Journal of Machine Learning Research 23 (2022) 1-57 Submitted 6/20; Revised 04/21; Published 2/22

A Stochastic Bundle Method for Interpolating Networks

Alasdair Paren * ALASDAIR.PAREN@GMAIL.COM
Department of Engineering Science
University of Oxford
Oxford, UK

Leonard Berrada * LBERRADA@ROBOTS.OX.AC.UK
Department of Engineering Science
University of Oxford
Oxford, UK

Rudra P. K. Poudel RUDRA.POUDEL@CRL.TOSHIBA.CO.UK
Cambridge Research Laboratory,
Toshiba Europe Ltd,
Cambridge, UK.

M. Pawan Kumar PAWAN@ROBOTS.OX.AC.UK
Department of Engineering Science
University of Oxford
Oxford, UK.

Editor: Julien Mairal

Abstract

We propose a novel method for training deep neural networks that are capable of interpolation, that is, driving the empirical loss to zero. At each iteration, our method constructs a stochastic approximation of the learning objective. The approximation, known as a bundle, is a pointwise maximum of linear functions. Our bundle contains a constant function that lower bounds the empirical loss. This enables us to compute an automatic adaptive learning rate, thereby providing an accurate solution. In addition, our bundle includes linear approximations computed at the current iterate and other linear estimates of the DNN parameters. The use of these additional approximations makes our method significantly more robust to its hyperparameters. Based on its desirable empirical properties, we term our method Bundle Optimisation for Robust and Accurate Training (BORAT). In order to operationalise BORAT, we design a novel algorithm for optimising the bundle approximation efficiently at each iteration. We establish the theoretical convergence of BORAT in both convex and non-convex settings. Using standard publicly available data sets, we provide a thorough comparison of BORAT to other single hyperparameter optimisation algorithms. Our experiments demonstrate BORAT matches the state-of-the-art generalisation performance for these methods and is the most robust.

Keywords: Bundle Methods, Stochastic Optimisation, Neural Network Optimisation, Interpolation, Adaptive Learning-rate

*Authors contributed equally to this work.

1. Introduction

Training a deep neural network (DNN) is a challenging optimization problem: it involves minimizing the average of many high-dimensional non-convex loss functions. In practice, the main algorithms utilised are Stochastic Gradient Descent (SGD) (Robbins and Monro, 1951) and adaptive gradient methods such as AdaGrad (Duchi et al., 2011) or Adam (Kingma and Welling, 2014). It has been observed that SGD tends to provide better generalization performance than adaptive gradient methods (Wilson et al., 2017). However, the downside of SGD is that it requires the manual design of a learning-rate schedule. Many forms of schedule have been proposed in the literature, including piece wise constant (Huang et al., 2017), geometrically decreasing (Szegedy et al., 2015) and warm starts with cosine annealing (Loshchilov and Hutter, 2017). Examples of these schemes are plotted in Figure 1. Consequently, practitioners who wish to use SGD in a novel setting need to select a learning-rate schedule for their learning task. To that end, they first need to choose the parameterization of the schedule (e.g. picking one of the examples given above). Then they need to tune the corresponding parameters to get good empirical performance. This typically results in a cross-validation that searches over many critical and sensitive hyper-parameters. For example, a piece wise linear scheme requires a starting learning rate value, a decay factor and a list or metric to determine at which points in training to decay the learning rate. Due to the high dimensionality of this search space performing a grid search can mean training a large number of models. This number increases exponentially in combination with other hyperparameters such as regularisation and batch size. Thus, finding an SGD learning rate schedule that produces strong generalisation performance for a new task is time and computationally demanding, often requiring hundreds of GPU hours.

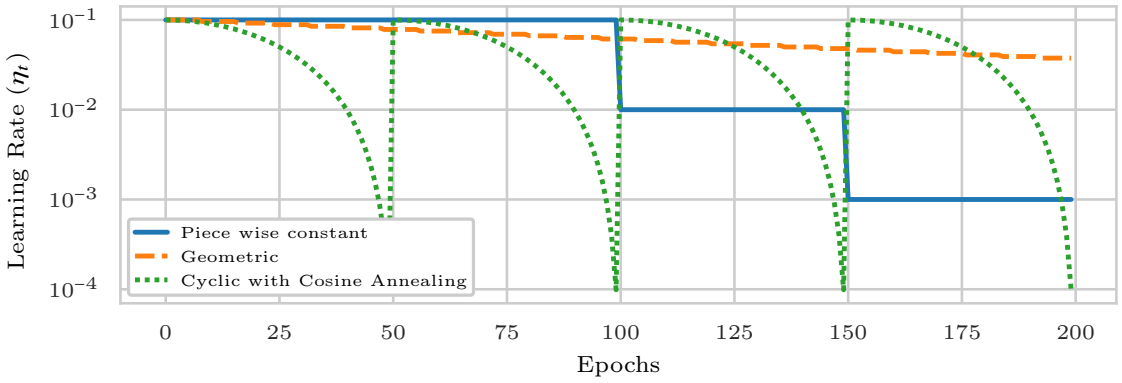

Figure 1: SGD Learning rate schedules proposed in the literature

In this work, we alleviate this issue by presenting a family of algorithms that achieve comparable generalisation performance to SGD with a highly refined learning rate schedule, while requiring far less tuning of hyperparameters. This in turn leads to a reduction in the time, cost and energy required when finding a highly accurate network for a new task.
In more detail, we present a novel family of proximal algorithms for the optimisation of DNNs that are capable of interpolation. A DNN is said to interpolate a data set if it has the ability

to simultaneously drive the loss of all the training samples in a data set to their minimum value. Thus a lower bound of the objective function is known: for instance, it is close to zero for a model trained with the cross-entropy loss. Our algorithms approximate the loss within each proximal problem by a bundle, that is, a point-wise maximum over linear functions. By including the interpolation lower bound within this bundle, we obtain the following two modelling benefits: (i) our model more closely mimics the true loss function than existing baselines like SGD, and; (ii) the learning rate gets automatically re-scaled at each iteration of the algorithm, thereby providing accurate updates. By increasing the number of linear approximations in the bundle, the true loss is better modelled. As an upshot, we obtain additional stability to optimisation and task-specific hyperparameters. Based on its highly desirable empirical properties, we term these methods Bundle optimization for Robust and Accurate Training (BORAT).
All the variants of BORAT use a single learning rate hyperparameter that requires minimal tuning. In particular, the learning rate hyperparameter is kept constant throughout the training procedure, unlike the learning rate of SGD that needs to be decayed for good generalization. The BORAT family of algorithms obtain state-of-the-art empirical performance for single hyperparameter training of neural networks.

Contributions

- We design a family of adaptive algorithms that have a single hyperparameter that does not need any decaying schedule. In contrast, the related APROX (Asi and Duchi, 2019) and L4 (Rolinek and Martius, 2018) use respectively two and four hyperparameters for their learning-rate.
- For the deep learning setting we establish a link between stochastic optimisation methods with adaptive learning rates and proximal bundle methods.
- We provide convergence rates in various stochastic convex settings and for a class of non-convex problems.
- We derive a novel algorithm for solving small quadratic programs where the constraints define the probability simplex. This algorithm permits a parallel implementation allowing for efficient solution on modern hardware.
- We empirically demonstrate the increased stability to hyperparameters when increasing the bundle size. We show this on the CIFAR100 and Tiny ImageNet data sets.
- We achieve state-of-the-art results for learning a differentiable neural computer; training variants of residual networks on the SVHN and CIFAR data sets, and training a Bi-LSTM on the Stanford Natural Language Inference data set.

A preliminary version of this work appeared in the proceedings of ICML 2020 (Berrada et al., 2019b). While this previous work has considered bundles of size 2 resulting in the ALI-G algorithm detailed with in Section 3.3. This article significantly differs from the previous work by (i) considering bundles of size greater than 2; (ii) introducing an novel algorithm to compute the exact solution of each bundle; (iii) investigating the robustness towards hyperparameters; and (iv) showing how the use of large bundles permits easy application of BORAT to challenging non-smooth losses.

2. Preliminaries

Loss Function. We consider a supervised learning task where the model is parameterized by $\mathbf{w} \in \mathbb{R}^d$. Usually, the objective function can be expressed as an expectation over $z \in \mathcal{Z}$, a random variable indexing the samples of the training set:

$$f(\mathbf{w}) \triangleq \mathbb{E}_{z \in \mathcal{Z}}[\ell_z(\mathbf{w})], \tag{1}$$

where each ℓ_z is the loss function associated with the sample z. We assume that each ℓ_z admits a known lower bound B. For simplicity we will often assume this lower bound is 0, which is the case for the large majority of loss functions used in machine learning. For instance, suppose that the model is a deep neural network with weights $\mathbf{w}$ performing classification. Then for each sample z, $\ell_z(\mathbf{w})$ can represent the cross-entropy loss, which is always non-negative. Other non-negative loss functions include the structured or multi-class hinge loss, and the ℓ_1 or ℓ_2 loss functions for regression. Note that it is always possible to subtract a non-zero lower bound B from the loss function to define a new equivalent problem that satisfies the aforementioned assumption.

Interpolation. In this work we consider DNNs that can interpolate the data. Formally, we assume:

$$\exists\ \mathbf{w}_\star :\ \forall z \in \mathcal{Z}, \ell_z(\mathbf{w}_\star) \leq \epsilon, \tag{2}$$

where ϵ is a tolerance on the amount of error in the interpolation assumption. We will often want to make reference to the case when $\epsilon = 0$. Following previous work (Ma et al., 2018b) we will refer to this setting as perfect interpolation. The interpolation property is satisfied in many practical cases, since modern neural networks are typically trained in an over-parameterized regime where the parameters of the model far exceed the size of the training data (Li et al., 2020). Additionally most modern DNN architectures can be easily increased in size and depth, allowing them to interpolate all but the largest data sets. Note, the data has to be labelled consistently for this to be possible. For instance it is impossible to interpolate a data set with two instances of the same image that have two different labels.

Regularisation. It is often desirable to encourage generalisation by the addition of a regularisation function $\phi(\mathbf{w})$ to the objective. Typical choices for ϕ include $\lambda\|\mathbf{w}\|_2$ and $\lambda\|\mathbf{w}\|_1$ where λ governs the strength of the regularisation. However, in this work we incorporate such regularisation as a constraint on the feasible domain: $\Omega = \left\{\mathbf{w} \in \mathbb{R}^d :\ \phi(\mathbf{w}) \leq r\right\}$ for some value of r. In the deep learning setting, this will allow us to assume that the objective function can be driven close to zero without unrealistic assumptions about the value of the regularisation term for the final set of parameters. Our framework can handle any constraint set Ω on which Euclidean projections are computationally efficient. This includes the feasible set induced by ℓ_2 regularization: $\Omega = \left\{\mathbf{w} \in \mathbb{R}^d :\ \|\mathbf{w}\|_2^2 \leq r\right\}$, for which the projection is given by a simple rescaling of $\mathbf{w}$. Finally, note that if we do not wish to use any regularization, we define $\Omega = \mathbb{R}^d$ and the corresponding projection is the identity.

Problem Formulation. The learning task can be expressed as the problem $(\mathcal{P})$ of finding a feasible vector of parameters $\mathbf{w}_\star \in \Omega$ that minimizes f:

$$\mathbf{w}_\star \in \operatorname*{argmin}_{\mathbf{w} \in \Omega} f(\mathbf{w}). \tag{$\mathcal{P}$}$$

We refer to the minimum value of f over Ω as $f_\star$: $f_\star \triangleq \min_{\mathbf{w} \in \Omega} f(\mathbf{w})$.

