Careful with that Scalpel: Improving Gradient Surgery with an EMA

Yu-Guan Hsieh [1] James Thornton [1] Eugene Ndiaye [1] Michal Klein [1] Marco Cuturi [1] Pierre Ablin [1]

Abstract

Beyond minimizing a single training loss, many deep learning estimation pipelines rely on an auxiliary objective to quantify and encourage desirable properties of the model (e.g. performance on another dataset, robustness, agreement with a prior). Although the simplest approach to incorporating an auxiliary loss is to sum it with the training loss as a regularizer, recent works have shown that one can improve performance by blending the gradients beyond a simple sum; this is known as *gradient surgery*. We cast the problem as a constrained minimization problem where the auxiliary objective is minimized among the set of minimizers of the training loss. To solve this bilevel problem, we follow a parameter update direction that combines the training loss gradient and the orthogonal projection of the auxiliary gradient to the training gradient. In a setting where gradients come from mini-batches, we explain how, using a moving average of the training loss gradients, we can carefully maintain this critical orthogonality property. We demonstrate that our method, Bloop, can lead to much better performances on NLP and vision experiments than other gradient surgery methods without EMA.

1. Introduction

Overparameterized neural networks trained on large datasets admit multiple solutions with the same optimal training loss (Cooper, 2018; Li et al., 2018). Although these parameters may seem equivalent when viewed through their training loss, they result in different functions, which may exhibit starkly different behaviors on unseen data points. Practitioners are usually interested in generalization — one would rather use the network with lower test loss between two networks — but there are countless other metrics of inter-

[1]Apple. Correspondence to: Pierre Ablin <p_ablin@apple.com>.

est, such as performance on another dataset, robustness, or model calibration. In all of these cases, one aims to train the neural network by minimizing a training loss L_{main} while keeping an eye on an auxiliary metric or loss L_{aux}.

Optimization trade-offs. Our focus in this paper is on methods that achieve the best possible trade-off between training and auxiliary losses, using a hyper-parameter $\lambda \geq 0$ to control that trade-off: $\lambda = 0$ corresponds to training on L_{main} exclusively, while increasing λ usually decreases L_{aux} at the expense of L_{main}. Using the auxiliary loss as a *regularizer* results in the *mixed training method*, arguably the simplest approach to control that trade-off:

$$\min_{\theta} L_{\mathrm{main}}(\theta) + \lambda L_{\mathrm{aux}}(\theta). \tag{1}$$

Mixed training, however, runs into optimization issues if the directions of the largest curvature of the training loss and that of the auxiliary loss are not aligned — see Section 3.3 for an example.

The Simple Bilevel Approach. Provided that modern deep neural networks are inherently overparameterized, leading to multiple minimizers, an ideal solution would be to find the minimizer of L_{main} that achieves the smallest auxiliary loss. This corresponds to solving Equation 1 in the limit where $\lambda \to 0$, and can also be expressed as the following *simple bilevel* problem (Dempe et al., 2010):

$$\min L_{\mathrm{aux}}(\theta) \ \text{ s.t. } \ \theta \in \arg\min L_{\mathrm{main}}(\theta). \tag{2}$$

Problem (2) is a constrained optimization problem on the set of minimizers of L_{main}, a high-dimensional set with no clear structure, except when L_{main} is convex, in which case several provably convergent approaches have been proposed (Sabach & Shtern, 2017; Gong & Liu, 2021; Cao et al., 2023). However, to the best of our knowledge, these methods have not been applied to training neural networks, where these convergence guarantees do not hold.

Connections to Multi-Task Learning. The problem of simultaneously optimizing the main and auxiliary loss is also a special case of *multi-task* learning (Caruana, 1997) involving only two tasks. Many of the approaches proposed to tackle this problem more efficiently rely on the idea of *gradient surgery*, which stitches together and possibly modify the gradients of both losses when they disagree (Yu et al.,

2020). While multi-task methods tend to treat the two losses equally, we are interested in our work in cases where there is a clear hierarchy between the two.

Two types of auxiliary losses. Auxiliary objectives largely fall into two categories. The first consists of objectives that guide optimization of the main loss but are not intrinsically meaningful; also known as inductive biases, they are only useful to reach a lower test loss. Weight decay, $L_{\text{aux}} = \frac{1}{2}\|\cdot\|^2$, fits this description: using it improves generalization, but practitioners rarely care about the final norm of their parameters. The second category of auxiliary losses quantify instead a desirable property: Trading off an increase in the main loss for a decrease in the auxiliary loss might be relevant to applications. For instance, the main objective might a loss on a large dataset, whereas the auxiliary objective may be a loss on a smaller, specialized dataset. Ideally, one wishes to achieve a model with high accuracy on both, and hope that the auxiliary loss might also help generalization on the large training set, but both objectives remain meaningful on their own. Another example is in training neural networks that are also smooth, i.e., with a small Lipschitz constant. This is beneficial for the networks' robustness (Cisse et al., 2017). To enforce this during training, one can use a proxy for the Lipschitz constant of the neural network as an auxiliary loss (Tsuzuku et al., 2018; Terjék, 2019).

Contributions. To handle the optimization tradeoff between main and auxiliary losses, we introduce in Section 2 the **Bloop** (**Bi**Level **O**ptimization with **O**rthogonal **P**rojection) method. Our method is inspired by the simple bilevel problem, but similar to the regularization approach, has a tunable hyperparameter, λ, to control the trade-off between losses. At the heart of the method is a projection of the auxiliary gradient to be orthogonal to the primary loss gradient. We first provide a theoretical justification for this approach in the full-batch case. In the stochastic setting, we rely on an exponential moving average (EMA) of the training gradient to estimate the projection direction, and retain most of the full-batch theoretical properties. In Section 3, we analyze Bloop's stationary points, and show that they are first-order stationary points of the simple bilevel problem. We demonstrate the convergence of the iterates towards the stationary points of the training loss, under appropriate hypothesis on the step size and the EMA accumulation factor, highlighting the importance of the EMA. In Section 4, we discuss related methods that perform variants of gradient surgery. In Section 5, we explore the applicability of our method to a variety of tasks: training network parameters with an explicit bias; multi-task learning; training language models to perform well on a large generic dataset and a small specific dataset. In our experiments, Bloop exhibits a better Pareto front than both the mixed method and multi-task methods that do not use an EMA.

2. The Bloop Algorithm

In this section, we introduce Bloop, a simple and intuitive iterative algorithm to optimize two losses simultaneously. We then discuss how the method can be extended to address stochasticity in the gradients, and multi-level optimization.

2.1. Full-batch setting and main intuition

At each step, Bloop builds a parameter update direction $d \in \mathbb{R}^p$ which is then fed to an optimizer (e.g. Adam (Kingma & Ba, 2014)) in order to converge to the solution of Equation 2. For instance, the gradient descent optimizer would iterate $\theta \leftarrow \theta - \eta d$. At the current iterate θ, we let $g_{\text{main}} = \nabla L_{\text{main}}(\theta)$ and $g_{\text{aux}} = \nabla L_{\text{aux}}(\theta)$.

We design our direction from first principles. We seek a direction in the span of these two gradients, $d = \omega g_{\text{main}} + \lambda g_{\text{aux}}$ with ω and λ two scalars. Our primary goal is to make progress on the main loss at the same speed as gradient descent; hence we target $L_{\text{main}}(\theta - \eta d) \simeq L_{\text{main}}(\theta - \eta g_{\text{main}})$.

At the first order in the step-size η, we see that the component of the direction in the direction g_{main} should be the same as that of g_{main}, i.e., we want $\langle d, g_{\text{main}} \rangle = \|g_{\text{main}}\|^2$. This gives the equation $(1 - \omega)\|g_{\text{main}}\|^2 = \lambda \langle g_{\text{main}}, g_{\text{aux}} \rangle$. Our secondary goal is the optimization of the auxiliary loss, hence we impose that the coefficient in front of g_{aux} is positive, i.e. that $\lambda > 0$. These two conditions alone give us our update rule: we find that such a direction is necessarily

$$d = g_{\text{main}} + \lambda \pi(g_{\text{aux}}, g_{\text{main}}), \text{ where}$$
$$\pi(g_{\text{aux}}; g_{\text{main}}) = g_{\text{aux}} - \frac{\langle g_{\text{aux}}, g_{\text{main}} \rangle}{\|g_{\text{main}}\|^2} g_{\text{main}} \quad (3)$$

Hyperparameter $\lambda \geq 0$ trades-off the two objectives, and $\pi(g_{\text{aux}}; g_{\text{main}})$ is the projection of g_{aux} orthogonal to g_{main}. This direction admits an intuitive explanation: since we primarily want to optimize the main loss, we follow g_{main}; the projection part is aligned with g_{aux}, and does not interfere with g_{main} thanks to the orthogonality condition. Moreover, the fact that $\langle d, g_{\text{main}} \rangle = \|g_{\text{main}}\|^2$ means that following this direction does not change the optimization with respect to L_{main} when step-sizes are small. Specifically, we write down the Taylor expansion at the first order

$$L_{\text{main}}(\theta - \eta d) \simeq L_{\text{main}}(\theta) - \eta \langle g_{\text{main}}, d \rangle,$$
$$(\text{Orthogonality}) \simeq L_{\text{main}}(\theta) - \eta \|g_{\text{main}}\|^2. \quad (4)$$

This is the same as standard gradient descent where $d = g_{\text{main}}$. Figure 1 illustrates the geometric principle of Bloop.
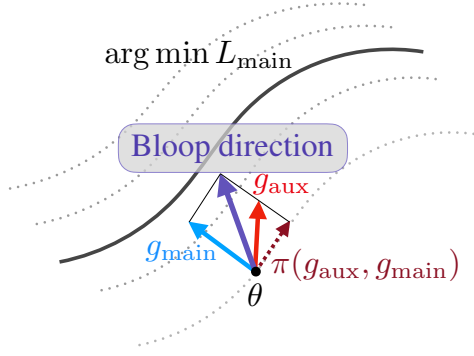
Figure 1. **Principle of the Bloop method**: the direction we follow is the sum of the gradient of the main loss g_{main}, and of the projection of the gradient of the auxiliary loss, orthogonal to g_{main}. This enforces that, at the first order, following this direction yields the same decrease in L_{main} as following g_{main}.

