the epoch with the best validation error.
- Hyperparameters are tuned via Bayesian optimization on `wandb` against validation accuracy, with three seeds per configuration; we report the test error of the configuration with the best validation accuracy.

D.2 Compute Requirements

Method	Batch size	Peak VRAM (mb)	FLOPs / step	Total Wall Clock (10 epochs)
BP+ADAM	0.01	103.49	1.08 GFLOP	6m 11s
KFAC	0.1	660.40	93.8 GFLOP	6m 24s
CGTP	0.1	3210.10	1.16 TFLOP	6m 47s

Table 3: Resource comparison between BP+ADAM and CGTP

While we emphasise that our implementation is an unoptimised proof-of-concept, we include runtime statistics in Table 3. Note that, while each CGTP batch is more expensive, because far fewer outer iterations are needed, total time-to-milestone is comparable. Additionally, other factors already favour our approach, e.g. reduced host-device data movement, since a batch is loaded to the GPU once and reused across all inner-loop iterations. Finally, our proof-of-concept solves layer subproblems sequentially. Realising parallelism would further reduce time-per-batch.

D.3 SIMPLEX SCHEMES

Theorem 9 describes how the multi-layer subproblem may be decomposed into non-interacting parts, and how alternating minimisation to find the optimal set of layer weights that maximises the expected decrease. Unfortunately, while updating Δz_n for fixed α_n can be done locally, updating the α_n requires a global syncronisation step across layers. To this end, we perform an aggressively truncated alternating minimisation sequence: we initialise Δz_n according to one of the strategies shared below, and then use this to take one α-step. These are the alphas that then enter into our CG solver, and an ablation is shown in Table 4.

The strategies are as follows:

- **Uniform**: No α-step. We simply initialise $\alpha_n^{(\ell)} = (L+1)^{-1}$.
- **Unit** Δz: We consider $||\Delta z_n^{(\ell)}|| = 1$, and approximate $||\Delta z_n^{(\ell)}||_{G_n^{(\ell)}} \approx ||G_n^{(\ell)}||_F$.
- **Large** λ: In the limit of large λ, the regulariser $B_n^{(\ell)}$ dominates, so that $\Delta z_n^{(\ell)} \approx -\frac{1}{\lambda}[B_n^{(\ell)}]^{-1} g_n^{(\ell)}$.

Table 4: Numerical comparison of the test error and number of iterations to reach a given test error for the different simplex schemes. Hyperparameters for each algorithm were optimised for validation error.

Preconditionr	Simplex Scheme	Test Score	milestone@0.13
False	Large λ	0.1085 ± 0.0016	6.6667 ± 0.5774
False	Uniform	0.1126 ± 0.0019	9.6667 ± 2.0817
False	Unit $\|\|\Delta z\|\|$	0.1107 ± 0.0008	12.6667 ± 2.0817
True	Large λ	0.1108 ± 0.0014	7.6667 ± 0.5774
True	Uniform	0.1240 ± 0.0027	9.0000 ± 1.7321
True	Unit $\|\|\Delta z\|\|$	0.1188 ± 0.0032	7.3333 ± 1.5275

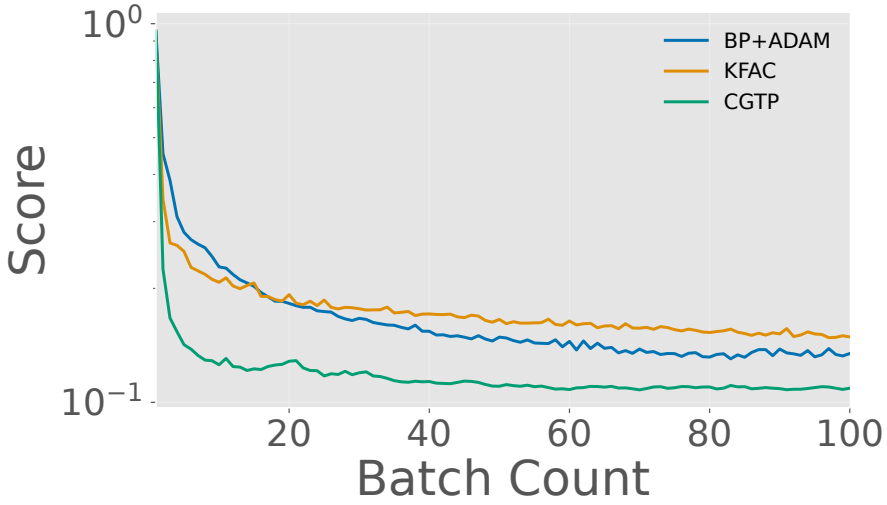

Figure 1: Comparing convergence of different optimisers at batch size 0.1 (Fashion-MNIST). CGTP hits the 0.13 score milestone after 7 batches. BP hits it at 50 batches, and KFAC does not reach it.