Proximal Perspective of Projected Stochastic Gradient Descent. In order to best introduce BORAT, we first detail the proximal interpretation of projected stochastic gradient descent (PSGD). The PSGD algorithm can be seen as solving a sequence of proximal problems. Within each proximal problem, a minimisation is performed over an approximate local model of the loss. This approximation is the first order Taylor's expansion of ℓ_{z_t} around the current iterate and a proximal term. At time step t, the PSGD proximal problem has the form:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\Omega}{\operatorname{argmin}} \left\{ \frac{1}{2\eta_t}\|\mathbf{w} - \mathbf{w}_t\|^2 + \ell_{z_t}(\mathbf{w}_t) + \nabla\ell_{z_t}(\mathbf{w}_t)^\top(\mathbf{w} - \mathbf{w}_t) \right\}. \tag{3}$$

Here $\mathbf{w}_t$ is the current iterate, z_t is the index of the sample chosen and η_t is the learning rate. For convex Ω, problem (3) can be solved in two steps: first solving the unconstrained problem and then using Euclidean projection onto Ω. Setting $\Omega = \mathbb{R}^d$ removes the need for projection and we recover SGD. When clear from the context we will use SGD to refer to both projected and un-projected variants. To solve the unconstrained problem one only needs to set the gradient to zero to recover the familiar closed form SGD update.

3. The BORAT Algorithm

In this section we detail the BORAT algorithm. We start by introducing BORAT's proximal problem that is exactly solved at each iteration. We explain its advantages and disadvantages in relation to the SGD proximal problem (3). Each BORAT proximal problem is best solved in the dual. Hence we next introduce the dual problem, which permits a far more efficient solution due to its low dimensionality. We then consider a special case of BORAT with minimal bundle size which we call Adaptive Learning-rates for Interpolation with Gradients (ALI-G). ALI-G permits a closed form solution and results in an automatically scaled gradient descent step. Specifically, it recovers a stochastic variant of the Polyak step size (Polyak, 1969), which offers competitive results in practice. This special case is used extensively in our experiments and in our analysis to establish the convergence rate of BORAT. Lastly, we detail our novel direct method for efficiently solving the general dual problem. This algorithm exploits the small size of the bundle to compute the exact optimum by solving a finite number of linear systems, removing the need for an inner iterative optimisation algorithm.

3.1 Primal Problem

For tackling problems of type $(\mathcal{P})$ we identify two deficiencies in the proximal view of SGD (3). First, the approximation of the loss permits negative values even though the loss for $(\mathcal{P})$ is defined to be non-negative. Second, the accuracy of a linear model quickly deteriorates for functions with high curvature away from the site of the approximation. Due to this crude model, the selection of η_t for all t is critical for achieving good performance with SGD. BORAT aims to address these deficiencies by using a model composed of a point-wise maximum over N linear approximations to better model the loss. One of the linear approximations is chosen to be a constant function equal to the loss lower bound, that is, 0. By including this linear approximation, we address the first deficiency. The second deficiency is addressed by making use of the remaining $N-1$ linear approximations. These extra approximations allow us to model variation over the parameter space and positive

curvature of the loss. A model of this form in combination with a proximal term is commonly known as a bundle of size N. The main disadvantage of bundles is that they require multiple gradient evaluations to be performed and then held in memory. Hence we in this work only consider $N \leq 5$, except where mentioned otherwise. Each linear approximation of the loss is constructed at a point $\hat{\mathbf{w}}_t^n$ using a different loss function $\ell_{z_t^n}$. The subscript t indicates the iteration number, and n indexes over the N linear approximations. With this notation we first introduce a bundle of size 1 as:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\Omega}{\operatorname{argmin}} \left\{ \frac{1}{2\eta}\|\mathbf{w} - \mathbf{w}_t\|^2 + \ell_{z_t^1}(\hat{\mathbf{w}}_t^1) + \nabla \ell_{z_t^1}(\hat{\mathbf{w}}_t^1)^\top (\mathbf{w} - \hat{\mathbf{w}}_t^1) \right\}. \tag{4}$$

If we set $\hat{\mathbf{w}}_t^1$ to $\mathbf{w}_t$ we recover the SGD proximal problem. Thus SGD effectively uses a bundle of size $N = 1$. We next introduce an expanded expression for a bundle of size N, before showing how to convert this into the compact form of $\max_{n\in[N]} \left\{ \boldsymbol{a}^{n\top}(\mathbf{w} - \mathbf{w}_t) + b^n \right\}$. Within a bundle each linear approximation is formed around a different point $\hat{\mathbf{w}}_t^n$. Hence in order to write each linear approximation in the aforementioned compact form we split each linear term in two. The first piece is a constant term, that does not depend on $\mathbf{w}$, and is a multiple of the distance between the current iterate and the centre of each approximation $\hat{\mathbf{w}}_t^n - \mathbf{w}_t$. The second term depends on the distance $\mathbf{w} - \mathbf{w}_t$, for all linear approximations. This gives the following expanded form for a bundle of size N as:

$$\max_{n\in[N]} \left\{ \ell_{z_t^n}(\hat{\mathbf{w}}_t^n) - \nabla \ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top (\hat{\mathbf{w}}_t^n - \mathbf{w}_t) + \nabla \ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top (\mathbf{w} - \mathbf{w}_t) \right\}, \tag{5}$$

where we use the notation $[N]$ to define the set of integers $\{1, ..., N\}$. If we define $b_t^n = \ell_{z_t^n}(\hat{\mathbf{w}}_t^n) - \nabla \ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top (\hat{\mathbf{w}}_t^n - \mathbf{w}_t)$, we can thus simplify the expression into the desired compact form. We now introduce the BORAT proximal problem at time t with a bundle of size N, which can be stated as:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\Omega}{\operatorname{argmin}} \left\{ \frac{1}{2\eta}\|\mathbf{w} - \mathbf{w}_t\|^2 + \max_{n\in[N]} \left\{ \nabla \ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top (\mathbf{w} - \mathbf{w}_t) + b_t^n \right\} \right\}. \tag{6}$$

For BORAT we always set $\hat{\mathbf{w}}_t^1 = \mathbf{w}_t$. We additionally use the last linear approximation to enforce the lower bound on the loss. This is done by setting $\nabla \ell_{z_t}(\hat{\mathbf{w}}_t^N) = [0, ..., 0]^\top$, $b_t^N = \mathrm{B} = 0$. We give details on how we select $\hat{\mathbf{w}}_t^n$ for $n \in \{2, ..., N-1\}$ in Section 3.5. Thus each bundle is composed of $N-1$ linear approximations of the function, and the lower bound on the loss. These linear approximations of the loss need to be stored in memory during each step. Hence, in order to fit on a single GPU we only consider small bundle sizes in this work ($N \leq 5$). For clarity we depict a 1D example for a bundle with $N = 3$ in Figure 2.
Unlike SGD, the BORAT proximal problem (6) is not smooth and hence cannot be solved by simply setting the derivatives to zero. Instead we choose to solve each proximal problem in the dual.

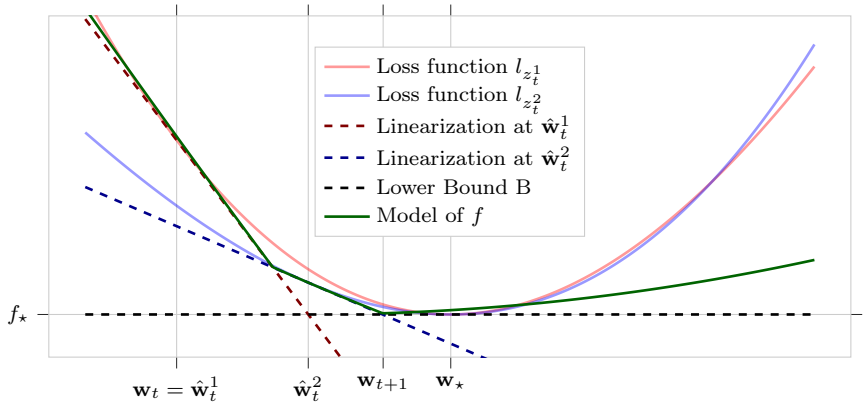

Figure 2: *Illustration of a BORAT bundle ($N = 3$) in 1D, shown in green. Two stochastic samples $\ell_{z_t^1}$ and $\ell_{z_t^2}$ of the loss function f are shown in red and blue (solid lines). The bundle is formed of the max of three linear approximations (dashed lines) and a proximal term. Two of these linear approximations are formed using the loss functions $\ell_{z_t^1}$ and $\ell_{z_t^2}$, and the last enforces the known lower bound on the loss. Here the BORAT approximation gives a more accurate model than approximation used by ALI-G, which would only include the linearization at $\hat{w}_t^1$ and the lower bound B. In this simple example this improved accuracy allows for a larger step to be taken towards the minimum.*

3.2 Dual Problem

The dual of (6) is a constrained concave quadratic maximisation over N dual variables $\alpha^1, ..., \alpha^N$, and can be concisely written as follows (see supplementary material for derivation):

$$\boldsymbol{\alpha}_t = \underset{\boldsymbol{\alpha} \in \Delta_N}{\operatorname{argmax}} D(\boldsymbol{\alpha}), \text{ where } \quad D(\boldsymbol{\alpha}) = -\frac{\eta}{2} \boldsymbol{\alpha}^\top A_t^\top A_t \boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \boldsymbol{b}_t. \tag{7}$$

Here A_t is a $N \times d$ matrix whose n^{th} row is $\nabla \ell_{z_t}(\hat{\mathbf{w}}_t^n)$. We define $\boldsymbol{b}_t = [b_t^1, ..., b_t^N]^\top$, $\boldsymbol{\alpha} = [\alpha^1, \alpha^2, \ldots, \alpha^N]^\top$ and Δ_N is a probability simplex over the N variables. The dual problem (7) has a number of features that make it more appealing for optimisation than the primal (6). First, the primal problem is defined over the parameters space $\mathbf{w} \in \mathbb{R}^d$, where p is in the order thousands if not millions for modern DNNs. In contrast, the dual variables are of dimension N, where is N typically a small number > 10 due to the memory requirements. Second, the dual is smooth and hence allows for faster convergence with standard optimisation techniques. Furthermore, as will be seen shortly, we use the fact that the dual feasible region is a tractable probability simplex to design a customised algorithm for its solution. We detail this algorithm in this Section 3.6. Once we have found the dual solution α_t, we recover the following update from the KKT conditions:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta A_t \boldsymbol{\alpha}_t. \tag{8}$$

The form of the update (8) deserves some attention. Since each row of A_t is either the gradient of the loss $\ell_{z_t^n}$ or a zero vector, and α_t belongs to the probability simplex, the update step moves in the direction of a non-negative linear combination of negative gradients $-\nabla \ell_{z_t^n}(\hat{\mathbf{w}}_t^n)$. Due to the definitions of $\nabla \ell_{z_t}(\hat{\mathbf{w}}_t^N) = [0, ..., 0]^\top$ and $b_t^N = 0$, any weight given

to α^N reduces the magnitude of the resulting step. This has the effect that as the loss value gets close to zero BORAT automatically decreases the size of the step taken.