2.2. Stochastic extension for large-scale problems

When dealing with neural networks trained over large datasets, the losses are written as sums over many samples:

$$L_{\text{main}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} L_{\text{main}}^{i}(\theta), \quad L_{\text{aux}}(\theta) = \frac{1}{m} \sum_{j=1}^{m} L_{\text{aux}}^{j}(\theta).$$

In practice, we can only use a mini-batch of gradients to make progress on the problem, as the computation of the full-batch gradient of these losses is out of the question. Concretely, we assume that we have computed the two mini-batch gradients $g_{\text{main}}^{\text{batch}}, g_{\text{aux}}^{\text{batch}}$, which are by design unbiased estimators of the full-batch gradients:

$$\mathbb{E}[g_{\text{main}}^{\text{batch}}] = g_{\text{main}} \text{ and } \mathbb{E}[g_{\text{aux}}^{\text{batch}}] = g_{\text{aux}}.$$

In the above, the expectation is taken over the randomness of the mini-batch choice. Extending the direction d to this *stochastic* setting is not straightforward, and careful design makes a big difference in the final performance. A key insight behind standard, single-level, stochastic gradient descent on L_{main} is that, for small step sizes, it has on average the same decrease as gradient descent:

$$\mathbb{E}[L_{\text{main}}(\theta - \eta g_{\text{main}}^{\text{batch}})] \simeq L_{\text{main}}(\theta) - \eta \mathbb{E}[\langle g_{\text{main}}, g_{\text{main}}^{\text{batch}} \rangle]$$

(Linearity of dot) $\simeq L_{\text{main}}(\theta) - \eta \langle g_{\text{main}}, \mathbb{E}[g_{\text{main}}^{\text{batch}}] \rangle$

(Unbiased gradient) $\simeq L_{\text{main}}(\theta) - \eta \|g_{\text{main}}\|^2$

We want to preserve this behavior as much as possible. A first idea is simply to plug the mini-batch gradients in Equation 3, i.e. consider

$$d_{\text{simple}}^{\text{batch}} = g_{\text{main}}^{\text{batch}} + \lambda \pi(g_{\text{aux}}^{\text{batch}}; g_{\text{main}}^{\text{batch}}).$$

The pitfall of projecting on stochastic gradients. The main issue with the above method is that the projection is
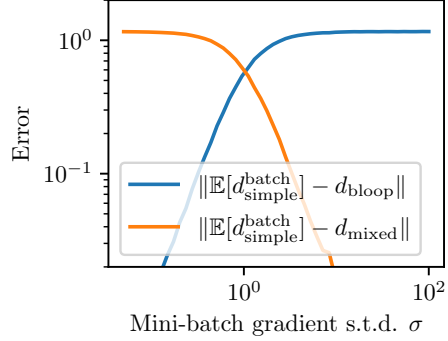

Figure 2. **Effect of randomness on the projection**: We fix the dimension of the parameter space to $p = 100$, and draw both g_{main} and g_{aux} from the Gaussian distribution $\mathcal{N}(\mathbf{0}, I)$. These two vectors are fixed in the remainder of the experiment. We draw $g_{\text{main}}^{\text{batch}} \sim g_{\text{main}} + \sigma \mathcal{N}(\mathbf{0}, I)$ and use Monte-Carlo simulation to estimate $\mathbb{E}[d_{\text{simple}}^{\text{batch}}] = g_{\text{main}} + \mathbb{E}[\pi(g_{\text{aux}}; g_{\text{main}}^{\text{batch}})]$. We compare its value against $d_{\text{bloop}} = g_{\text{main}} + \pi(g_{\text{aux}}; g_{\text{main}})$, its theoretical value when $\sigma = 0$ (the target direction), and $d_{\text{mixed}} = g_{\text{main}} + (1 - 1/100)g_{\text{aux}}$, its theoretical value when σ tends to infinity. We see that the $\mathbb{E}[d_{\text{simple}}^{\text{batch}}]$ becomes closer to the gradient of the mixed method when the noise starts to dominate.

nonlinear with respect to its second argument: in general, $\mathbb{E}[\pi(g_{\text{aux}}^{\text{batch}}; g_{\text{main}}^{\text{batch}})] \neq \pi(g_{\text{aux}}; g_{\text{main}})$. As a consequence, it is not true anymore that $\langle d_{\text{simple}}^{\text{batch}}, g_{\text{main}} \rangle = \|g_{\text{main}}\|^2$, even in expectation, which in turn leads to a behavior starkly different from SGD on L_{main}. We can improve this intuition using a simplified model of the training dynamics. Assume that $g_{\text{main}}^{\text{batch}} = g_{\text{main}} + \sigma \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, I)$ is the random gradient noise, and $\sigma > 0$ is the noise variance. In the limit where σ is large in front of $\|g_{\text{main}}\|$, we get that on average $\mathbb{E}_{\varepsilon}[\pi(g_{\text{aux}}^{\text{batch}}; g_{\text{main}}^{\text{batch}})] = (1 - \frac{1}{p})g_{\text{aux}}^{\text{batch}}$ with p the parameter's dimension. Therefore, the simple direction is on average $d_{\text{simple}}^{\text{batch}} = g_{\text{main}}^{\text{batch}} + \lambda(1 - \frac{1}{p})g_{\text{aux}}^{\text{batch}}$. We recover the same direction as that of the mixed training method, with a new $\lambda' = \lambda(1 - \frac{1}{p})$, and the orthogonalization becomes useless.

In order to illustrate this intuition, we conduct a synthetic experiment, explained in Figure 2.

The EMA solution. The previous analysis indicates that we need a better estimate of g_{main} than the mini-batch gradient. A simple solution to this is to use an Exponential Moving Average (EMA) of the previous batch gradients, $g_{\text{main}}^{\text{EMA}}$, which is updated at each iteration by doing $g_{\text{main}}^{\text{EMA}} \leftarrow (1 - \rho)g_{\text{main}}^{\text{EMA}} + \rho g_{\text{main}}^{\text{batch}}$, with $\rho \in [0, 1]$ a parameter that controls the speed of the EMA. This can be a much better estimator of g_{main} than $g_{\text{main}}^{\text{batch}}$, because it averages gradients over the optimization trajectory, drastically reducing the variance. Intuitively, we need to accumulate the EMA faster than the speed of the optimization algorithm that updates the parameters. Hence, ρ should be greater than the step-size η. We use this gradient EMA solely in the

Algorithm 1 The Bloop algorithm

Input: Hyperparameter λ, EMA parameter ρ, initial parameters θ, optimizer `optim`, optimizer state s, initial EMA $g_{\mathrm{main}}^{\mathrm{EMA}}$

for $t = 0, \ldots, T - 1$ **do**

    Sample gradients $g_{\mathrm{main}}^{\mathrm{batch}}, g_{\mathrm{aux}}^{\mathrm{batch}}$

    Compute the Bloop direction d^{batch} using Equation 5

    Update $\theta, s \leftarrow \mathtt{optim}(d^{\mathrm{batch}}, \theta, s)$

    Update EMA: $g_{\mathrm{main}}^{\mathrm{EMA}} \leftarrow (1 - \rho)g_{\mathrm{main}}^{\mathrm{EMA}} + \rho g_{\mathrm{main}}^{\mathrm{batch}}$

end for

projection, and propose the direction

$$d^{\mathrm{batch}} = g_{\mathrm{main}}^{\mathrm{batch}} + \lambda\pi(g_{\mathrm{aux}}^{\mathrm{batch}}; g_{\mathrm{main}}^{\mathrm{EMA}}) \qquad (5)$$

We do not replace the first $g_{\mathrm{main}}^{\mathrm{batch}}$ in the formula by the EMA, because d^{batch} is an optimization *direction*, that is then plugged into any optimizer like Adam, which will use a smart adaptive step to reach the solution quickly. Since the EMA does not depend on the current batch, and the projection is linear with respect to its first argument, we have that $\mathbb{E}[d^{\mathrm{batch}}] = g_{\mathrm{main}} + \lambda\pi(g_{\mathrm{aux}}; g_{\mathrm{main}}^{\mathrm{EMA}})$, and as a consequence, the expected decrease on L_{main} following this direction is $\mathbb{E}[L_{\mathrm{main}}(\theta - \eta d^{\mathrm{batch}})] \simeq L_{\mathrm{main}}(\theta) - \eta\|g_{\mathrm{main}}\|^2 + \eta\lambda\langle\pi(g_{\mathrm{aux}}; g_{\mathrm{main}}^{\mathrm{EMA}}), g_{\mathrm{main}}\rangle$. When the EMA accumulation $g_{\mathrm{main}}^{\mathrm{EMA}}$ is close to g_{main}, the last term becomes small because the two vectors are approximately orthogonal. Thus,

$$\mathbb{E}[L_{\mathrm{main}}(\theta - \eta d^{\mathrm{batch}})] \simeq L_{\mathrm{main}}(\theta) - \eta\|g_{\mathrm{main}}\|^2,$$

and we recover the same behavior as SGD on L_{main}. The new direction is no longer in the span of $(g_{\mathrm{main}}^{\mathrm{batch}}, g_{\mathrm{aux}}^{\mathrm{batch}})$ because it also has a component in the direction of $g_{\mathrm{main}}^{\mathrm{EMA}}$.

The theory presented in the next section clearly highlights the importance of this EMA, and in our experiments, we find that this simple EMA modification drastically improves the performance of the algorithm on a variety of tasks. In fact, we found that in many cases, standard multi-task methods without EMA have very similar performances to the mixed training method.