D.4 Ablation Studies

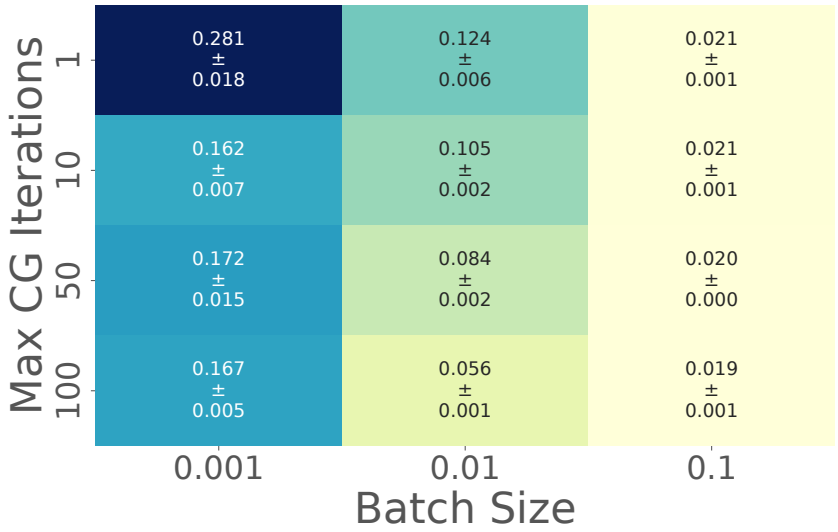

Figure 2: Results of a grid search over max CG iterations and batch size on validation error (MNIST dataset). By far the batch size has the biggest impact on model performance, due to its importance in estimating curvature (training curves may be seen in Figure4). However, even with a poor curvature estimate, increasing the max cg iter can improve performance - though its impact is diminished once sufficient batch size is reached.

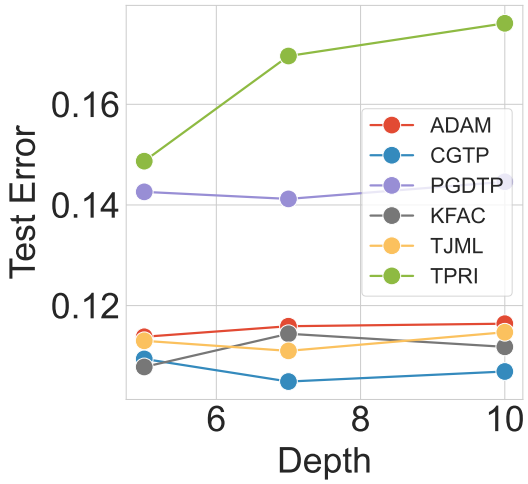

Figure 3: Left: a graph of the impact of network depth on test error (Fashion-MNIST). In comparison with methods such as TPRI, CGTP is stable across depth. This is likely due to the difference between the regularised composite Jacobian, and the product of regularised layer Jacobians.

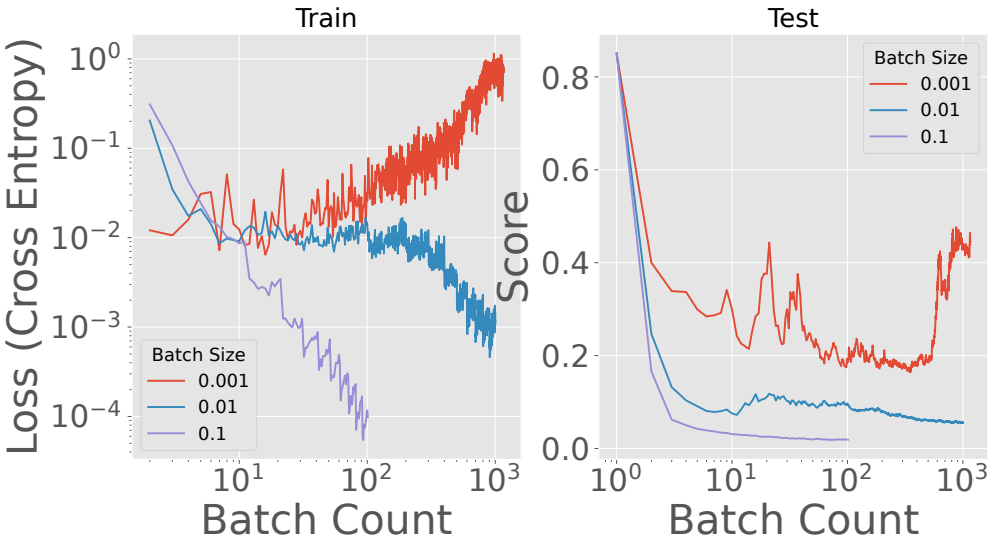

Figure 4: Varying the batch size at fix max cg iter (Fashion MNIST): We keep the max cg iter fixed at 100, and vary the batch size for CGTP. Larger batches result drastically accelerated learning, which is also reflected in the test error (with diminishing returns). This highlights a known phenomenon of second order optimizers, as they need larger batch sizes to get better second order estimates of the curvature.