3.3 ALI-G (N=2)

We now consider a special case of BORAT with $N = 2$. Here the bundle is the point wise maximum over the linear approximation of the loss around the current point and the global lower bound B, which we assume is 0. This special case is worthy of extra attention for the following four reasons. First, this special case only requires one gradient evaluation per step and has a similar time complexity to SGD. Second, it admits a closed form solution. Third, we use this special case extensively in our analysis of the convergence rate of BORAT. Fourth, given the definitions $\hat{\mathbf{w}}_t^1 = \mathbf{w}_t$ and $\hat{\mathbf{w}}_t^N = 0$, if we set $N = 2$ we recover an algorithm that simply scales the SGD learning rate. Specifically, it automatically scales down a maximal learning rate η by a factor $\alpha^1 \in [0, 1]$ to an appropriate value close to optimality. This is clear from the simplified version of Equation (8), which has the following form:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha^1 \eta \nabla l_{z_t}(\mathbf{w}_t). \tag{9}$$

Hence we will call this special case Adaptive Learning-rates for Interpolation with Gradients (ALI-G). For ALI-G the primal problem (6) simplifies to the following:

$$\underset{\mathbf{w}\in\Omega}{\operatorname{argmin}} \left\{ \frac{1}{2\eta}\|\mathbf{w} - \mathbf{w}_t\|^2 + \max\{0, \ell_{z_t}(\mathbf{w}_t) + \nabla \ell_{z_t}(\mathbf{w}_t)^\top (\mathbf{w} - \mathbf{w}_t)\} \right\}. \tag{10}$$

Likewise, the dual problem (7) can be reduced to the following:

$$\alpha^1 = \underset{\alpha^1\in[0,1]}{\operatorname{argmax}} \left\{ -\frac{\eta}{2}\|\alpha^1 \nabla l_{z_t}(\mathbf{w}_t)\|^2 + \alpha^1 l_{z_t}(\mathbf{w}_t) \right\}. \tag{11}$$

The ALI-G dual is a maximisation over a constrained concave function in one dimension. Hence we can obtain the optimal point by projecting the unconstrained solution on to the feasible region. This results in the following closed form solution:

$$\alpha^1 = \min \left\{ \frac{l_{z_t}(\mathbf{w}_t)}{\eta\|\nabla l_{z_t}(\mathbf{w}_t)\|^2}, 1 \right\}. \tag{12}$$

This value of α^1 is then used in (9). The ALI-G update can be viewed as a stochastic analog of the Polyak step size (Polyak, 1969) with the addition of a maximal value η. Recall that, from the interpolation assumption, we have $f_\star = 0$. The ALI-G update is computationally cheap with the evaluation of $\|\nabla l_{z_t}(\mathbf{w}_t)\|^2$ being the only extra computation required over SGD. Hence when the interpolation assumption holds we suggest that ALI-G can be easily used in place of SGD.

3.4 The Advantage of Bundles with more than Two Pieces

While ALI-G has many favourable qualities, its local model of the loss is still crude. We next give two motivating examples to help demonstrate why using a more complex model of the loss often increases the robustness to η. Thus it may prove useful to use larger values of N

in settings that are sensitive to step size η.
In the convex setting any function can be perfectly modelled by the point-wise maximum over an infinite number of linear approximations. While intractable, performing a minimisation over this model would recover the true optimum by definition. With this perfect model any value of η could be used. Setting η to a large enough value would recover the optimum in a single step. This example demonstrates that, at least asymptotically as the accuracy of the local model increases we can expect a reduced dependence on the correct scale of the step size.

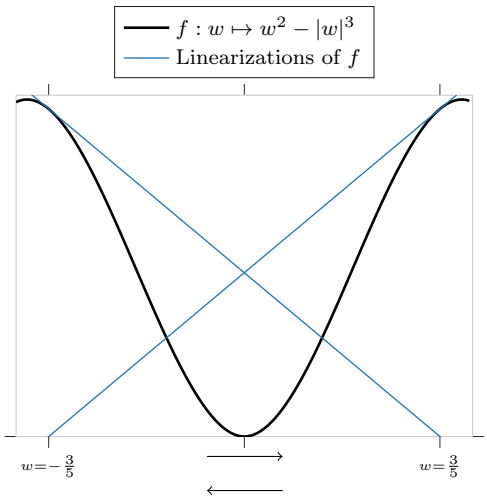

Figure 3: *A simple example where the ALI-G step-size oscillates due to non-convexity. For this problem, ALI-G only converges ifs its maximal learning-rate η is less than* 10*. By contrast for the same example BORAT with $N > 2$ converges for all values of η. Additionally for $\eta \geq 10$ it converges to the optimum in a single update.*

Figure 3 provides a non-convex motivating example for use of larger values of N. Here we demonstrate a 1D symmetric function where ALI-G does not converge for large η. Instead it oscillates between the two values $w = -3/5$ and $w = 3/5$. However, if we were to use a bundle with a $N \geq 3$ our model of the loss would include both the linear approximations at $w = -3/5$ and $w = 3/5$ simultaneously and hence when minimising over this model we converge to the optimum for any value of η. While this is a carefully constructed synthetic example it highlights why we would expect a more accurate model of the loss can help to reduce the dependence on the step size.

3.5 Selecting Additional Linear Approximations or the Bundle

When constructing bundles of size larger than two, we are faced with two design decisions regarding how to select additional linear approximations to add to the bundle. First, where in parameter space should we construct the additional linear approximations $\hat{\mathbf{w}}_t^n$? And second, should we use the same mini-batch of data when constructing the stochastic linear approximations, or should we sample a new batch to evaluate each linear approximation?

Selecting $\hat{\mathbf{w}}_t^n$. Ideally we would select the location of the linear approximations $\hat{\mathbf{w}}_t^n$ for $n \in \{2, ..., N-1\}$ in order to maximise the progress made towards $\mathbf{w}_\star$ at each step. However, without the knowledge of $\mathbf{w}_\star$ a priori, due to the high dimensional and non-convex nature of problem $(\mathcal{P})$ this is infeasible. Instead we make use of a a heuristic. Inspired by the work of previous bundle methods for convex problems (Smola et al., 2007, Asi and Duchi, 2019) we select $\hat{\mathbf{w}}_t^n$ using the following method:

$$\hat{\mathbf{w}}_t^n = \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^d} \left\{ \frac{1}{2\eta} \|\mathbf{w} - \mathbf{w}_t\|^2 + \max_{k \in [n-1]} \left\{ \nabla \ell_{z_t^k}(\hat{\mathbf{w}}_t^k)^\top (\mathbf{w} - \mathbf{w}_t) + b_t^k \right\} \right\}. \tag{13}$$

In other words we construct the bundle by recursively adding linear approximations centred at the current optimum. This method of selecting additional linear approximations is appealing as it requires no extra hyperparameters and helps refine the approximation in the region of parameter space that would be explored by the next update.

Re-sampling z for additional linear approximations. When constructing additional linear approximations we choose to re-sample z. Concretely, we use a new mini-batch of data to construct each stochastic linear approximation. While it is possible to construct all $N-1$ non-zero linear approximations using the same batch of data we find this does not work well in practice. Indeed, such a method behaves similarly to taking multiple consecutive steps of SGD on the same mini-batch, which tends to produce poor optimisation performance.

Summary. We now summarise the bundle construction procedure for $N > 2$. We construct a bundle around $\mathbf{w}_t$ by first using two linear approximations, one centred at $\mathbf{w}_t$ and the second given by a known lower bound on the loss. We then sequentially add linear approximations until we have N. These extra linear approximations are constructed one at a time using new batches of data and centred around the point that is the current minimizer of the bundle. We note that each parameter update of BORAT uses $N-1$ batches of data. Therefore BORAT updates the parameters $N-1$ fewer times than SGD in an epoch (given the same batch-size). Once the bundle is fully constructed we update $\mathbf{w}_t$. At this stage we apply momentum (if enabled) and project back on the feasible set Ω.
The construction of the bundle requires solving a minimization problem for each newly added piece when $N \geq 2$. Therefore it is critical that such a problem gets solved very efficiently. We next detail how we do this by solving the corresponding dual problem for $N \geq 2$.

3.6 Efficient Dual Algorithm to Compute $N \geq 2$ Linear Pieces

In the general case ($N > 2$), the dual has more than 2 degrees of freedom and can not be written as a 1D minimisation. Thus the method derived for the case $N = 2$ is no longer applicable. This means it is not possible to obtain a simple closed form update. We must instead run an inner optimisation to solve (7) at each step. Many methods exist for maximising a concave quadratic objective over a simplex. Two algorithms particularly well suited to problems of the form (7) are (Frank and Wolfe, 1956) or Homotopy Methods (Bach et al., 2011). However, we propose a novel algorithm that exploits the fact that N is small to find the maximum directly. This method decomposes the problem of solving (7) into several sub-problems which provides two computational conveniences. First, the BORAT algorithm repeatedly searches for solutions to a bundle with only one newly added linear piece since

the last search. As one might expect, this task shares a great number of sub-problems with the previous solution and allows for much of the computation to be reused. Second, our dual algorithm allows for a parallel implementation, which makes it very fast on the hardware commonly used for deep learning. To illustrate the efficiency of our dual method, the run time of the dual solution, that is finding $\boldsymbol{\alpha}_t$ once we have constructed (7), takes less than 5% of the time spent in the call of the optimiser. Note due to the large size of the the networks and the small size of N the majority of the call time is dominated computing $A_t^\top A_t$ and $A_t \boldsymbol{\alpha}_t$.

We now formally introduce our dual solution algorithm. Our method uses the observation that at the optimal solution in a simplex the partial derivatives will be equal for all nonzero dimensions. This observation can be formally stated as:

Proposition 1 (*Simplex Optimally Conditions*)

Let $F : \mathbb{R}^N \to \mathbb{R}$ be a concave function. Let us define $\boldsymbol{\alpha}_* = \operatorname{argmax}_{\boldsymbol{\alpha} \in \Delta} F(\boldsymbol{\alpha})$. Then there exists $c \in \mathbb{R}$ such that:

$$\forall n \in [N] \text{ such that } \alpha_*^n > 0, \text{ we have: } \left.\frac{\partial F(\boldsymbol{\alpha})}{\partial \alpha^n}\right|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_*} = c. \tag{14}$$

In other words, the value of the partial derivative is shared among all coordinates of $\boldsymbol{\alpha}_*$ that are non-zero.

This proposition can be easily proved by contradiction. If the partial derivatives are not equal, then moving in the direction of the largest would result in an increase in function value. Likewise moving in the negative direction would produce a decrease. Hence the current point cannot be optimal. Please see Appendix B for a formal proof of Proposition 1.

In the following paragraphs we explain how this proposition can be used to break up the task of solving problem (7) into $2^N - 1$ subproblems. Given a unique subset I of non-zero dimensions of $\boldsymbol{\alpha}$ Each of these subproblems involves finding the unconstrained optimum and checking if this point lies within the simplex. We now give an example of a single subproblem. To simplify notation, let $Q \triangleq \eta A_t^\top A_t$. We note that:

$$\left.\frac{\partial D(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_*} = -Q\boldsymbol{\alpha}_* + \boldsymbol{b}_t. \tag{15}$$

If we knew that $\boldsymbol{\alpha}_*$ had exclusively non-zero coordinates, then by applying Proposition (1) to the dual objective D we can recover a solution $\boldsymbol{\alpha}_*$ and by solving the following linear system:

$$\begin{bmatrix} \boldsymbol{\alpha}_* \\ -c \end{bmatrix} = \texttt{solve}_{\boldsymbol{x} \geq 0} \left(\begin{bmatrix} Q & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{b}_t \\ 1 \end{bmatrix} \right), \tag{16}$$

The first N rows of the system would enforce that $\boldsymbol{\alpha}_*$ satisfies the condition given by Proposition 1, and the last row of the linear system would ensure that the coordinates of $\boldsymbol{\alpha}_*$ sum to one. In the general case, we do not know which coordinates of $\boldsymbol{\alpha}_*$ are non-zero. However, since typical problems are in low-dimension N, we can enumerate all possibilities of subsets of non-zero coordinates for $\boldsymbol{\alpha}_*$.