Algorithm 1 gives the full pseudo-code of the Bloop method. We use `optax`-like notations (DeepMind et al., 2020) for the optimizer, which is abstracted as a method that, given a direction d, current parameters θ and a state s containing all its hyper-parameters like learning rate and internal state like EMAs for adaptive methods, returns the updated parameters θ and updated state s.

2.3. Extension to multi-level hierarchical optimization

Our algorithm can be extended to multi-level optimization, where we have more than two losses and they have a *hi-*

erarchy. For simplicity, we present here the case with 3 losses: L_{main}, L_{aux}^1 and L_{aux}^2. The hierarchy means that we minimize L_{main}, and then, among this set of minimizers, we minimize L_{aux}^1. Finally, we minimize L_{aux}^2 among this new set. This gives the trilevel optimization problem:

$$\min L_{\mathrm{aux}}^2(\theta) \text{ s.t.}$$
$$\theta \in \left(\arg\min L_{\mathrm{aux}}^1(\theta) \text{ s.t. } \theta \in \arg\min L_{\mathrm{main}}(\theta)\right) \qquad (6)$$

Our algorithm can be straightforwardly extended to this case by following a Gram-Schmidt like orthogonalization process: letting g_{main}, g_{aux}^1 and g_{aux}^2 the gradients of the three losses, we go in the direction

$$d = g_{\mathrm{main}} + \lambda^1\pi(g_{\mathrm{aux}}^1; g_{\mathrm{main}}) + \lambda^2\pi(g_{\mathrm{aux}}^2; (g_{\mathrm{main}}, g_{\mathrm{aux}}^1))$$

where $\pi(g_{\mathrm{aux}}^2; (g_{\mathrm{main}}, g_{\mathrm{aux}}^1))$ is the projection of g_{aux}^2 on the orthogonal of the span of $(g_{\mathrm{main}}, g_{\mathrm{aux}}^1)$. Thanks to orthogonality, this direction satisfies $\langle d, g_{\mathrm{main}}\rangle = \|g_{\mathrm{main}}\|^2$; hence in terms of optimization with respect to L_{main}, the direction behaves just like g_{main}, and $\langle d, g_{\mathrm{aux}}^1\rangle = \langle g_{\mathrm{main}} + \lambda^1\pi(g_{\mathrm{aux}}^1; g_{\mathrm{main}}), g_{\mathrm{aux}}^1\rangle$; hence in terms of optimization with respect to L_{aux}^1, the direction behaves just like the bilevel direction d introduced in Equation 3.

3. Theoretical Analysis

This section aims at understanding the theoretical properties of the proposed direction in the full-batch and the mini-batch settings by linking it with the simple bilevel problem (Equation 2). All the proofs are deferred to Appendix A.

3.1. Approximate stationary points of Bloop

At a solution to the simple bilevel problem, we have $\nabla L_{\mathrm{main}}(\theta) = 0$, hence the solutions to the bilevel problem are also solutions of

$$\min L_{\mathrm{aux}}(\theta) \text{ s.t. } \nabla L_{\mathrm{main}}(\theta) = 0.$$

The Lagrangian for this equation is $\mathcal{L}(\theta, v) = L_{\mathrm{aux}}(\theta) - \langle v, \nabla L_{\mathrm{main}}(\theta)\rangle$ with $v \in \mathbb{R}^p$ the Lagrange multiplier. Accordingly, the first-order optimality conditions are $g_{\mathrm{main}} = 0$ and that there exists v such that $g_{\mathrm{aux}} = \nabla^2 L_{\mathrm{main}}(\theta)v$. A first natural question to ask is whether the direction that we propose in Equation 3 cancels at these points. However, the projection is ill-defined when $g_{\mathrm{main}} = 0$. We thus assume that $\|g_{\mathrm{main}}\|$ is positive hereinafter and focus on the case where d is small but non-zero.[1] To analyze this, we introduce the following assumption.

Assumption 1 (Local Error Bound Luo & Tseng, 1993)**.** There exists $c > 0$ such that for ε small enough and for any

---

[1] Although we can simply set $d = \lambda g_{\mathrm{aux}}$ when $g_{\mathrm{main}} = 0$, the study of this particular case is straightforward and gives little insight on the general case. We therefore omit it here.

θ satisfying $\|g_{\mathrm{main}}(\theta)\| \leq \varepsilon$, we have

$$\mathrm{Dist}(\theta, \nabla L_{\mathrm{main}}^{-1}(\{0\})) \leq c\|g_{\mathrm{main}}\|.$$

This local error bound condition is implied by a local Polyak-Lojasiewicz inequality, which is verified, for instance, for overparameterized least-squares and some neural network loss functions (Liu et al., 2022). With this in hand, we are now ready to present our result regarding the approximate first-order stationary points of the full-batch Bloop method.

Proposition 1 (Stationary points). *If d in Equation 3 is such that $\|d\| \leq \varepsilon$, then we have $\|g_{\mathrm{main}}\| \leq \varepsilon$. Moreover if Assumption 1 holds, the Hessian of L_{main} is $M-$Lipschitz, and ε is small enough, then there exists $v \in \mathbb{R}^p$ such that*

$$\|g_{\mathrm{aux}} - \nabla^2 L_{\mathrm{main}}(\theta)v\| \leq (\lambda^{-1} + Mc^2\|g_{\mathrm{aux}}\|/2)\varepsilon.$$

Conversely, given a point θ^* that satisfies the first order optimality conditions of Equation 2, we have that $\lim_{\varepsilon \to 0} d(\theta^* + \varepsilon v) = 0$ where v is the Lagrange multiplier.

In short, Proposition 1 relates the (approximate) stationary points of Bloop to the (approximate) stationary points of the bilevel problem. Moreover, as an immediate consequence of the proposition, we see that we additionally assume L_{aux} to be Lipschitz continuous, the limit points of Bloop must be stationary points of the simple bilevel problem.

3.2. Convergence of stochastic Bloop

Our main theorem is a convergence result of the *stochastic version* of Bloop. It clearly highlights the role of the EMA: without EMA, obtaining such results would be impossible.

Theorem 2 (Convergence of Bloop). *Consider the Bloop method in the stochastic setting with the SGD optimizer. Let ρ be the EMA parameter and η be the step-size of the algorithm. Assume that (i) L_{main} is L-smooth, (ii) the stochastic directions are uniformly bounded, i.e., $\|d^t\| \leq D$ for all t, (iii) the variance of the gradients of L_{main} is bounded with $\mathbb{E}_i[\|\nabla L_{\mathrm{main}}^i(\theta) - \nabla L_{\mathrm{main}}(\theta)\|^2] \leq C^2$, and (iv) the auxiliary gradients are bounded as $\|\nabla L_{\mathrm{aux}}(\theta)\| \leq B$. Then, for a number of iterations T, taking a step size $\eta \simeq T^{-\frac{3}{4}}$ and an EMA parameter $\rho \simeq \eta^{\frac{2}{3}}$ gives*

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla L_{\mathrm{main}}(\theta^t)\|^2] = O(T^{-\frac{1}{4}})$$

If L_{main} is additionally μ-PL (Karimi et al., 2016), we have

$$\mathbb{E}[L_{\mathrm{main}}(\theta^T) - \min L_{\mathrm{main}}] \leq (1 - 2\eta\mu)^T L_{\mathrm{main}}(\theta^0) + O(\eta^{\frac{1}{3}}).$$

Theorem 2 demonstrates the convergence of stochastic Bloop either in terms of the expected gradient norm or the expected optimiality gap. In spirit, this suggests that the

Bloop iterate would end up being arbitrarily close to the stationary points of L_{main}. The theorem also instructs us on the role of the EMA coefficient ρ compared to the learning rate η. We see that we should take ρ to be slightly larger than η: in this regime, the gradient EMA $g_{\mathrm{train}}^{\mathrm{EMA}}$ is a good approximation of g_{train}.

Also note that this result differs significantly from those obtained in the multi-task learning literature, which show convergence of the algorithms to points where either both losses are minimized or where their gradients are opposed (Yu et al., 2020). Here, even in the extreme case where losses are the exact opposite ($L_{\mathrm{aux}} = -L_{\mathrm{main}}$), full-batch Bloop provably converges to the minimizers of L_{main} under PL condition. This is not a surprise since in that case, the projection $\pi(g_{\mathrm{aux}}, g_{\mathrm{main}})$ cancels and the iterates of Bloop are that of gradient descent on L_{main}.

Unlike Gong & Liu (2021), we do not demonstrate the convergence of our algorithm to the KKT points of the simple bilevel problem. Our results are thus weaker in that regard, albeit in a different setting since Gong & Liu (2021) are not in the stochastic case.

3.3. Conditioning compared to regularization method

We illustrate below that the regularization method can lead to poorly conditioned problems, resulting in hard optimization problems, while our method alleviates this. For this, we take the following simple 2D example, where $\theta = (a, b)$:

$$L_{\mathrm{main}}(\theta) = \frac{1}{2}a^2, \quad L_{\mathrm{aux}}(\theta) = \frac{1}{2}((a-1)^2 + b^2).$$

The solution to the bilevel problem is $\theta^* = 0$, while the solution to the regularized problem is $\theta = (\lambda/(1+\lambda), 0)$. We recover the same solution in the limit $\lambda \to 0$. However, the Hessian of the regularized problem is $\mathbf{diag}(1 + \lambda, \lambda)$; hence the conditioning of the regularized problem is $1 + 1/\lambda$ which goes to infinity as $\lambda \to 0$. In view of this, the regularized method either converges to a point far from the solution (λ large) or converges slowly (λ small). On the contrary, the projection method goes in the direction $d = (a, \lambda b)$. This is equivalent to gradient descent on a quadratic loss with the correct θ^* minimizer — *regardless of λ* — and Hessian equal to $\mathbf{diag}(1, \lambda)$, which is well conditioned when λ is not too far from 1.