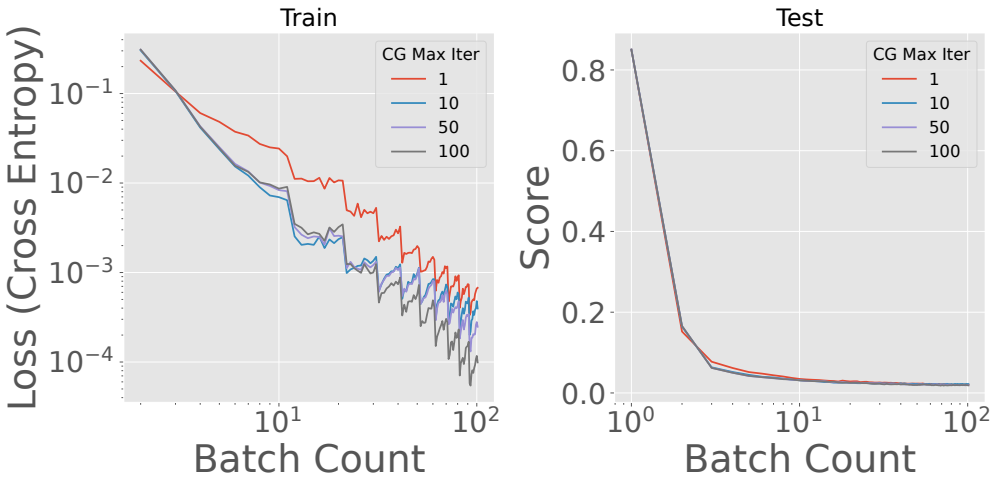

Figure 5: Varying the max cg iter at fixed batch size (Fashion-MNIST): We keep the batch sized fixed at batch size 0.1 for CGTP. We see small improvements in terms of learning speed on the train set, but this does little for the test error.

E ALGORITHM LISTINGS

Algorithm 1 Preconditioned conjugate gradient for the layer-ℓ realisation problem

Require: inputs $\tilde{U}^{(\ell-1)}$ with rows $u_n^\top$; per-sample $J_n^{(\ell)}$, M_n, r_n, $\alpha_n^{(\ell)}$, $B_n^{(\ell)}$; trust parameter λ; preconditioner damping $\delta \geq 0$; tolerance ε; iteration budget T
Ensure: approximate solution $\Delta W^{(\ell-1)}$ of equation 26

function APPLYG(V) ▷ $\mathcal{G}_\lambda^{(\ell)}(V)$, matrix-free
  $Y \leftarrow \tilde{U}^{(\ell-1)} V$ ▷ rows $y_n^\top = u_n^\top V$
  for $n = 1, \ldots, N$ **do**
    $\hat{y}_n \leftarrow \frac{1}{\alpha_n^{(\ell)}} J_n^{(\ell)\top} M_n J_n^{(\ell)} y_n + \lambda B_n^{(\ell)} y_n$ ▷ one JVP, one VJP
  end for
  return $\tilde{U}^{(\ell-1)\top} \hat{Y}$
end function

function PRECOND(R) ▷ inverse of $(A + \delta I) \otimes \bar{G}$
  return $(A + \delta I)^{-1} R \bar{G}^{-1}$ ▷ two Cholesky solves
end function

$g_n \leftarrow J_n^{(\ell)\top} M_n r_n$ for all n; $\quad \Gamma \leftarrow -\sum_n u_n g_n^\top$
$A \leftarrow \tilde{U}^{(\ell-1)\top} \tilde{U}^{(\ell-1)}$; $\quad \bar{G} \leftarrow \frac{1}{N} \sum_n \tilde{G}_n^{(\ell)}(\lambda; \alpha)$
$\Delta z_n^{(\ell)\star} \leftarrow -\tilde{G}_n^{(\ell)}(\lambda; \alpha)^{-1} g_n$ for all n ▷ targets
$X \leftarrow (A + \delta I)^{-1} \tilde{U}^{(\ell-1)\top} \Delta Z^{(\ell)\star}$ ▷ warm start, or $X \leftarrow 0$
$R \leftarrow \Gamma - \text{APPLYG}(X)$; $\quad \hat{R} \leftarrow \text{PRECOND}(R)$; $\quad P \leftarrow \hat{R}$; $\quad \rho \leftarrow \langle R, \hat{R} \rangle$
for $t = 1, \ldots, T$ **do**
  if $\|R\|_F \leq \varepsilon \|\Gamma\|_F$ **then break**
  end if
  $\Theta \leftarrow \text{APPLYG}(P)$
  $a \leftarrow \rho / \langle P, \Theta \rangle$
  $X \leftarrow X + aP$; $\quad R \leftarrow R - a\Theta$
  $\hat{R} \leftarrow \text{PRECOND}(R)$; $\quad \rho' \leftarrow \langle R, \hat{R} \rangle$
  $P \leftarrow \hat{R} + (\rho'/\rho) P$; $\quad \rho \leftarrow \rho'$
end for
return $\Delta W^{(\ell-1)} \leftarrow X$