We detail this further. We consider a non-empty subset $I \subseteq [N]$, for which we define:

$$Q_{[I \times I]} \triangleq (Q_{i,j})_{i \in I, j \in I}\,, \quad \boldsymbol{b}_{[I]} \triangleq (b_{t,i})_{i \in I}\,. \tag{17}$$

We then solve the corresponding linear subsystem:

$$\begin{bmatrix} \boldsymbol{\phi}^{(I)} \\ -c \end{bmatrix} \triangleq \texttt{solve}_{\boldsymbol{x} \geq 0} \left(\begin{bmatrix} Q_{[I \times I]} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{b}_{[I]} \\ 1 \end{bmatrix} \right). \tag{18}$$

This $\boldsymbol{\phi}^{(I)} \in \mathbb{R}^{|I|}$ can then be lifted to $\mathbb{R}^N$ by setting zeros at coordinates non-contained in I. Formally, we define $\boldsymbol{\psi}^{(I)} \in \mathbb{R}^N$ such that:

$$\forall i \in [N],\ \psi_i^{(I)} = \begin{cases} \phi_i^{(I)} \text{ if } i \in I, \\ 0 \text{ otherwise.} \end{cases} \tag{19}$$

Therefore, given $I \subseteq [N]$, we can generate a candidate solution $\boldsymbol{\psi}^{(I)}$ for problem (7) by solving a linear system in dimension $|I|$. In the following proposition, we establish that doing so for all possibilities of I guarantees to find the correct solution:

Proposition 2 (*Problem Equivalence*)

We define the set of feasible solutions reached by the different candidates $\boldsymbol{\psi}^{(I)}$:

$$\Psi = \left\{ \boldsymbol{\psi}^{(I)} : I \subseteq [N], I \neq \varnothing \right\} \cap \Delta_N. \tag{20}$$

Then we have that:

$$\operatorname*{argmax}_{\boldsymbol{\alpha} \in \Delta_N} D(\boldsymbol{\alpha}) = \operatorname*{argmax}_{\boldsymbol{\psi} \in \Psi} D(\boldsymbol{\psi}). \tag{21}$$

In other words, we can find the optimal solution of (7) by enumerating the members of Ψ and picking the one with highest objective value.

This proposition is trivially true because Ψ is simply the intersection between (i) the original feasible set Δ_N and (ii) the set of vectors that satisfy the necessary condition of Optimality given by Proposition 1. This insight results in Algorithm 1 which returns a optimal solution to the dual problem. We characterise this claim in the following proposition.

Proposition 3 (*Sets of Solutions*)

Algorithm 1 returns a solution $\boldsymbol{\alpha}^*$ that satisfies $\boldsymbol{\alpha}^* \in \operatorname{argmax}_{\boldsymbol{\alpha} \in \Delta_N} D(\boldsymbol{\alpha})$. This is true even when a the dual does not have a unique solution. Proof given in Appendix C.

Procedurally Algorithm 1 starts by computing $Q = -\eta A_t^\top A_t$ and $\boldsymbol{b}_t$ to form a "master" system $Q\boldsymbol{x} = \boldsymbol{b}$. We consider each of the $2^N - 1$ subsystems of $Q\boldsymbol{x} = \boldsymbol{b}_t$ defined by an element of the set set I (lines 1-3), where I represents the set of none-zero dimensions of each subsystem. For each subsystem we get a independent subproblem. To ensure the solution to each subproblem will satisfy $\sum_{n=1}^N x^n = 1$ and all partial derivatives have equal value an extra row-column is introduced to each system (line 2). We then compute the point which satisfies the optimality conditions detailed in Proposition 1 for each subsystem, by solving for $\boldsymbol{x}$. We then check if each of these points is feasible, that is, belongs to Δ_N by examining signs $x^n \geq 0, \forall n \in \{1, ..., n\}$ (line 4). Note, we have by construction $\sum_{n=1}^N x^n = 1$. Finally, we select $\boldsymbol{\alpha}^*$ as the feasible point with maximum dual value (lines 7-8). The optimal $\boldsymbol{\alpha}^*$ is then used to define the weight update (8). For example, if $\boldsymbol{\alpha}^* = [1, 0, ..., 0]^\top$ an SGD step $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla \ell_{z_t}(\mathbf{w}_t)$ will be taken.

The BORAT algorithm can be viewed as automatically picking the best step out of a maximum of $(2^N - 1)$ possible options, where each option has a closed form solution. Although the computational complexity of this method is exponential in N, this algorithm is still very efficient in practice for two reasons. First, we only consider small N. Second, as sub problems can be solved independently, it permits a parallel implementation. With a fully parallel implementation the time complexity of this algorithm reduces to $\mathcal{O}\left(N^3\right)$. Empirically with such an implementation, for $N \leq 10$ we observe approximately a linear relationship between N and time taken per training epoch. See Appendix F for a comparison of training epoch time between BORAT and SGD.

Algorithm 1 Dual Optimisation Algorithm

Require: η, N, $\mathcal{P} = \{S : S \subseteq \{1, 2, ..., N\}, S \neq \varnothing\}$, $Q = \eta A_t^\top A_t$, $\boldsymbol{b}_t$, $d_{max} = 0$.

1: **for** $I \in P$ **do**
2: $\quad \hat{Q} = \begin{bmatrix} Q_{[I \times I]} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix}$, $\hat{\boldsymbol{b}} = \begin{bmatrix} \boldsymbol{b}_{[I]} \\ 1 \end{bmatrix}$ ▷ see Equation (17) for definitions
3: $\quad \boldsymbol{\phi}^{[I]} = \texttt{solve}_{\boldsymbol{x}}\left(\hat{Q}\boldsymbol{x} = \hat{\boldsymbol{b}}\right)$ ▷ solve the subsystem, see Equation (18)
4: $\quad$ **if** $\boldsymbol{\phi}_i^{[I]} \geq 0,\ \forall i \in \{1, 2, ..., |I|\}$ **then** ▷ check for non negativity of solution
5: $\quad\quad \boldsymbol{\psi}^{(I)} = \texttt{select}(\boldsymbol{\phi}^{[I]}, I)$ ▷ select elements according to Equation (19)
6: $\quad\quad d = -\frac{1}{2}\boldsymbol{\psi}^{(I)\top} Q \boldsymbol{\psi}^{(I)} + \boldsymbol{\psi}^{(I)\top} \boldsymbol{b}_t$ ▷ compute the dual value
7: $\quad\quad$ **if** $d \geq d_{max}$ **then** ▷ save maximum value
8: $\quad\quad\quad d_{max} = d, \boldsymbol{\alpha}^* = \boldsymbol{\psi}^{(I)}$
9: **Return** $\boldsymbol{\alpha}^*$ ▷ return optimal value

3.7 Computation considerations

While the method of adding linear approximations detailed in (13) requires running Algorithm 1 at each inner loop iteration, when adding an additional element α^n if we keep track of the best dual value we need only compute the 2^{N-1} new subproblems that include non-zero α^n. Thus we only need to run Algorithm 1 once for each $\mathbf{w}_t$ update, that is once per $N - 1$ batches.

3.8 Summary of Algorithm

The full BORAT method is outlined in Algorithm 2. The bundle is constructed in lines 4-6. The update is obtained in lines 7-9 using Algorithm 1. Finally, the updated parameters are projected to the feasible region Ω in line 9.

In the majority of our experiments, we accelerate BORAT with Nesterov momentum. We use Nesterov momentum as we find this helps produce strong generalise performance. The update step at line 9 of Algorithm 2 is then replaced by (i) a velocity update $\boldsymbol{v}_t = \mu \boldsymbol{v}_{t-1} - \eta A_t \boldsymbol{\alpha}$ and (ii) a parameter update $\mathbf{w}_{t+1} = \Pi_\Omega\left(\mathbf{w}_t + \mu \boldsymbol{v}_t\right)$.

Algorithm 2 The BORAT Algorithm

Require: maximal learning-rate η, maximum bundle size $N \geq 2$, initial feasible $\mathbf{w}_0 \in \Omega$.

$t = 0$
while not converged **do**
  Get $\ell_{z_t}(\hat{\mathbf{w}}_t^1)$, $\nabla \ell_{z_t}(\hat{\mathbf{w}}_t^1)$ with z_t drawn i.i.d.
  for $n = 2, ..., N-1$ **do** ▷ sample additional points
    Sample $z_{t,n} \in \mathcal{Z}$, $\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)$, $\nabla \ell_{z_t^n}(\hat{\mathbf{w}}_t^n)$
    compute $\hat{\mathbf{w}}_t^{n+1}$ according to (13)
  compute $-\eta A_t^\top A_t$ and $\boldsymbol{b}_t$
  $\boldsymbol{\alpha}^* = \text{argmax}_{\boldsymbol{\alpha} \in \Delta_N} \left\{ -\frac{\eta}{2} \boldsymbol{\alpha}^\top A_t^\top A_t \boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \boldsymbol{b}_t \right\}$ ▷ see Algorithm 1 for and details
  $\mathbf{w}_{t+1} = \Pi_\Omega \left(\mathbf{w}_t - \eta A_t \boldsymbol{\alpha}^* \right)$ ▷ here Π_Ω is the projection onto Ω
  $t = t + 1$
end while

4. Justification and Analysis

The interpolation setting gives by definition, $f_\star = 0$. However, more subtly, it also allows the updates to rely on the stochastic estimate $\ell_{z_t}(\mathbf{w}_t)$ rather than the exact but expensive $f(\mathbf{w}_t)$. Intuitively, this is possible because in the interpolation setting, we know the global minimum is achieved for each loss function $\ell_{z_t}(\mathbf{w}_t)$ simultaneously. The following results formalise the convergence guarantee of BORAT in the stochastic setting. Note here we prove these result for BORAT with the minor modification, that is, all linear approximations are formed using the same mini-batch of data, $\ell_{z_t^n} = \ell_{z_t}$ for all $n \in \{2, ..., N-1\}$. First, we consider the standard convex setting, where we additionally assume the interpolation assumption is satisfied and that each ℓ_z is Lipschitz continuous. Next we consider an important class of non-convex functions used for analysis in previous works related to interpolation (Vaswani et al., 2019a).

Theorem 1 (*Convex and Lipschitz*)

Let Ω be a convex set. We assume that for every $z \in \mathcal{Z}$, ℓ_z is convex and C-Lipschitz. Let $\mathbf{w}_\star$ be a solution of $(\mathcal{P})$, and assume that we have perfect interpolation: $\forall z \in \mathcal{Z}$, $\ell_z(\mathbf{w}_\star) = 0$. Then BORAT for $N \geq 2$ applied to f satisfies:

$$f\left(\frac{1}{T+1}\sum_{t=0}^{T} \mathbf{w}_t\right) - f_\star \leq C\sqrt{\frac{\|\mathbf{w}_0 - \mathbf{w}_\star\|^2}{(T+1)}} + \frac{\|\mathbf{w}_0 - \mathbf{w}_\star\|^2}{\eta(T+1)}. \tag{22}$$

Hence BORAT recovers the same asymptotic rate as SGD without the need to reduce the learning rate η. In the Appendix D we show that for convex and β-smooth and the α-strongly convex settings BORAT recovers rates of $O(1/T)$ and $O(\exp(\alpha T))$ respectively. Note the earlier version of this work provides convergence results for ALI-G without perfect interpolation.