4. Related Works

Our work sits at the intersection of two fields of machine learning: the solution of the simple bilevel problem and multi-task learning. There are however a number of differences between the two. In particular, in the multi-task learning problem each task is considered jointly whereas in the bilevel setting there is a hierarchy to the primary and auxiliary objectives. Another key difference is in the notion

Table 1. Comparison of similar gradient surgery methods for the two tasks setting. For brevity, we write $g_{\mathrm{m}} := g_{\mathrm{main}}$ and $\phi := \cos(g_{\mathrm{m}}, g_{\mathrm{aux}}) = \frac{\langle g_{\mathrm{m}}, g_{\mathrm{aux}} \rangle}{\|g_{\mathrm{m}}\| \|g_{\mathrm{aux}}\|}$. $(\bar{\cdot})$ indicates that EMA has been applied, and ψ is a dynamic barrier function described in (Gong & Liu, 2021).

Method	Modified Direction
Bloop (ours)	$g_{\mathrm{m}} + \lambda \left(g_{\mathrm{aux}} - \frac{\langle g_{\mathrm{aux}}, \bar{g}_{\mathrm{m}} \rangle}{\|\bar{g}_{\mathrm{m}}\|^2} \bar{g}_{\mathrm{m}} \right)$
Mixed (Regularized)	$g_{\mathrm{m}} + \lambda g_{\mathrm{aux}}$
A-GEM Chaudhry et al. (2018)	$g_{\mathrm{m}} - \frac{\min(0, \langle g_{\mathrm{m}}, g_{\mathrm{aux}} \rangle)}{\|g_{\mathrm{aux}}\|^2} g_{\mathrm{aux}}$
Dynamic Barrier Gong & Liu (2021)	$g_{\mathrm{aux}} + \max(0, \frac{\psi(\theta) - \langle g_{\mathrm{m}}, g_{\mathrm{aux}} \rangle}{\|g_{\mathrm{m}}\|^2}) g_{\mathrm{m}}$
MTL-MOO Sener & Koltun (2018)	$\frac{\langle g_{\mathrm{m}} - g_{\mathrm{aux}}, g_{\mathrm{aux}} \rangle}{\|g_{\mathrm{m}} - g_{\mathrm{aux}}\|^2} g_{\mathrm{m}} +$ $(1 - \frac{\langle g_{\mathrm{m}} - g_{\mathrm{aux}}, g_{\mathrm{aux}} \rangle}{\|g_{\mathrm{m}} - g_{\mathrm{aux}}\|^2}) g_{\mathrm{aux}}$
Cosine Similarity Du et al. (2018)	$g_{\mathrm{m}} + g_{\mathrm{aux}} \max(0, \phi)$
GradVac Wang & Tsvetkov (2021)	$g_{\mathrm{m}} + \frac{\|g_{\mathrm{m}}\| \left(\bar{\phi} \sqrt{1 - \phi^2} - \phi \sqrt{1 - \bar{\phi}^2} \right)}{\|g_{\mathrm{aux}}\| \sqrt{1 - \bar{\phi}^2}\|}$
PCGrad Yu et al. (2020)	$g_{\mathrm{m}} - \min(0, \langle g_{\mathrm{aux}}, g_{\mathrm{m}} \rangle) \frac{g_{\mathrm{m}}}{\|g_{\mathrm{m}}\|^2}$ $+ g_{\mathrm{aux}} - \min(0, \langle g_{\mathrm{aux}}, g_{\mathrm{m}} \rangle) \frac{g_{\mathrm{aux}}}{\|g_{\mathrm{aux}}\|^2}$
Meta-Balance He et al. (2022)	$g_{\mathrm{m}} + \frac{\|g_{\mathrm{m}}\|}{\|g_{\mathrm{aux}}\|} g_{\mathrm{aux}}$

of task versus auxiliary objective. A task typically requires a dataset as input, whereas an auxiliary objective is more general and can incorporate losses without the need for data, such as the L^2 norm in weight decay.

Given the similarity, a number of gradient surgery methods that have been proposed in multi-task literature can be used to minimize both the main and the auxiliary objectives. We summarize the most relevant ones in Table 1. Some works try to leverage the auxiliary loss to obtain improvements on the main loss only (Du et al., 2018; Dery et al., 2021).

The Dynamic Barrier (DB) algorithm of Gong & Liu (2021), as detailed in Table 1, uses a similar orthogonal projection as in our proposal. It provably solves the bilevel problem. However, DB includes an additional barrier function, ϕ e.g. $\phi = \|g_{\mathrm{aux}}\|^2$, to control the trade-off between objectives, whereas we use a scalar, λ, similar to regularization methods, for this purpose. The other main differences between our proposal and the DB method are that we always use the projection, rather than conditioning on $\langle g_{\mathrm{m}}, g_{\mathrm{aux}} \rangle$, and most importantly, we use an EMA of main gradients to compute the projection, rather than the stochastic gradient. With $\phi = \|g_{\mathrm{aux}}\|^2$ and without the conditional update or EMA, the approaches would be the same. Gong & Liu (2021) do not discuss stochastic extensions of the method, which is of key importance to practitioners.

Yu et al. (2020) propose PCGrad, which, as shown in Table 1, can be regarded as a symmetrized version of our method. Unlike our method, the projection is again conditioned. Concretely, the parameters are updated in the direction of the combined gradient $g_{\mathrm{main}} + g_{\mathrm{aux}}$ when they are aligned, and projections are performed when this is not the case. The gradient alignment condition and the symmetry between the gradients implies that the algorithm does not solve the bilevel problem; instead (Yu et al., 2020, Thm.1) show that it minimizes the sum of the two losses or finds a point where g_{aux} and g_{main} go in opposite directions. Similarly to the DB method, no EMA is used in the projection.

5. Experiments

In this section we demonstrate the effectiveness of Bloop via numerical experiments on problems of three distinct categories: the use of auxiliary loss for imposing an explicit bias, multi-task learning, and joint dataset training. For each of these experiments, we use an optimizer with hyperparameters that work well for the minimization of solely the main loss, and never change these hyperparameters. As for the EMA parameter of Bloop, we take it as $\rho = 0.01$ in all experiments unless otherwise stated. Further experimental details can be found in Appendix B.

Note that Bloop incurs a negligible training cost compared to the standard regularized training, as it only requires two additional dot products in the parameter space.

5.1. Baselines and evaluation

We compare Bloop (Algorithm 1) to other popular gradient surgery methods that follow a similar design. We focus on the stochastic setup where we only have access to the gradients over a mini-batch of samples at each iteration.

Mixed. This method minimizes the regularized objective $L_{\mathrm{main}} + \lambda L_{\mathrm{aux}}$ with the direction $d = g_{\mathrm{main}}^{\mathrm{batch}} + \lambda g_{\mathrm{aux}}^{\mathrm{batch}}$.

Dynamic Barrier (DB). The original formulation of the DB method requires both an estimate of a lower bound on L_{main}, as well as an estimate of $L_{\mathrm{main}}(\theta)$, which are cumbersome to estimate in deep learning setups. We therefore forgo this part of the algorithm and instead incorporate the scaling factor λ to control the trade-off. We also replace the gradients in the original method by stochastic gradients. This results in the update direction $d = \mu g_{\mathrm{main}}^{\mathrm{batch}} + \lambda g_{\mathrm{aux}}^{\mathrm{batch}}$ where $\mu = \max \left(1 - \lambda \frac{\langle g_{\mathrm{main}}^{\mathrm{batch}}, g_{\mathrm{aux}}^{\mathrm{batch}} \rangle}{\|g_{\mathrm{main}}^{\mathrm{batch}}\|^2}, 0 \right)$.

PCGrad. Being motivated from a multi-task perspective, the original formulation of PCGrad does not use the scaling factor λ. By incorporating this factor, the update direction becomes $d = g_{\mathrm{main}}^{\mathrm{batch}} + \lambda g_{\mathrm{aux}}^{\mathrm{batch}}$ if $\langle g_{\mathrm{main}}^{\mathrm{batch}}, g_{\mathrm{aux}}^{\mathrm{batch}} \rangle > 0$, and $d = \pi(g_{\mathrm{main}}^{\mathrm{batch}}, g_{\mathrm{aux}}^{\mathrm{batch}}) + \lambda \pi(g_{\mathrm{aux}}^{\mathrm{batch}}, g_{\mathrm{main}}^{\mathrm{batch}})$ otherwise.
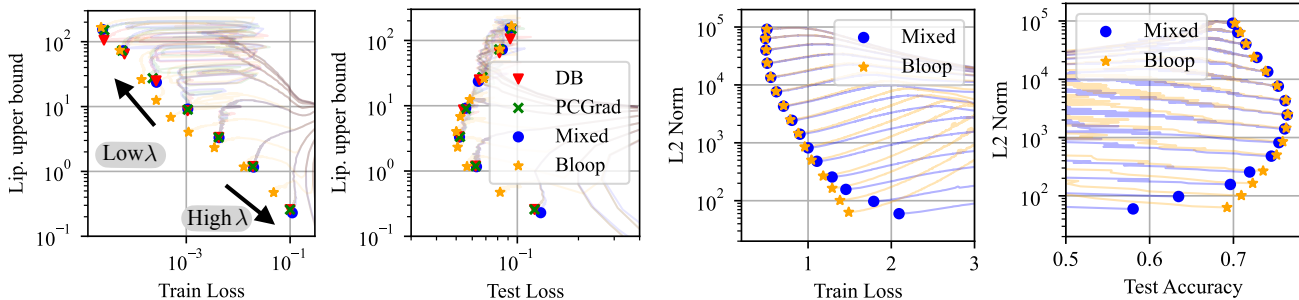
(a) Training an MLP on MNIST with an auxiliary loss that is a proxy for its Lipschitz constant.

(b) Training a ResNet50 on Imagenet with squared L2 norm as the auxiliary loss.

Figure 3. Trade-offs between the main and the auxiliary objectives in problems where the auxiliary loss is used to impose an explicit bias on the neural network. The symbols correspond to the parameters reached at the end of training and form a Pareto front, the transparent curves are the training trajectories. Bloop achieves a better trade-off than the other methods, which all perform similarly here.

Evaluation of the algorithms. To provide a comprehensive insight into how the algorithm design affects the training dynamics, we report the metrics on both the training and the test sets. Moreover, we trace the evolution of these metrics along training.

Pareto fronts. All algorithms that we consider here have thus a parameter λ that trades-off between the train and the auxiliary losses. After a fixed number of iterations, the algorithm `algo` finds a final parameter $\theta^{\texttt{algo}}(\lambda)$ that explicitly depends on λ. Generally, $L_{\mathrm{main}}(\theta^{\texttt{algo}}(\lambda))$ is a decreasing function of λ while $L_{\mathrm{aux}}(\theta^{\texttt{algo}}(\lambda))$ is increasing with λ. We can then vary λ to get the set of pairs $\mathcal{P}(\texttt{algo}) = \{(L_{\mathrm{main}}(\theta^{\texttt{algo}}(\lambda)), L_{\mathrm{aux}}((\theta^{\texttt{algo}}(\lambda)))| \ \lambda \geq 0\}$, called the Pareto front of `algo`.

5.2. Imposing an explicit bias during training

To begin with, we first investigate the situation where the auxiliary objective is used to enforce a certain desirable property (bias) on the neural network.

Training smooth neural networks. Following our discussion in Section 1, we explore the potential of Bloop in training smooth neural networks. For this, we use the MNIST dataset (LeCun et al., 2010) and an MLP of two hidden layers. With this minimal architecture, a simple induction argument shows that the Lipschitz constant of the network is upper-bounded by $\prod_{l=1}^{L} \|W_l\|_2$, where W_l is the weight matrix of the $l-$th linear layer, $\| \cdot \|_2$ is the spectral norm, and $L = 3$ is the number of layers. We thus define the auxiliary loss as $L_{\mathrm{aux}} = \log(\prod_{l=1}^{L} \|W_l\|_2)$. The use of logarithm here makes training easier. On the other hand, we use the standard cross-entropy loss as the main loss.

Training networks with small weights. For this experiment, we train a ResNet50 using standard cross-entropy loss on Imagenet, and try to simultaneously achieve a low ℓ_2 norm of the parameters of the network. The auxiliary
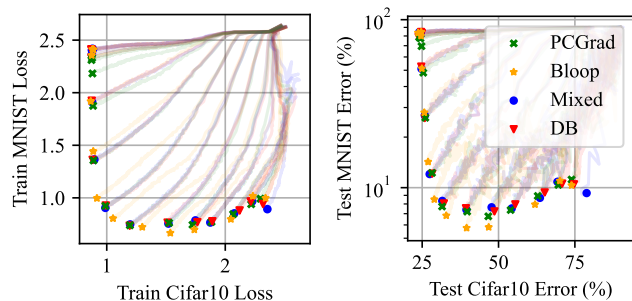

Figure 4. Trade-off between the performances in the Cifar10Mnist multi-task learning problem. Bloop gives a better Pareto front.

loss is therefore $L_{\mathrm{aux}}(\theta) = \frac{1}{2}\|\theta\|^2$. In that case, the mixed method is similar to training with a weight decay λ.

Results. The results are reported in Figure 3. We see Bloop induces training trajectories that are fundamentally different from all other methods, and leads to better Pareto fronts when trading off the main and the auxiliary training losses. In both experiments, we observe that Bloop leads to a significantly better Pareto front when looking at the training loss (Figure 3a, left and Figure 3b, left). Whether this translates or not to a better Pareto front in terms of test loss is problem dependent: in Figure 3a, right, the Pareto front of Bloop is only slightly better than that of the other methods, while in Figure 3b, right, it is significantly better.

5.3. Multi-task learning

As discussed in Section 1, multi-task learning represents another typical scenario in which such auxiliary objectives emerge. Following Hotegni et al. (2023), we construct a Cifar10Mnist dataset by overlapping digits from MNIST on images from CIFAR-10 (Krizhevsky et al., 2009) — see Figure 9 in Appendix B for an illustration. The main and the auxiliary tasks correpond respectively to identifying the label for the background CIFAR-10 image and for the MNIST digit. There is a natural hierachy between the two tasks here
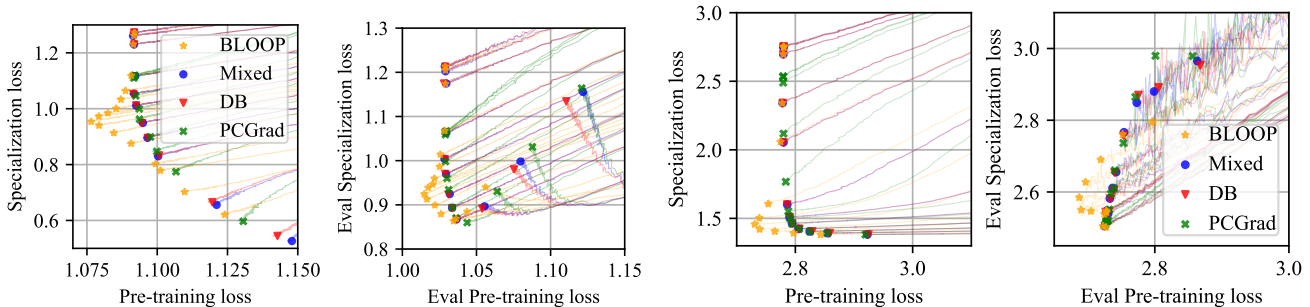
(a) Results on the language modeling task. The main, pre-training loss is the next-token-prediction loss over the large c4 dataset, while the auxiliary, specialization loss is the next-token-prediction loss over the small RCV-1 dataset.

(b) Results on the translation task. The main pre-training loss is the translation loss over the large paracrawl dataset, while the auxiliary specialization loss is the translation loss over the small WMT dataset.

Figure 5. Trade-offs between the main and the auxiliary objectives in problems in natural language processing experiments with transformer models, where the main loss is the loss over a large dataset and the auxiliary loss is a loss over a small dataset that can be overfitted easily. We observe that Bloop gets a significantly better Pareto front than all other methods, which perform similarly to the mixed method. Bloop gains in terms of optimization on the training losses transfer to the evaluation losses.

because identifying the CIFAR-10 label is more difficult than identifying the MNIST one. For this dataset, we train a ResNet18 with two classification heads to minimize the two cross-entropy losses. In this experiment, we found that taking $\rho = 0.001$ for Bloop gave better results.

Results. As shown in Figure 4, the trajectories of Bloop are again much more different than those of the other methods, which share quite similar behaviors. Moreover, Bloop gets a slightly improved Pareto front over those methods.

5.4. Joint training on two datasets

With the advent of large foundation models, it becomes increasingly common to train a model on multiple data sources (Gunasekar et al., 2023; Sun et al., 2023; Xu et al., 2023; Oquab et al., 2024). Yet, these datasets could have intrinsically different characteristics, and it may be natural to prioritize one over another, for instance when one dataset has far more samples than another. We explore the benefit of Bloop in such multi-dataset setting. Our experimental setup is similar to that of Grangier et al. (2023).

Transformer pre-training. We consider the problem of performing next-token-prediction with a decoder-only transformer on text data. The network is a transformer with 12 decoder layers, 8 attention heads, a residual dimension of 256, and a feed-forward latent dimension of 1024. The main loss corresponds to the prediction loss over a large pre-training dataset, while the auxiliary loss corresponds to that on a smaller but higher-quality dataset. Due to the lack of data, training only on the small high-quality dataset leads to severe overfitting and poor performance; hence, we resort to training on both datasets, using the proposed baselines or Bloop. For the training set, we use 30M examples from the c4 dataset (Raffel et al., 2020), while the auxiliary loss cor-

responds to 20K examples from the RCV-1 dataset (Lewis et al., 2004).

Translation. In this experiment, we train a network to translate English into German. The network is a transformer with 6 encoder layers and 6 decoder layers, 16 attention heads, a residual dimension of 1,024, and a feed-forward latent dimension of 4,096. Like in the pre-training experiment, we have a large generic dataset, the Paracrawl dataset (Bañón et al., 2020), with 36m sentence pairs, which defines the main loss. The auxiliary loss is the loss over a smaller but higher quality dataset, the 2009-2019 WMT dataset, yielding 10k sentence pairs (Farhad et al., 2021). We use the 2020 WMT dataset (2k pairs) as an evaluation set.

Results. Figure 5 displays the results. We observe siginificantly improved results for Bloop, which has once again a better Pareto front, and achieves smaller pre-training loss. These gains are kept when looking at the evaluation losses. Figure 6 gives a different perspective on those results.

5.5. Role of the EMA

We investigate the importance of the EMA parameter ρ in Bloop. As already seen in Section 3, it is critical from a theoretical point-of-view for the algorithm's convergence. We further illustrate this via the transformer pre-training experiment with a fixed $\lambda = 0.2$.