We follow earlier work (Vaswani et al., 2019a) and provide a convergence rate for BORAT applied to non-convex functions that satisfy the Restricted Secant Inequality (RSI). A function is said to satisfy the RSI condition with constant μ over a the set Ω if the following

holds:

$$\forall \mathbf{w} \in \Omega, \langle \nabla \ell_z(\mathbf{w}), \mathbf{w} - \mathbf{w}_\star \rangle \leq \mu ||\mathbf{w} - \mathbf{w}_\star||. \tag{23}$$

Theorem 2 (*RSI*)

We consider problems of type ($\mathcal{P}$). We assume ℓ_z satisfies the RSI with constant μ, smoothness constant β and perfect interpolation e.g. $l_z(\mathbf{w}^*) = 0$, $\forall z \in \mathcal{Z}$. Then if set $\eta \leq \hat{\eta} = \min\{\frac{1}{4\beta}, \frac{1}{4\mu}, \frac{\mu}{2\beta^2}\}$ then in the worst case we have:

$$f(\mathbf{w}_{T+1}) - f^* \leq exp\left(\left(-\frac{3}{8}\hat{\eta}\mu\right)T\right)||\mathbf{w}_0 - \mathbf{w}^*||^2. \tag{24}$$

In this setting BORAT recovers the same asymptotic rate as SGD.

5. Related work

Bundle Methods. Bundle methods have been primarily proposed for the optimisation of non-smooth convex functions (Lemaréchal et al., 1995, Smola et al., 2007, Auslender, 2009). However, these works do not treat the stochastic case, as they consider small problems where the full gradient can be cheaply evaluated. To our knowledge our work is the first to propose a bundle method for the optimisation of stochastic non-convex problems.

Interpolation in Deep Learning. The interpolation property of DNNs was utilised by early efforts to analyse the convergence speed of SGD. These works demonstrate that SGD achieves the convergence rates of full-batch gradient descent in the interpolation setting (Ma et al., 2018a, Vaswani et al., 2019b, Zhou et al., 2019). Such works are complementary to ours in the sense that they provide a convergence analysis of an existing algorithm for deep learning. In a different line of work, Liu et al. (2019a) propose to exploit interpolation to prove convergence of a new acceleration method for deep learning. However, their experiments suggest that the method still requires the use of a hand-designed learning-rate schedule.

Adaptive Gradient Methods. Similarly to BORAT, most adaptive gradient methods also rely on tuning a single hyperparameter, thereby providing a more pragmatic alternative to SGD that needs a specification of the full learning-rate schedule. While the most popular ones are Adagrad (Duchi et al., 2011), RMSPROP (Tieleman and Hinton, 2012), Adam (Kingma and Welling, 2014) and AMSGrad (Reddi et al., 2018), there have been many other variants (Zeiler, 2012, Orabona and Pál, 2015, Défossez and Bach, 2017, Levy, 2017, Mukkamala and Hein, 2017, Zheng and Kwok, 2017, Bernstein et al., 2018, Chen and Gu, 2018, Shazeer and Stern, 2018, Zaheer et al., 2018, Chen et al., 2019, Loshchilov and Hutter, 2019, Luo et al., 2019). However, as pointed out in (Wilson et al., 2017), adaptive gradient methods tend to give poor generalization in supervised learning. In our experiments, the results provided by BORAT are significantly better than those obtained by the most popular adaptive gradient methods. Recently, Liu et al. (2019b) have proposed to "rectify" Adam with a learning-rate warm up, which partly bridges the gap in generalization performance between Adam and SGD. However, their method still requires a learning-rate schedule, and thus remains difficult to tune on new tasks.

Adaptive Learning-Rate Algorithms. Vaswani et al. (2019b) show that one can use line search in a stochastic setting for interpolating models while guaranteeing convergence. This work is complementary to ours, as it provides convergence results with weaker assumptions on the loss function, but is less practically useful as it requires up to four hyperparameters, instead of one for BORAT. Less closely related methods, included second-order ones, adaptively compute the learning-rate without using the minimum (Schaul et al., 2013, Martens and Grosse, 2015, Tan et al., 2016, Zhang et al., 2017, Baydin et al., 2018, Wu et al., 2018, Li and Orabona, 2019, Henriques et al., 2019), but do not demonstrate competitive generalization performance against SGD with a well-tuned hand-designed schedule.

L_4 **Algorithm.** The L_4 algorithm (Rolinek and Martius, 2018) also uses a modified version of the Polyak step-size. However, the L_4 algorithm computes an online estimate of $f_\star$ rather than relying on a fixed value. This requires three hyperparameters, which are in practice sensitive to noise and crucial for empirical convergence of the method. In addition, L_4 does not come with convergence guarantees. In contrast, by utilizing the interpolation property and a single learning rate, our method is able to (i) provide reliable and accurate minimization with only a single hyperparameter, and (ii) offer guarantees of convergence in the stochastic convex setting.

Frank-Wolfe Methods. The proximal interpretation in Equation (10) allows us to draw additional parallels between ALI-G and existing methods. In particular, the formula of the learning-rate α^1 may remind the reader of the Frank-Wolfe algorithm (Frank and Wolfe, 1956) in some of its variants (Locatello et al., 2017), or other dual methods (Lacoste-Julien and Jaggi, 2013, Shalev-Shwartz and Zhang, 2016). This is because such methods solve in closed form the dual of problem (10), and problems in the form of (10) naturally appear in dual coordinate ascent methods (Shalev-Shwartz and Zhang, 2016).
When no regularization is used, ALI-G and Deep Frank-Wolfe (DFW) (Berrada et al., 2019a) are procedurally identical algorithms. This is because in such a setting, one iteration of DFW also amounts to solving (10) in closed-form – more generally, DFW is designed to train deep neural networks by solving proximal linear support vector machine problems approximately. However, we point out the two fundamental advantages of BORAT over DFW: (i) BORAT can handle arbitrary (lower-bounded) loss functions, while DFW can only use convex piece-wise linear loss functions; and (ii) as seen previously, BORAT provides convergence guarantees in the convex setting.

SGD with Polyak's Learning-Rate. Oberman and Prazeres (2019) extend the Polyak step-size to rely on a stochastic estimation of the gradient $\nabla \ell_{z_t}(\mathbf{w}_t)$ only, instead of the expensive deterministic gradient $\nabla f(\mathbf{w}_t)$. However, they still require evaluating $f(\mathbf{w}_t)$, the objective function over the entire training data set, in order to compute its learning-rate, which makes the method impractical. In addition, since they do not do exploit the interpolation setting nor the fact that regularization can be expressed as a constraint, they also require the knowledge of the optimal objective function value $f_\star$. We also refer the interested reader to the recent analysis of Loizou et al. (2020), which provides a set of improved theoretical results.

aProx **Algorithm.** Most similar to this work (Asi and Duchi, 2019) have recently introduced the aProx algorithm, a family of proximal stochastic optimisation algorithms for

convex problems. The aProx "truncated model" and the aProx "relatively accurate models" share aspects to ALI-G and BORAT respectively. However, there are four clear advantages of this work over (Asi and Duchi, 2019) in the interpolation setting, in particular for training neural networks. First, our work is the first to empirically demonstrate the applicability and usefulness of the algorithm on varied modern non-convex deep learning tasks – most of our experiments use several orders of magnitude more data and model parameters than the small-scale convex problems of (Asi and Duchi, 2019). Second, our analysis and insights allow us to make more aggressive choices of learning rate than (Asi and Duchi, 2019). Indeed, (Asi and Duchi, 2019) assume that the maximal learning-rate is exponentially decaying, even in the interpolating convex setting. In contrast, by avoiding the need for an exponential decay, the learning-rate of BORAT requires only one hyperparameters instead of two for aProx. Third, our analysis proves fast convergence in function space rather than iterate space. Fourth, unlike BORAT Asi and Duchi (2019) do not include the global lower bound in their aProx "relatively accurate models" and does not provide details on how to solve each proximal problem efficiently.

6. Experiments

We split the experimental results into two sections. The first section demonstrates the strong generalisation performance of ALI-G and BORAT on a wide range of tasks. Here we compare ALI-G and BORAT to other single hyperparameter optimisation algorithms. The second section shows for tasks where SGD and ALI-G are sensitive to the learning rate, using larger N increases both the robustness to the learning rate and task regularisation hyperparameter. For experiments we chosen to investigate BORAT with $(N = 3)$ and $(N = 5)$ and refer to the resulting algorithms as BORAT3 and BORAT5 respectively. For $N \geq 2$ BORAT uses more than one batch of data for each update. In order to give a fair comparison we keep the number of passes through the data constant for all experiments. This has the effect that BORAT3 and BORAT5 respectively make a half and a quarter of the weight updates of SGD and ALIG. The time per epoch of BORAT is very similar to that of all other methods, see Appendix F for more details. Hence all methods approximately have the same time cost per epoch. Consequently, faster convergence in terms of number of epochs translates into faster convergence in terms of wall-clock time.
The code to reproduce our results is publicly available[†]. For baselines we use the official implementation where available in PyTorch (Paszke et al., 2017). We use our implementation of L_4, which we unit-test against the official TensorFlow implementation (Abadi et al., 2015). we employ the official implementation of DFW[‡] and we re-use their code for the experiments on SNLI and CIFAR.

6.1 Effectiveness of ALI-G and BORAT

We empirically compare ALI-G and BORAT to the optimisation algorithms most commonly used in deep learning using standard loss functions. Where the hyperparameters of each algorithm are cross validated to select a best performing model. We consider a wide range of

[†] `https://github.com/oval-group/borat`
[‡] `https://github.com/oval-group/dfw`

problems: training a wide residual network on SVHN, training a Bi-LSTM on the Stanford Natural Language Inference data set, training both wide residual networks and densely connected networks on the CIFAR data sets and lastly training a wide residual network on the Tiny Imagenet data set. For these problems, we demonstrate that ALI-G ($N = 2$) and BORAT for $N \in \{3, 5\}$ obtains comparable performance to SGD with a hand-tuned learning rate schedule, and typically outperforms adaptive gradient methods. Finally, we empirically assess the performance of BORAT and its competitors in terms of training objective on CIFAR-100 and ImageNet, in order to demonstrate the scalability of BORAT to large-scale settings. Note that the tasks of training wide residual networks on SVHN and CIFAR-100 are part of the DeepOBS benchmark (Schneider et al., 2019), which aims at standardising baselines for deep learning optimisers. In particular, these tasks are among the most difficult ones of the benchmark because the SGD baseline benefits from a manual schedule for the learning rate where as ALI-G and BORAT uses a single fixed value. Despite this, our set of experiments demonstrate that ALI-G and BORAT obtains competitive performance in relation to SGD. In addition, our methods significantly outperforms adaptive gradient methods. All Experiments were performed on a single GPU (SVHN, SNLI, CIFAR) or on up to 4 GPUs (ImageNet).

6.1.1 Wide Residual Networks on SVHN

Setting. The SVHN data set contains 73k training samples, 26k testing samples and 531k additional easier samples. From the 73k difficult training examples, we select 6k samples for validation; we use all remaining (both difficult and easy) examples for training, for a total of 598k samples. We train a wide residual network 16-4 following (Zagoruyko and Komodakis, 2016).

Method. For SGD, we use the manual schedule for the learning rate of Zagoruyko and Komodakis (2016). For L_4Adam and L_4Mom, we cross-validate the main learning-rate hyperparameter α to be in $\{0.0015, 0.015, 0.15\}$ (0.15 is the value recommended by (Rolinek and Martius, 2018)). For other methods, the learning rate hyperparameter is tuned as a power of 10. The ℓ_2 regularization is cross-validated in $\{0.0001, 0.0005\}$ for all methods but BORAT. For BORAT, the regularization is expressed as a constraint on the ℓ_2-norm of the parameters, and its maximal value is set to 100. SGD, BORAT and BPGrad use a Nesterov momentum of 0.9. All methods use a dropout rate of 0.4 and a fixed budget of 160 epochs, following (Zagoruyko and Komodakis, 2016). A batch size of 128 is used for all experiments.

Results. A comparison to other methods is presented in Table 1. On this relatively easy task, most methods achieve about 98% test accuracy. Despite the cross-validation, L_4Mom does not converge on this task. However, note that L_4Adam achieves accurate results. Even though SGD benefits from a hand-designed schedule, BORAT and other adaptive methods obtain comparable performance on this task.

6.1.2 Bi-LSTM on SNLI

Setting. We train a Bi-LSTM of 47M parameters on the Stanford Natural Language Inference (SNLI) data set (Bowman et al., 2015). The SNLI data set consists of 570k pairs of sentences, with each pair labeled as entailment, neutral or contradiction. This large scale

Test Accuracy on SVHN (%)			
Adagrad	98.0	BPGrad	98.1
AMSGrad	97.9	L_4Adam	**98.2**
DFW	98.1	ALI-G	98.1
L_4Mom	19.6	BORAT3	98.1
Adam	97.9	BORAT5	98.1
SGD	98.3	SGD†	98.4

Table 1: *In red, SGD benefits from a hand-designed schedule for its learning-rate. In black, adaptive methods, including ALI-G, have a single hyperparameter for their learning-rate. SGD† refers to the performance reported by (Zagoruyko and Komodakis, 2016).*

data set is commonly used as a pre-training corpus for transfer learning to many other natural language tasks where labeled data is scarcer (Conneau et al., 2017) – much like ImageNet is used for pre-training in computer vision. We follow the protocol of (Berrada et al., 2019a) and we re-use their results for the baselines.

Method. For L_4Adam and L_4Mom, the main hyperparameter α is cross-validated in $\{0.015, 0.15\}$ – compared to the recommended value of 0.15, this helped convergence and considerably improved performance. The SGD algorithm benefits from a hand-designed schedule, where the learning-rate is decreased by 5 when the validation accuracy does not improve. Other methods use adaptive learning-rates and do not require such a schedule. The value of the main hyperparameter η is cross-validated as a power of ten for the BORAT algorithm and for previously reported adaptive methods. Following the implementation by (Conneau et al., 2017), no ℓ_2 regularization is used. The algorithms are evaluated with the Cross-Entropy (CE) loss and the multi-class hinge loss (SVM), except for DFW which is designed for use with an SVM loss only. For all optimisation algorithms, the model is trained for 20 epochs, using a batch size of 64, following (Conneau et al., 2017).

Results. Table 2 compares ALI-G and BORAT against the other optimises. Moreover, ALI-G, which requires a single hyperparameter for the learning-rate, outperforms all other methods for both the SVM and the CE loss functions. BORAT3 and BORAT5 achieve the same results to ALI-G for both losses.

6.1.3 Wide Residual Networks and Densely Connected Networks on CIFAR

Setting. We follow the methodology of our previous work (Berrada et al., 2019a). We test two architectures: a Wide Residual Network (WRN) 40-4 (Zagoruyko and Komodakis, 2016) and a bottleneck DenseNet (DN) 40-40 (Huang et al., 2017). We use 45k samples for training and 5k for validation. The images are centred and normalized per channel. We apply standard data augmentation with random horizontal flipping and random crops.

Method. We compare ALI-G and BORAT to other common single hyperparameter optimisers. Here we cross validate the hyperparameters in order to find a best performing network for each method. AMSGrad was selected in (Berrada et al., 2019a) because it was the best adaptive method on similar tasks, outperforming in particular Adam and Adagrad.

Test Accuracy on SNLI (%)					
	CE	SVM		CE	SVM
Adagrad*	83.8	84.6	Adam*	84.5	85.0
AMSGrad*	84.2	85.1	BPGrad*	83.6	84.2
DFW*	-	**85.2**	L_4Adam	83.3	82.5
L_4Mom	83.7	83.2	BORAT3	84.4	**85.2**
ALI-G	**84.8**	**85.2**	BORAT5	84.5	**85.2**
SGD*	84.7	85.2	SGD†	84.5	-

Table 2: *In red, SGD benefits from a hand-designed schedule for its learning-rate. In black, adaptive methods have a single hyperparameter for their learning-rate. With an SVM loss, DFW and ALI-G are procedurally identical algorithms – but in contrast to DFW, ALI-G can also employ the CE loss. Methods in the format X* re-use results from (Berrada et al., 2019a). SGD† is the result from (Conneau et al., 2017).*

In addition to these baselines, we also provide the performance of L_4Adam, L_4Mom, AdamW (Loshchilov and Hutter, 2019) and Yogi (Zaheer et al., 2018). We follow the cross validation scheme of (Berrada et al., 2019a) restating it here for completeness, All methods. employ the CE loss only, except for the DFW algorithm, which is designed to use an SVM loss. The batch-size is cross-validated in $\{64, 128, 256\}$ for the DN architecture, and $\{128, 256, 512\}$ for the WRN architecture. For L_4Adam and L_4Mom, the learning-rate hyperparameter α is cross-validated in $\{0.015, 0.15\}$. For AMSGrad, AdamW, Yogi, DFW, ALI-G and BORAT the learning-rate hyperparameter η is cross-validated as a power of 10 (in practice $\eta \in \{0.1, 1\}$ for BORAT). SGD, DFW and BORAT use a Nesterov momentum of 0.9. For all methods excluding ALI-G, BORAT and AdamW, the ℓ_2 regularization is cross-validated in $\{0.0001, 0.0005\}$ on the WRN architecture, and is set to 0.0001 for the DN architecture. For AdamW, the weight-decay is cross-validated as a power of 10. For ALI-G, BORAT, ℓ_2 regularization is expressed as a constraint on the norm of the vector of parameters; and its value is cross validated in $\{50, 75, 100\}$. For all optimisation algorithms, the WRN model is trained for 200 epochs and the DN model for 300 epochs, following (Zagoruyko and Komodakis, 2016) and (Huang et al., 2017) respectively.

Results. Table 3 details the results of the comparison of ALI-G and BORAT against other single hyperparameter optimisers for the CE loss only. In this setting, ALI-G and BORAT outperforms AMSGrad, AdamW, Yogi, L_4Adam and L_4 Mom and constant step size SGD by a large margin. This is true for all values of N. The second best method is DFW which also has the restriction that it can only be used in conjunction with the hinge loss. ALI-G and BORAT produce the test accuracy achievable using SGD with the manual learning rate schedules from (Huang et al., 2017) and (Zagoruyko and Komodakis, 2016) for half of the model data set combinations considered here. For these tasks BORAT provides state of the art results without the need for the learning rate to be manually adapted through out training. For the remaining two combinations; training a DN on Cifar10 and training a WRN on Cifar100, BORAT lags in performance by approximately 0.2% and 2% respectively. With minor variation depending on which version of BORAT is used. Note SGD with a

hand tuned learning rate schedule provides a reasonable upper limit of the generalisation performance achievable due to the amount of time that has been put into improving the schedule.

Test Accuracy on CIFAR (%)				
	CIFAR-10		CIFAR-100	
	WRN	DN	WRN	DN
SGD	91.2	91.5	67.8	67.5
AMSGrad	90.8	91.7	68.7	69.4
AdamW	92.1	92.6	69.6	69.5
Yogi	91.2	92.1	68.7	69.6
DFW	94.2	94.6	76.0	73.2
L_4Adam	90.5	90.8	61.7	60.5
L_4Mom	91.6	91.9	61.4	62.6
ALI-G	**95.4**	94.5	**76.1**	76.2
BORAT3	**95.4**	**94.9**	76.0	**76.5**
BORAT5	95.0	**94.9**	75.8	75.7

Table 3: *Test accuracy of single hyperparameter optimisation methods. Each reported result is an average over three independent runs; the standard deviations and optimal hyperparameters are reported in Appendix F (the standard deviations are at most* 0.4 *for ALI-G and BORAT).*

6.1.4 Wide Residual Networks on Tiny ImageNet

Setting. Tiny ImageNet contains 200 classes for training where each class has 500 images. The validation set contains 10,000 images. All images are RGB with 64x64 pixels. The test-set contains 10,000 images however ground truth labels are not freely available. We again use the Wide Residual Network (WRN) detailed in Section 6.1.1. The images are centred and normalized per channel. We apply standard data augmentation with random horizontal flipping and random crops.

Method. We investigate using SGD (with a constant step-size), ALI-G and BORAT to train a WRN on Tiny ImageNet. We use Adam (Kingma and Ba, 2015) as an indicator of what can be expected from a popular adaptive method applied to the same task. We make use of both the cross entropy (CE) and multi-class hinge (SVM) losses. The learning-rate hyperparameter η is cross validated in powers of 10, a batch-size of 128 and a training time of 200 epochs was used for all experiments. The ℓ_2 regularisation is cross validated in powers of 10 for ADAM. For constant step-size SGD, ALI-G and BORAT we make use of the constrained base regularisation detailed in Section 2, cross validating $r \in \{50, 100, 150, 200, 250\}$. Additionally all methods excluding Adam use a Nesterov momentum of 0.9.

Results. The best results for SGD, ALI-G, BORAT and Adam are shown in table 4. For both losses Adam performs worst, only achieving 2.4% validation accuracy when optimising the SVM loss. The best results are achieved by BORAT with $N = 5$ and $N = 3$ for the

CE and SVM losses respectively. These results suggest the added gain of BORAT is more pronounced for larger more challenging data sets.

Validation Accuracy on Tiny ImageNet (%)		
	CE Loss	Hinge Loss
Adam	55.0	2.4
SGD	59.4	23.2
ALI-G	61.1	24.9
BORAT3	61.1	**44.1**
BORAT5	**62.1**	39.4

Table 4: *Validation accuracy of single hyperparameter optimisation methods on the Tiny ImageNet data set for cross entropy and hinge loss.*

6.1.5 Comparing Training Performance on CIFAR-100

In this section, we empirically assess the performance of ALI-G and its competitors in terms of training objective on CIFAR-100. In order to have comparable objective functions, the ℓ_2 regularization is deactivated. The learning-rate is selected as a power of ten for best final objective value, and the batch-size is set to its default value. For clarity, we only display the performance of SGD, Adam, Adagrad and BORAT (DFW does not support the cross-entropy loss). The L_4 methods diverge in this setting. Here SGD uses a constant learning-rate to demonstrate the need for adaptivity. Therefore all methods use one hyperparameter for their learning-rate. All methods use a fixed budget of 200 epochs for WRN-CIFAR-100 and 300 epochs for DN-CIFAR-100. As can be seen, ALI-G and BORAT provides better training performance than the baseline algorithms on both tasks. Here BORAT3 and BORAT5 are slightly slower than ALI-G due to the fewer number of parameter updates.