Figure 7 displays the results. We see that when the EMA is too small ($\rho = 0.001$), the value of $g_{\text{main}}^{\text{EMA}}$ is outdated compared to the current value of the gradient g_{main}, and therefore, the performance on both the main and auxiliary losses is bad. On the contrary, taking a too-large EMA ($\rho = 0.9$) means that $g_{\text{main}}^{\text{EMA}}$ has a high variance, and we recover a trajectory extremely similar to that of the mixed
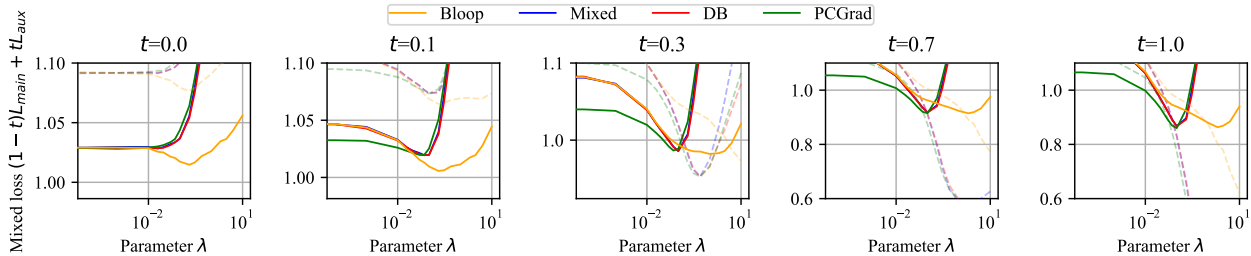
Figure 6. A different look at the results in Figure 5a. We display the value of the final mixed loss $(1-t)L_{\mathrm{main}} + tL_{\mathrm{aux}}$ for the different values of λ in the algorithms we used. Bold lines correspond to evaluation loss, while dotted lines correspond to train loss. We see that Bloop allows to get to a lower mixed loss when t is small. This is a striking phenomenon, since the mixed method directly minimizes that loss.
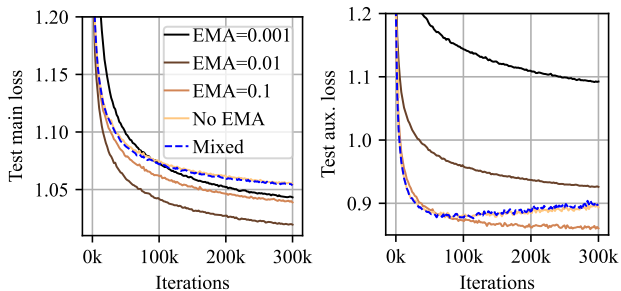

Figure 7. Effect of the EMA parameter ρ on Bloop's performance. We use the same next-token prediction losses as in Figure 5a, and display the training curves for a fixed $\lambda = 0.2$.

method. Choices between these two extremes ($\rho = 0.01$, or $\rho = 0.1$) lead to a tradeoff between main and auxiliary loss.

Discussion

A striking phenomenon that we observe in all our experiments is that PCGrad and DB work very similarly to the mixed method. We posit that this observation is due to the high gradient variance coming from the main loss, which is also what our theory predicts. Adding an EMA to reduce this variance leads to the Bloop method, which here has a different behavior to the other methods, often leading to improved Pareto fronts.

In the Appendix C, we describe an experiment where Bloop does not work better than the other methods. We attempted to train a ResNet to have a good performance on Imagenet and Cifar10, with a shared trunk and two classification heads. We found that all methods performed equally well; in that case, Bloop leads to the same Pareto front as the other method. Yet, once again, PCGrad and DB have the same practical performance as the mixed method.

Overall, adding an EMA to reduce variance in the projection direction is a simple idea that can have a big impact on gradient surgery methods.

Impact Statement

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

References

Bañón, M., Chen, P., Haddow, B., Heafield, K., Hoang, H., Esplà-Gomis, M., Forcada, M., Kamran, A., Kirefu, F., Koehn, P., et al. Paracrawl: Web-scale acquisition of parallel corpora. Association for Computational Linguistics (ACL), 2020.

Cao, J., Jiang, R., Abolfazli, N., Hamedani, E. Y., and Mokhtari, A. Projection-free methods for stochastic simple bilevel optimization with convex lower-level problem. *arXiv preprint arXiv:2308.07536*, 2023.

Caruana, R. Multitask learning. *Machine learning*, 28: 41–75, 1997.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2018.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, pp. 854–863. PMLR, 2017.

Cooper, Y. The loss landscape of overparameterized neural networks. *arXiv preprint arXiv:1804.10200*, 2018.

DeepMind, Babuschkin, I., Baumli, K., Bell, A., Bhupatiraju, S., Bruce, J., Buchlovsky, P., Budden, D., Cai, T., Clark, A., Danihelka, I., Dedieu, A., Fantacci, C., Godwin, J., Jones, C., Hemsley, R., Hennigan, T., Hessel, M., Hou, S., Kapturowski, S., Keck, T., Kemaev, I., King, M., Kunesch, M., Martens, L., Merzic, H., Mikulik, V., Norman, T., Papamakarios, G., Quan, J.,

Ring, R., Ruiz, F., Sanchez, A., Sartran, L., Schneider, R., Sezener, E., Spencer, S., Srinivasan, S., Stanojević, M., Stokowiec, W., Wang, L., Zhou, G., and Viola, F. The DeepMind JAX Ecosystem, 2020. URL http://github.com/google-deepmind.

Dempe, S., Dinh, N., and Dutta, J. Optimality conditions for a simple convex bilevel programming problem. *Variational Analysis and Generalized Differentiation in Optimization and Control: In Honor of Boris S. Mordukhovich*, pp. 149–161, 2010.

Dery, L. M., Dauphin, Y., and Grangier, D. Auxiliary task update decomposition: The good, the bad and the neutral. *arXiv preprint arXiv:2108.11346*, 2021.

Du, Y., Czarnecki, W. M., Jayakumar, S. M., Farajtabar, M., Pascanu, R., and Lakshminarayanan, B. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018.

Farhad, A., Arkady, A., Magdalena, B., Ondřej, B., Rajen, C., Vishrav, C., Costa-jussa, M. R., Cristina, E.-B., Angela, F., Christian, F., et al. Findings of the 2021 conference on machine translation (wmt21). In *Proceedings of the Sixth Conference on Machine Translation*, pp. 1–88. Association for Computational Linguistics, 2021.

Gong, C. and Liu, X. Bi-objective trade-off with dynamic barrier gradient descent. *NeurIPS 2021*, 2021.

Grangier, D., Ablin, P., and Hannun, A. Adaptive training distributions with scalable online bilevel optimization. *arXiv preprint arXiv:2311.11973*, 2023.

Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., Del Giorno, A., Gopi, S., Javaheripi, M., Kauffmann, P., de Rosa, G., Saarikivi, O., et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

He, Y., Feng, X., Cheng, C., Ji, G., Guo, Y., and Caverlee, J. Metabalance: improving multi-task recommendations via adapting gradient magnitudes of auxiliary tasks. In *Proceedings of the ACM Web Conference 2022*, pp. 2205–2215, 2022.

Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. Flax: A neural network library and ecosystem for jax, 2020. *URL http://github.com/google/flax*, 1.

Hotegni, S. S., Berkemeier, M., and Peitz, S. Multi-objective optimization for sparse deep multi-task learning. *arXiv preprint arXiv:2308.12243*, 2023.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pp. 795–811. Springer, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Lewis, D. D., Yang, Y., Russell-Rose, T., and Li, F. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Liu, C., Zhu, L., and Belkin, M. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022.

Luo, Z.-Q. and Tseng, P. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Sabach, S. and Shtern, S. A first order method for solving convex bilevel optimization problems. *SIAM Journal on Optimization*, 27(2):640–660, 2017.

Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Sun, Q., Cui, Y., Zhang, X., Zhang, F., Yu, Q., Luo, Z., Wang, Y., Rao, Y., Liu, J., Huang, T., et al. Generative multimodal models are in-context learners. *arXiv preprint arXiv:2312.13286*, 2023.

Terjék, D. Adversarial lipschitz regularization. *arXiv preprint arXiv:1907.05681*, 2019.

Tsuzuku, Y., Sato, I., and Sugiyama, M. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *Advances in neural information processing systems*, 31, 2018.

Wang, Z. and Tsvetkov, Y. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Xu, H., Xie, S., Tan, X. E., Huang, P.-Y., Howes, R., Sharma, V., Li, S.-W., Ghosh, G., Zettlemoyer, L., and Feichtenhofer, C. Demystifying clip data. *arXiv preprint arXiv:2309.16671*, 2023.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33: 5824–5836, 2020.

Appendix

A. Convergence analysis

In this appendix we provide proofs for the theoretical results of Section 3.

A.1. Proof of Proposition 1

In the following, we will prove the two implications in the proposition separately.

Small Bloop Vector → Near-Stationary Point. By orthogonality, we have $\|d\|^2 = \|g_{\mathrm{main}}\|^2 + \lambda^2 \|\pi(g_{\mathrm{aux}}; g_{\mathrm{main}})\|^2$. This implies immediately $\|g_{\mathrm{main}}\| \leq \varepsilon$ and $\|\pi(g_{\mathrm{aux}}; g_{\mathrm{main}})\| \leq \varepsilon\lambda^{-1}$ provided that $\|d\| \leq \varepsilon$.

Let us next consider the case where Assumption 1 holds and that the Hessian of L_{main} is M-Lipschitz continuous. With the local error bound, i.e., Assumption 1, we know there exists θ^* such that $\nabla L_{\mathrm{main}}(\theta^*) = 0$ and $\|\theta - \theta^*\| \leq c\|g_{\mathrm{main}}\|$. Performing a Taylor expansion with Lagrange form of the remainder of order 2, we obtain

$$\nabla L_{\mathrm{main}}(\theta) = \nabla L_{\mathrm{main}}(\theta^*) + \nabla^2 L_{\mathrm{main}}(\theta^*)(\theta - \theta^*) + \frac{1}{2}\nabla^3 L_{\mathrm{main}}(\theta')[\theta - \theta^*, \theta - \theta^*]$$

$$= \nabla^2 L_{\mathrm{main}}(\theta^*)(\theta - \theta^*) + \frac{1}{2}\nabla^3 L_{\mathrm{main}}(\theta')[\theta - \theta^*, \theta - \theta^*], \tag{7}$$

for some θ' that lies on the line that connects θ and θ^*. Using the M-Lipschitzness of $\nabla^2 L_{\mathrm{main}}$, the norm of $r = \nabla L_{\mathrm{main}}(\theta) - \nabla^2 L_{\mathrm{main}}(\theta)(\theta - \theta^*)$ can then be bounded by

$$\|r\| = \frac{1}{2}\nabla^3 L_{\mathrm{main}}(\theta')[\theta - \theta^*, \theta - \theta^*] \leq \frac{M}{2}\|\theta - \theta^*\|^2 \leq \frac{Mc^2}{2}\|g_{\mathrm{main}}\|^2 \ .$$

We now claim that the desired inequality holds true with

$$v = \frac{\langle g_{\mathrm{aux}}, g_{\mathrm{main}}\rangle}{\|g_{\mathrm{main}}\|^2}(\theta - \theta^*).$$

For this, we decompose

$$\frac{\langle g_{\mathrm{aux}}, g_{\mathrm{main}}\rangle}{\|g_{\mathrm{main}}\|^2}g_{\mathrm{main}} = \nabla^2 L_{\mathrm{main}}(\theta)v + \frac{\langle g_{\mathrm{aux}}, g_{\mathrm{main}}\rangle}{\|g_{\mathrm{main}}\|^2}r$$

Subsequently,

$$\|g_{\mathrm{aux}} - \nabla^2 L_{\mathrm{main}}(\theta)v\| \leq \left\| g_{\mathrm{aux}} - \frac{\langle g_{\mathrm{aux}}, g_{\mathrm{main}}\rangle}{\|g_{\mathrm{main}}\|^2}g_{\mathrm{main}} \right\| + \left\| \frac{\langle g_{\mathrm{aux}}, g_{\mathrm{main}}\rangle}{\|g_{\mathrm{main}}\|^2}r \right\|$$

$$\leq \|\pi(g_{\mathrm{aux}}; g_{\mathrm{main}})\| + \frac{Mc^2\|g_{\mathrm{aux}}\|\|g_{\mathrm{main}}\|}{2}$$

$$\leq \left(\lambda^{-1} + \frac{Mc^2}{2}\|g_{\mathrm{aux}}\| \right)\varepsilon.$$

Near-Stationary Point → Small Bloop Vector. Reciprocally, by plugging $\theta = \theta^* + \varepsilon v$ into (7), we get

$$g_{\mathrm{main}} = \nabla L_{\mathrm{main}}(\theta) = \varepsilon\nabla^2 L_{\mathrm{main}}(\theta^*)v + \frac{\varepsilon^2}{2}\nabla^3 L_{\mathrm{main}}(\theta')[v, v] = \varepsilon\nabla^2 L_{\mathrm{main}}(\theta^*)v + o(\varepsilon).$$

Moreover, by continuity of ∇L_{aux} and the optimality condition $\nabla L_{\mathrm{aux}}(\theta^*) = \nabla^2 L_{\mathrm{main}}(\theta)v$, we have $g_{\mathrm{aux}} = \nabla^2 L_{\mathrm{main}}(\theta^*)v + o(1)$ when ε goes to 0.

From here, we will prove $\lim_{\varepsilon \to 0} \pi(g_{\text{aux}}; g_{\text{main}}) = 0$ by distinguishing between two cases:

Case 1: $L_{\text{aux}}(\theta^*) = 0$. In other words, $\nabla^2 L_{\text{main}}(\theta^*)v = 0$. Thus $g_{\text{aux}} = o(1)$. $\pi(g_{\text{aux}}; g_{\text{main}})$ being a projection of g_{aux}, we have $\|\pi(g_{\text{aux}}; g_{\text{main}})\| \leq \|g_{\text{aux}}\|$. This show $\pi(g_{\text{aux}}; g_{\text{main}}) = o(1)$.

Case 2: $\nabla L_{\text{aux}}(\theta^*) \neq 0$. This indicates $\nabla^2 L_{\text{main}}(\theta^*)v \neq 0$. We use the formula

$$\pi(g_{\text{aux}}; g_{\text{main}}) = g_{\text{aux}} - \frac{\langle g_{\text{aux}}, g_{\text{main}} \rangle}{\|g_{\text{main}}\|^2} g_{\text{main}}. \tag{8}$$

With $\langle g_{\text{aux}}, g_{\text{main}} \rangle = \varepsilon \|\nabla^2 L_{\text{main}}(\theta^*)v\|^2 + o(\varepsilon)$ and $\|g_{\text{main}}\|^2 = \varepsilon^2 \|\nabla^2 L_{\text{main}}(\theta^*)v\|^2 + o(\varepsilon^2)$, we have

$$\frac{\varepsilon \langle g_{\text{aux}}, g_{\text{main}} \rangle}{\|g_{\text{main}}\|^2} = \frac{\|\nabla^2 L_{\text{main}}(\theta^*)v\|^2 + o(1)}{\|\nabla^2 L_{\text{main}}(\theta^*)v\|^2 + o(1)} = 1 + o(1),$$

where the last equality holds since $\|\nabla^2 L_{\text{main}}(\theta^*)v\|^2 \neq 0$.

On the other hand,

$$\frac{g_{\text{main}}}{\varepsilon} = \nabla^2 L_{\text{main}}(\theta^*)v + o(1).$$

We have thus

$$\frac{\langle g_{\text{aux}}, g_{\text{main}} \rangle}{\|g_{\text{main}}\|^2} g_{\text{main}} = \nabla^2 L_{\text{main}}(\theta^*)v + o(1).$$

Using (8) and $g_{\text{aux}} = \nabla^2 L_{\text{main}}(\theta^*)v + o(1)$, we deduce $\pi(g_{\text{aux}}; g_{\text{main}}) = o(1)$.

Conclude. In the two cases, we have shown $\lim_{\varepsilon \to 0} \pi(g_{\text{aux}}; g_{\text{main}}) = 0$. Moreover, we also have $\lim_{\varepsilon \to 0} g_{\text{main}} = 0$. Adding the two we get exactly $\lim_{\varepsilon \to 0} d(\theta^* + \varepsilon v) = 0$.

A.2. Proof of Theorem 2

Here, L_{main} and L_{aux} are the empirical risks

$$L_{\text{main}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} L_i(\theta) \quad \text{and} \quad L_{\text{aux}}(\theta) = \frac{1}{m} \sum_{j=1}^{m} L'_j(\theta).$$

We consider the Bloop method with SGD, which has an EMA g_{EMA}^t and parameters θ^t which are updated following

$$\text{Sample } i, j \sim \text{Uniform}$$
$$g_{\text{EMA}}^{t+1} = (1 - \rho)g_{\text{EMA}}^t + \rho \nabla L_i(\theta^t)$$
$$d^t = \nabla L_i(\theta^t) + \lambda \pi(\nabla L'_j(\theta^t); g_{\text{EMA}}^t)$$
$$\theta^{t+1} = \theta^t - \eta d^t$$

Our analysis works by controlling two quantities: the distance from the EMA to the full-batch train gradient

$$\phi_1^t = \mathbb{E} \left[\|g_{\text{EMA}}^{t+1} - \nabla L_{\text{main}}(\theta^t)\|^2 \right]$$

and the train loss

$$\phi_2^t = \mathbb{E} \left[L_{\text{main}}(\theta^t) \right].$$

Control of the EMA. For the EMA, we get by expanding

$$\phi_1^{t+1} = \mathbb{E} \left[\|g_{\text{EMA}}^t - \rho(g_{\text{EMA}}^t - \nabla L_i(\theta^t)) - \nabla L_{\text{main}}(\theta^t)\|^2 \right]$$
$$= (1 - \rho)^2 \mathbb{E} \left[\|g_{\text{EMA}}^t - \nabla L_{\text{main}}(\theta^t)\|^2 \right] + \rho^2 \mathbb{E} \left[\|\nabla L_i(\theta^t) - \nabla L_{\text{main}}(\theta^t)\|^2 \right]$$
$$\leq (1 - \rho)\mathbb{E} \left[\|g_{\text{EMA}}^t - \nabla L_{\text{main}}(\theta^t)\|^2 \right] + \rho^2 C^2$$

where C^2 upper bounds the train gradients variance and where $\rho < 1$. Let $a = g_{\text{EMA}}^t - \nabla L_{\text{main}}(\theta^{t-1})$ and $b = \nabla L_{\text{main}}(\theta^{t-1}) - \nabla L_{\text{main}}(\theta^t)$. Since the inequality $\|a + b\|^2 \leq (1+\delta)\|a\|^2 + (1+\delta^{-1})\|b\|^2$ holds true for all δ, we have specifically that

$$\|g_{\text{EMA}}^t - \nabla L_{\text{main}}(\theta^{t-1})\|^2 \leq (1+\delta)\phi_1^t + (1+\delta^{-1})L^2\eta^2\|d^{t-1}\|^2$$

for $\delta = \frac{\rho}{2}$. Using $(1-\rho)(1+\frac{\rho}{2}) \leq 1 - \frac{\rho}{2}$ then gives the descent lemma on the EMA:

$$\phi_1^{t+1} \leq \left(1 - \frac{\rho}{2}\right)\phi_1^t + \rho^2 C^2 + \frac{2L^2\eta^2}{\rho}\|d^{t-1}\|^2.$$

Next, we bound crudely $\|d^{t-1}\| \leq D$, and equalize the last two terms, i.e. take $\rho = \left(\frac{2L^2 D^2}{C^2}\right)^{\frac{1}{3}} \eta^{\frac{2}{3}}$, so that the descent on the EMA becomes

$$\phi_1^{t+1} \leq \left(1 - \frac{\rho}{2}\right)\phi_1^t + 2\rho^2 C^2$$

which in turn implies that

$$\phi_1^t \leq 4\rho C^2.$$

Control of the loss. The L-smoothness of L_{main} and the fact that $\mathbb{E}_{i,j}[d^t] = \nabla L_{\text{main}}(\theta^t) + \lambda\pi(\nabla L_{\text{aux}}; g_{\text{EMA}}^t)$ gives:

$$\phi_2^{t+1} \leq \phi_2^t - \eta\|\nabla L_{\text{main}}(\theta^t)\|^2 - \eta\lambda\langle\pi(\nabla L_{\text{aux}}; g_{\text{EMA}}^t), \nabla L_{\text{main}}(\theta^t)\rangle + \frac{L\eta^2}{2}\|d^t\|^2.$$