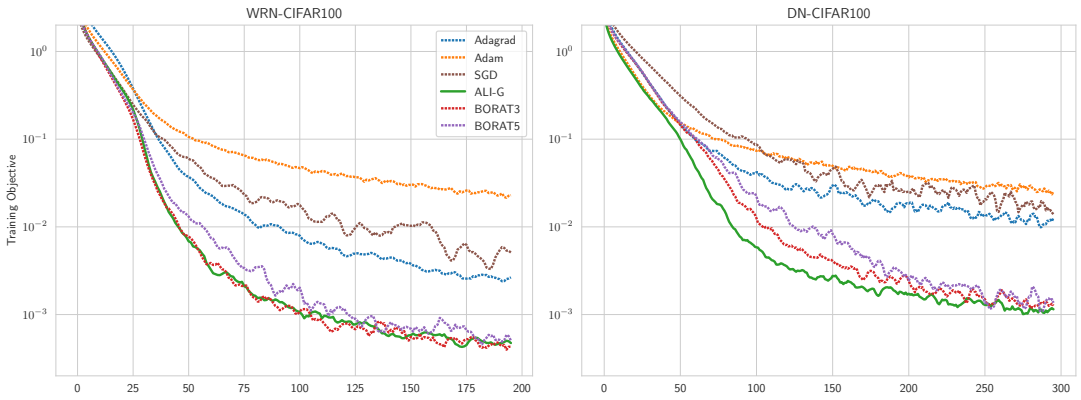

Figure 4: *Objective function over the epochs on CIFAR-100 (smoothed with a moving average over 5 epochs). ALI-G and BORAT reaches a value that is an order of magnitude better than the baselines.*

6.1.6 Training at Large Scale

We demonstrate the scalability of BORAT by training a ResNet18 (He et al., 2016) on the ImageNet data set. In order to satisfy the interpolation assumption, we employ a loss function tailored for top-5 classification (Lapin et al., 2016), and we do not use data augmentation. Our focus here is on the training objective and accuracy. ALI-G and BORAT uses the following training setup: a batch-size of 1024 split over 4 GPUs, a ℓ_2 maximal norm of 400 for $\mathbf{w}$, a maximal learning-rate of 10 and no momentum. As can be seen in figure 5, ALI-G and BORAT reaches 99% top-5 accuracy in 12 epochs, and accurately minimises the objective function to $2 \cdot 10^{-4}$ within 90 epochs.

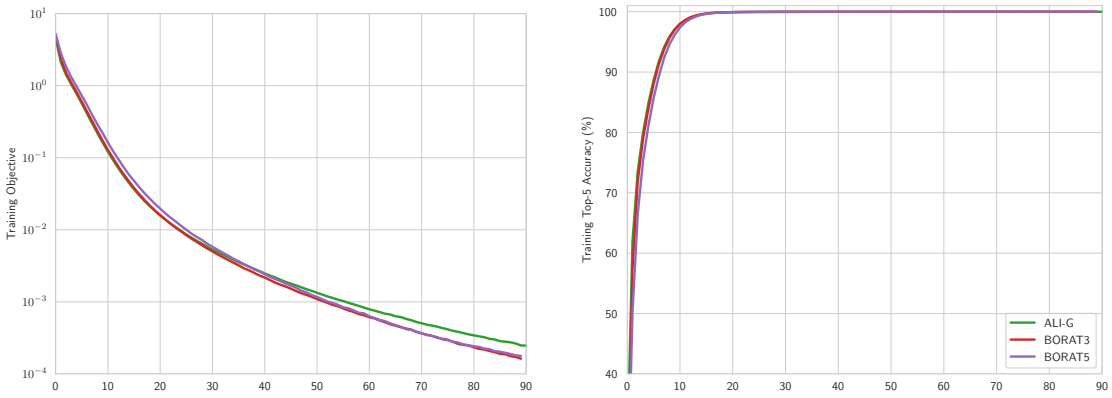

Figure 5: *Training performance of a ResNet-18 learning with different versions of BORAT on ImageNet. Note, that all versions converge at similar rates even though for larger values of N, BORAT3 and BORAT5 makes far fewer updates.*

6.2 Robustness of BORAT

In this section we show that using additional linear approximations increases the robustness of BORAT to its learning rate η and the problem regularisation amount r. BORAT produces high accuracy models for a wider range of values than SGD and ALI-G. Hence on problems where SGD and ALI-G are sensitive to their learning rate BORAT with $N \geq 2$ produces the best performance. These results demonstrate the advantage of using a larger bundle to model the loss in each proximal update.

In order to illustrate this increased robustness we assess the stability of constant step size SGD, ALI-G, BORAT3 and BORAT5 to their hyperparameters. This is done by completing a grid search over η and r for a number of tasks while holding the batch size and epoch budget constant. This allows us to assess the range of values where these algorithms produce high accuracy models. We perform this grid search for six tasks split over the CIFAR100 and Tiny ImageNet data sets. We choose these data sets as they are challenging yet a model capable of interpolation can still fit on a single GPU. We compare against SGD with a constant learning rate for two reasons. First, SGD effectively uses a bundle of size 1, composed of only the linearization around the current iterate. Second, constant step size SGD has a single

learning rate hyperparameter with the same scale as BORAT and also permits the constraint based regularisation described in Section 2.

6.2.1 Wide Residual Networks on CIFAR100 and Tiny ImageNet

Setting. For a description of the CIFAR100 and Tiny Imagenet data sets please refer to sections 6.1.3 and 6.1.4 respectively.

Method. We run four separate experiments on the CIFAR100 data set. The first two of these experiments examine the robustness to hyperparameters of SGD, ALI-G and BORAT when in combination with the cross entropy (CE) and multi-class hinge (SVM) losses. For the second two experiments on CIFAR100 we investigate the same algorithms in the presents of label noise. We limit our self to the CE loss for these two experiments. Label noise is applied by switching the label of the images in the training set with probability p to a random class label in the same super class. We repeat the experiments without label noise on the more challenging Tiny ImageNet data set, resulting in six different settings. For all algorithms we train a wide residual network (WRN) detailed in Section 6.1.3 and make use of a Nesterov momentum of 0.9 as we found its use produced superior generalisation performance. For all experiments we use a batch size of 128 and perform a grid search over the 20 hyper parameter combinations given by $r \in \{50, 100, 150, 200, 250\}$ and $\eta \in \{0.01, 0.1, 1.0, 10.0\}$.

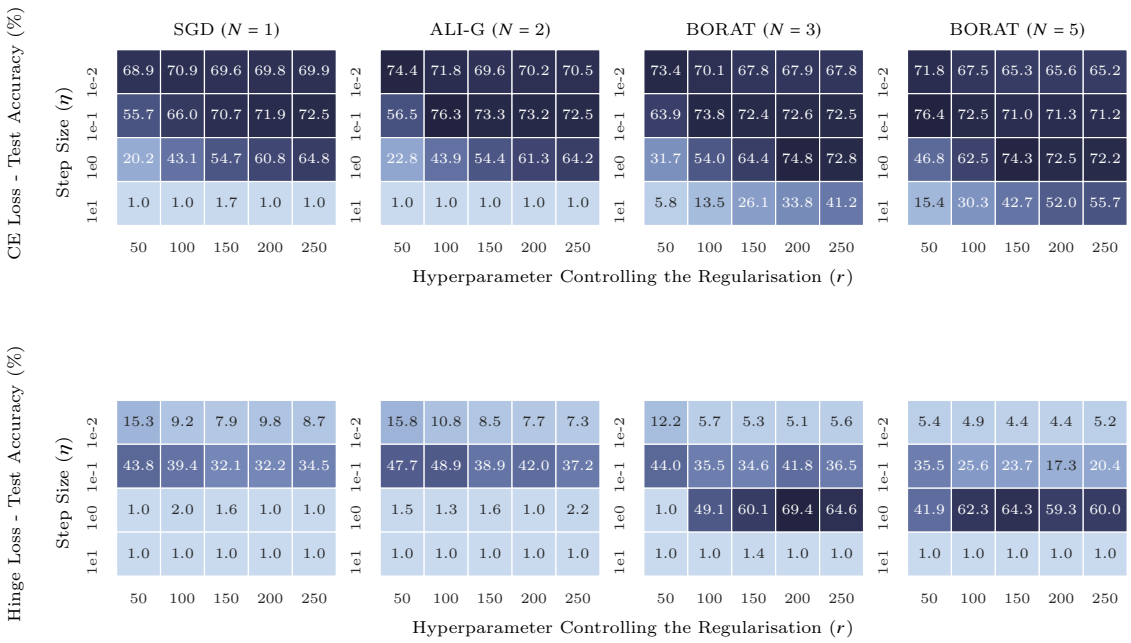

Figure 6: Comparison of SGD and BORAT's robustness to hyperparameters when increasing N for the CE and SVM losses. Experiments are performed on the CIFAR100 data set. Colour represents test performance, where darker colours correspond to higher values. For both losses, increasing N allows for higher learning rates to be used while still producing convergent behaviour. Additionally for the CE loss and $\eta \in \{1.0, 10.0\}$ larger N allows for a smaller r or greater levels of regularisation to be used. Consequently increasing N improves the over all robustness to hyperparameters, while not sacrificing generalisation performance.

Results. Figure 6 details the robustness of SGD, ALI-G and BORAT for the CE and SVM losses without label noise. In these two settings ALI-G exhibits similar robustness to SGD, however it outperforms SGD in terms of the performance of the best model trained. On the CE loss SGD and ALI-G are reasonably robust, providing good results for the majority of hyperparameter combinations with $\eta \leq 1$. For the SVM loss all algorithms are sensitive to their learning rate η. SGD and ALI-G only produce an increase in accuracy for small values of η, hence no model achieves high accuracy in the 200 epoch budget. For both losses, increasing N improves the robustness and produces convergent behaviour for larger values of η and smaller values of r. This is particularly pronounced for the SVM loss. Here permitting larger η allows for a high accuracy network to be trained within the 200 epoch budget. For both losses and $\eta = 0.01$ BORAT3 and BORAT5 slightly under perform compared to ALI-G and SGD. This is simply a consequence of these methods making significantly fewer updates. In spite of this the resultant effect of increasing the bundle size is a larger range of hyperparameters that produce good generalisation performance.
The results from the label noise experiments are shown in Figure 7. For $p = 0.1$ the results closely mirror those where no label noise is used, shown in Figure 6, however here all models achieve roughly $0 - 5\%$ worse test performance. When the noise is increased to $p = 0.5$ the test error increases drastically by $0 - 25\%$. In both cases $p = 0.1$ and $p = 0.5$, increasing N

permits to obtain good results for a larger range of values of η and r. Interestingly, using $N = 3, 5$ results in the best performance when $p = 0.5$. This is somewhat expected since using a larger bundle size means more samples are used to calculate each parameter update and hence helps reduce the effect of the label noise.