We omit expectation from the above formula for the ease of presentation, and we will continue doing so for this part of the proof. The annoying middle term is controlled by

$$
\begin{aligned}
-\eta\lambda\langle\pi(\nabla L_{\text{aux}}; g_{\text{EMA}}^t), \nabla L_{\text{main}}(\theta^t)\rangle &= -\eta\lambda\langle\pi(\nabla L_{\text{aux}}; g_{\text{EMA}}^t), \nabla L_{\text{main}}(\theta^t) - g_{\text{EMA}}^t\rangle \\
&\leq \eta\lambda B\|\nabla L_{\text{main}}(\theta^t) - \nabla L_{\text{main}}(\theta^{t+1})\| + \eta\lambda B\|\nabla L_{\text{main}}(\theta^{t+1}) - g_{\text{EMA}}^t\| \\
&\leq \eta^2\lambda L B\|d^t\| + \eta\lambda B\|\nabla L_{\text{main}}(\theta^{t+1}) - g_{\text{EMA}}^t\|
\end{aligned}
$$

where B upper bounds $\|\nabla L_{\text{aux}}\|$. The last $\mathbb{E}[\|d^t\|^2]$ is simply bounded by D^2. Hence we get the descent lemma on the train loss:

$$\phi_2^{t+1} \leq \phi_2^t - \eta\|\nabla L_{\text{main}}(\theta^t)\|^2 + \eta\lambda B\sqrt{\phi_1^t} + \eta^2\left(\frac{LD^2}{2} + \lambda L B D\right).$$

Plugging the rate for ϕ_1^t, we finally get

$$\phi_2^{t+1} \leq \phi_2^t - \eta\|\nabla L_{\text{main}}(\theta^t)\|^2 + \eta^{\frac{4}{3}} C_1 + \eta^2 C_2$$

for some constants $C_1, C_2 \geq 0$. In the above we have also used that

$$\mathbb{E}[\|\nabla L_{\text{main}}(\theta^{t+1}) - g_{\text{EMA}}^t\|]^2 \leq \mathbb{E}[\|\nabla L_{\text{main}}(\theta^{t+1}) - g_{\text{EMA}}^t\|^2].$$

Taking $\eta \leq \left(\frac{C_1}{C_2}\right)^{\frac{3}{2}}$ ensures that the last term is smaller than the previous, yielding the simple inequality:

$$\phi_2^{t+1} \leq \phi_2^t - \eta\|\nabla L_{\text{main}}(\theta^t)\|^2 + 2\eta^{\frac{4}{3}} C_1.$$

We now have two kinds of results depending on the context:

Non-convex result. Without further assumption, summing the previous inequalities for $t = 0 \ldots T-1$ gives

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla L_{\text{main}}(\theta^t)\|^2] \leq \frac{L_{\text{main}}(\theta^0)}{\eta T} + 2\eta^{\frac{1}{3}} C_1.$$

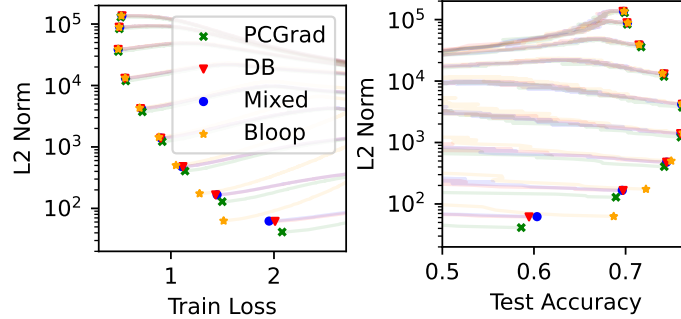Hence, taking $\eta \simeq T^{-\frac{3}{4}}$ gives a $O(T^{-\frac{1}{4}})$ rate.

Figure 8. Results of all methods on the imagenet + L2 problem. PCGrad and DB have similar performance to the mixed method.

PL-result. We here assume that L_{main} verifies the PL inequality $\frac{1}{2}\|\nabla L_{\mathrm{main}}(\theta)\|^2 \geq \mu L_{\mathrm{main}}(\theta)$, where we posit $\min L_{\mathrm{main}} = 0$ without loss of generalitiy. The descent lemma gives

$$\phi_2^{t+1} \leq (1 - 2\eta\mu)\phi_2^t + 2\eta^{\frac{4}{3}}C_1.$$

By unrolling it we obtain

$$\mathbb{E}[L_{\mathrm{main}}(\theta^T)] \leq (1 - 2\eta\mu)^T L_{\mathrm{main}}(\theta^0) + (1 - (1 - 2\eta\mu)^T)\frac{\eta^{\frac{1}{3}}C_1}{\mu}.$$

This shows a linear convergence to a radius proportional to $\eta^{\frac{1}{3}}$.

B. Experimental Details

In this appendix we report the missing details from Section 5.

Training smooth networks. For this experiment we use an MLP with ReLU activations. The features are of size $728 \to 256 \to 128 \to 10$. All the methods are trained with Adam optimizer at learning rate of 3×10^{-4} for 100 epochs and a cosine learning rate schedule. For consistency with the other classification experiments we also include 5 epochs of warm-up. The batch size is fixed at 256, and we take a grid of λ with $\log_{10}(\lambda) = -4, -3.5, \dots, -0.5, 0$.

Imagenet training with L2 regularization. For ImageNet training, we employ SGD with a batch size of 2048, Nesterov momentum of 0.9, and a learning rate of 0.8. This learning rate is derived by scaling the base rate of 0.1 by a factor of 8, corresponding to the ratio $2048/256$. Additionally, we apply a cosine learning rate schedule with 5 warm-up epochs and utilize random cropping and flipping for data augmentation during training. The network is trained for 100 epochs. This configuration is known to work well for the ResNet50 architecture that we are using here. The grid of λ is 14 uniform values in log scale between 10^{-6} and 10^{-2}, and 0. We display results for all methods in Figure 8 with a slightly smaller grid of λ's.

Multi-task learning with Cifar10Mnist. The overall setup for this problem is similar to that for Imagenet training, with the exceptions that we use a smaller architecture—ResNet18 instead of ResNet50, and a smaller batch size—256 instead of 2048. We also scale down the learning rate to 0.1 to account for the smaller batch size. The values of the trade-off parameter λ goes from 10^{-3} to 10^3 and are split equally on log scale. Unlike Adam, SGD does not adjust the learning rate scale automatically. This causes unstable training when λ is too large. We thus futher scale the learning rate 0.1 by $1/(1 + \lambda)$ for each independent run.

Next token prediction. Our model is a byte-level decoder-only transformer. It has 12 layers, 8 attention heads, a residual dimension of 256, and a feed-forward dimension 1024. We use a batch-size of 128 for both datasets, the optimizer is Adam with a learning rate of 0.002. We train the model for $300K$ iterations. The grid of λ consists of 16 values evenly spaced in log-space between 10^{-4} and 10, as well as 0.

Translation. Our model is an encore-decoder transformer. It has 6 encoder and decoder layers, 16 attention heads, a residual dimension of 1024, and a feed-forward dimension 4096. We use a batch-size of 256 for both datasets, the optimizer is

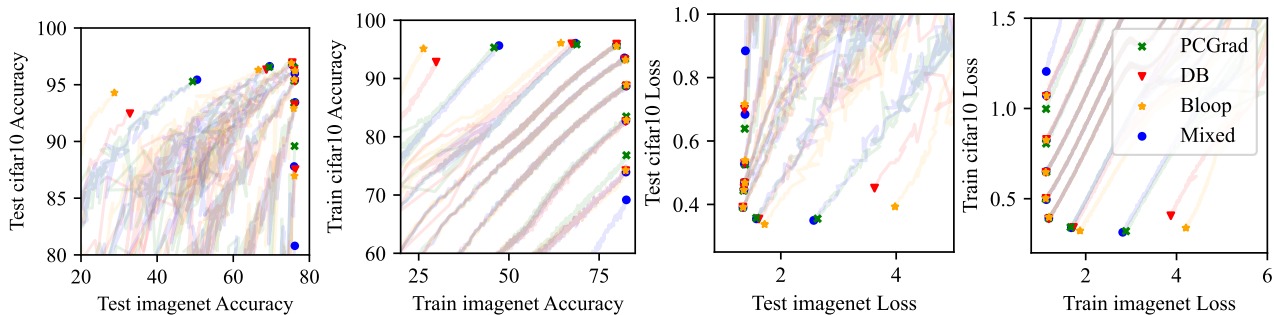*Figure 9.* Sample images from the Cifar10Mnist dataset.

Figure 10. Results on the Imagenet / Cifar10 experiment. All algorithms perform generally similarly except for very high values of λ, which leads to worse performance for all algorithms.

Adam with a learning rate of 0.0002. We train the model for $500K$ iterations. Our implementation is derived from the flax example (Heek et al.). The grid of λ consists of 16 values evenly spaced in log-space between 10^{-4} and 10, as well as 0.

C. Additional Experiment

We present the results of another experiment, where all methods, including Bloop, gave similar Pareto fronts. Here, we aim to perform classification on both the Imagenet and the CIFAR-10 datasets. The network is a ResNet50 with with two separate classification heads. This problem sits in the middle ground between the multi-task learning and the joint dataset training problem that we describe in Section 5: we have two separate datasets for the two distinct tasks. Similar to before, the main loss is the training loss on the larger dataset, i.e., Imagenet, and the auxiliary loss is the training loss on the smaller dataset, i.e. Cifar10. We choose λ to be equally split on log scale from 10^{-3} to 10. The remaining configurations follow the experiment of Imagenet training with L2 regularization, except that we also scale the learning rate by $1/(1 + \lambda)$ to avoid instability as in the multi-task experiment.

The results are shown in Figure 10. Unlike the experiments of Section 5, there is little trade-off between the two tasks. We can increase accuracy on CIFAR-10 without sacrificing performance on Imagenet. For this reason, there are only very few points at the Pareto front and all methods perform similarly at these points. We posit that here, the two losses are not conflicting enough to see the gradient surgery methods have an edge.