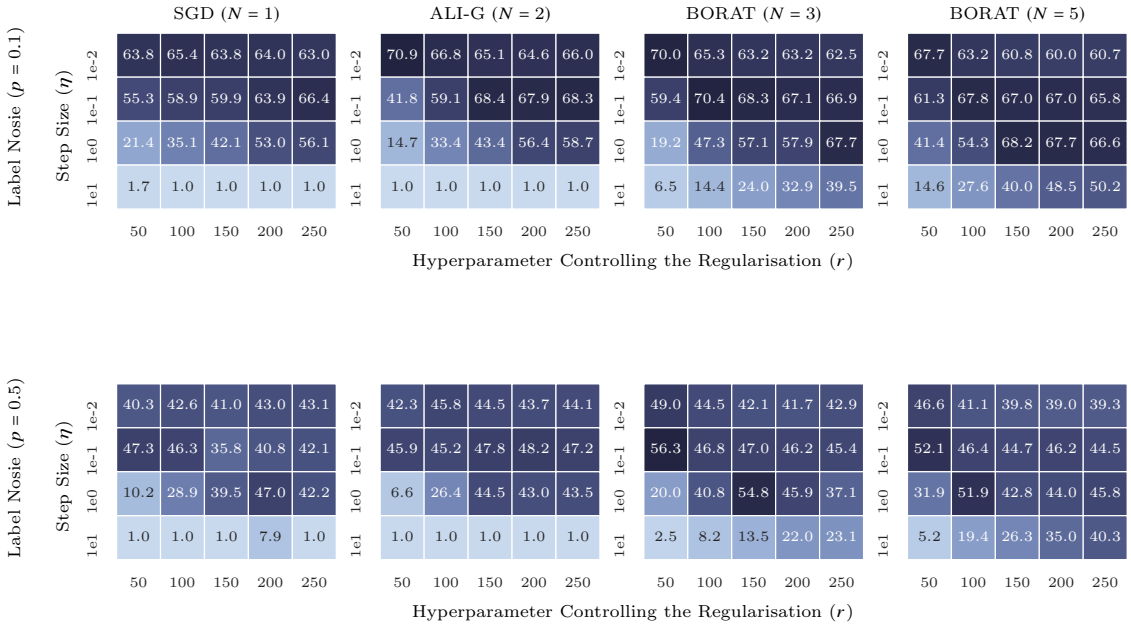

Figure 7: Test accuracy of SGD, ALI-G, BORAT3 and BORAT5 robustness to hyperparameters when trained on CIFAR100 with noisy labels. Here we give results with two different levels of noise $p = 0.1$ (upper row) and $p = 0.5$ (lower row). For readability purposes, the colour encodes test accuracy, where darker colours correspond to higher values. Increasing N allows for higher learning rates and greater levels of regularisation to be used. Additionally, when the level of noise is high ($p = 0.5$, lower row), BORAT3 and BORAT5 significantly outperforms SGD ($N = 1$) and ALI-G ($N = 2$).

Figure 8 details the robustness of SGD, ALI-G and BORAT for the Tiny Imagenet experiments. These results show BORAT offers improved robustness of on multiple data sets as we recover similar performance to CIFAR100. For the CE loss increasing N allows for slightly higher learning rates and greater levels of regularisation to be used for the CE loss. For the SVM loss BORAT produces models with reasonable accuracy with $\eta \in \{0.1, 1.0\}$ opposed to SGD and ALI-G that produce poor results for all learning rates excluding $\eta = 1.0$. Consequently, when training with the SVM loss for 200 epochs, BORAT produces a drastically better models than SGD and ALI-G. This happens despite BORAT making significantly fewer updates of $\mathbf{w}_t$ in the 200 epochs.

Figure 8: Comparison of SGD, ALI-G and BORAT's robustness to hyperparameters on the Tiny ImageNet data set. Here we investigate the affect of increasing the bundle size when in use with the CE and Hinge losses. Colour represents validation performance, where darker colours correspond to higher values. The the CE loss is used BORAT produces an increase in the range of hyperparameters that result in high accuracy models. For the SVM loss BORAT is the only optimiser to produce accurate results

Across all six experiments BORAT consistently produces high accuracy models for a larger range of hyperparameter combinations than SGD or ALI-G. This is particularly pronounced for the SVM loss experiments where the decrease sensitivity of BORAT makes all the difference between finding hyperparameters that result in a high accuracy model and not. Consequently, in comparison to its competitors, BORAT is systematically more robust to the choice of learning-rate and regularization hyper-parameter, and also offers better generalization more often than not. Therefore we particularly recommend using BORAT for tasks where hyper-parameter tuning is a highly time consuming task.

7. Discussion

In this work have introduced BORAT a bundle method designed for the optimisation of DNNs capable of interpolation. We detailed a special case of BORAT with a minimal bundle size that we name ALI-G. ALI-G produces a algorithm that automatically decays the steps size as the loss of the iterate approaches its minimal value. By only needing to find a good maximal learning rate that is automatically decayed we remove the labour and compute intensive task of finding a good learning rate schedule. Using standard publicly available data sets, we have shown that a ALI-G is highly effective in a number of settings. When tuning a

single hyperparameter for the learning-rate, ALI-G achieves state of the art performance while being simple to implement and having minor additional computational cost over SGD. For problems where ALI-G is sensitive to its learning rate its robustness and performance can be enhanced by using BORAT which makes use of a greater bundle size. BORAT produces high accuracy models for a larger range of hyperparameters and is particularly effective when using the multi-class hinge loss or in the presence of label noise. However, BORAT also achieves similar generalisation performance to ALI-G in all settings considered. Our results suggest that BORAT is presently the most robust single hyperparameter optimisation method for DNNs.

It may be possible to modify BORAT so that it can make a parameter update after each gradient evaluation by cleverly selecting how linear approximations are added and removed from the bundle. If done correctly this may lead to a further increase in the speed of convergence. However we leave this to future work. When keeping the total number of gradient evaluations constant, a downside of using a larger bundle size for BORAT is that the number of parameter updates decreases. Despite this, in our experiments, BORAT is able to obtain good results within the same budget of passes through the data. Notably, this also means that BORAT has a time per epoch comparable to that of adaptive gradient methods. Another potential criticism of BORAT is that its memory footprint grows linearly with the bundle size. However in practice, our results show that using a bundle size of three, which corresponds to the same memory cost as ADAM, is often sufficient to obtain improved robustness. Finally, the applicability of BORAT can be limited by the required assumption of interpolation. While we argue that this interpolation property is satisfied in many interesting use cases, it may also not hold true for one or more of the following confounding factors: (i) limited size of the neural network, such as those used in embedded devices; (ii) large size of the training data set, which is becoming increasingly common in machine learning; and (iii) complexity of the loss function, such as in adversarial training. Furthermore, the concept of interpolation itself is ill-defined for unsupervised tasks. Thus, another interesting direction of future work would be to generalise BORAT to non-interpolating settings. This could be achieved by adapting an estimate of the minimum value of the objective function online whilst retaining the desirable quality of minimal hyperparameter tuning.

Appendix A. Dual Derivation

Lemma 1

The dual of the primal problem (6), can be written as follows:

$$\sup_{\boldsymbol{\alpha}\in\Delta_N}\left\{-\frac{\eta}{2}\boldsymbol{\alpha}^\top A_t^\top A_t\boldsymbol{\alpha}+\boldsymbol{\alpha}^\top \boldsymbol{b}_t^n\right\}.$$

Where A_t is defined as a $d\times N$ matrix, whose n^{th} row is $\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)$, $\boldsymbol{b}_t^n=[b_t^1,...,b_t^N]^\top$ and $\boldsymbol{\alpha}=[\alpha^1,\alpha^2,\ldots,\alpha^N]^\top$.

Proof: We start from the primal problem:

$$\operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^d}\left\{\frac{1}{2\eta}||\mathbf{w}-\mathbf{w}_t||^2+\max_{n\in[N]}\left\{\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top(\mathbf{w}-\mathbf{w}_t)+b_t^n\right\}\right\},\tag{25}$$

We define $\tilde{\mathbf{w}}=\mathbf{w}-\mathbf{w}_t$.

$$\min_{\mathbf{w}\in\mathbb{R}^d}\left\{\frac{1}{2\eta}||\tilde{\mathbf{w}}||^2+\max_{n\in[N]}\left\{\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top\tilde{\mathbf{w}}+b_t^n\right\}\right\},$$

$$\min_{\mathbf{w}\in\mathbb{R}^d,\zeta}\left\{\frac{1}{2\eta}||\tilde{\mathbf{w}}||^2+\ \zeta\right\}\ \text{ subject to: } \zeta\geq\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top\tilde{\mathbf{w}}+b_t^n,\ \ \forall n\in[N],$$

$$\min_{\tilde{\mathbf{w}}\in\mathbb{R}^d,\zeta}\sup_{\alpha^n\geq0,\forall n}\left\{\frac{1}{2\eta}||\tilde{\mathbf{w}}||^2+\ \zeta-\sum_{n=1}^N\alpha^n(\zeta-\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top\tilde{\mathbf{w}}-b_t^n)\right\},$$

$$\sup_{\alpha^n\geq0,\forall n}\min_{\tilde{\mathbf{w}}\in\mathbb{R}^d,\zeta}\left\{\frac{1}{2\eta}||\tilde{\mathbf{w}}||^2+\ \zeta-\sum_{n=1}^N\alpha^n(\zeta-\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top\tilde{\mathbf{w}}-b_t^n)\right\},\ \ \text{(strong duality)}$$

$$\sup_{\alpha^n\geq0,\forall n}\min_{\tilde{\mathbf{w}}\in\mathbb{R}^d,\zeta}\left\{\frac{1}{2\eta}||\tilde{\mathbf{w}}||^2+\ \zeta+\sum_{n=1}^N\alpha^n(\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top\tilde{\mathbf{w}}+b_t^n-\zeta)\right\}.$$

The inner problem is now smooth in $\tilde{\mathbf{w}}$ and ζ. We write the KKT conditions:

$$\frac{\partial\cdot}{\partial\tilde{\mathbf{w}}}=\frac{\tilde{\mathbf{w}}}{\eta}+\sum_{n=1}^N\alpha^n\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)=0$$

$$\frac{\partial\cdot}{\partial\zeta}=1-\sum_{n=1}^N\alpha^n=0$$

We define $\Delta_N\triangleq\{\boldsymbol{\alpha}\in\mathbb{R}^N:\sum_{n=1}^N\alpha^n=1,\alpha^n\geq0,n=1,...,N\}$ as probability simplex over the elements of $\boldsymbol{\alpha}$. Thus when we plug in the KKT conditions we obtain:

$$\sup_{\boldsymbol{\alpha}\in\Delta_N}\left\{\frac{1}{2\eta}||\eta\sum_{n=1}^N\alpha^n\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)||^2+\sum_{n=1}^N\alpha^n\left(-\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top(\eta\sum_{m=1}^N\alpha^m\nabla\ell_{z_t^m}(\hat{\mathbf{w}}_t^m))+b_t^n\right)\right\},$$

$$\sup_{\boldsymbol{\alpha}\in\Delta_N}\left\{\frac{\eta}{2}||\sum_{n=1}^N\alpha^n\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)||^2-\eta\sum_{n=1}^N\alpha^n(\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)^\top\sum_{m=1}^N\alpha^m\nabla\ell_{z_t^m}(\hat{\mathbf{w}}_t^m))+\sum_{n=1}^N\alpha^nb_t^n\right\},$$

$$\sup_{\boldsymbol{\alpha}\in\Delta_N}\left\{\frac{\eta}{2}||\sum_{n=1}^N\alpha^n\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)||^2-\eta||\sum_{n=1}^N\alpha^n\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)||^2+\sum_{n=1}^N\alpha^nb_t^n\right\},$$

$$\sup_{\boldsymbol{\alpha}\in\Delta_N}\left\{-\frac{\eta}{2}||\sum_{n=1}^{N}\alpha^n\nabla\ell_{z_t^n}(\hat{\mathbf{w}}_t^n)||^2+\sum_{n=1}^{N}\alpha^n b_t^n\right\}.$$

Using the definitions of A_t, $\boldsymbol{b}_t^n$ and $\boldsymbol{\alpha}$, we can recover the following compact form for the dual problem:

$$\sup_{\boldsymbol{\alpha}\in\Delta_N}\left\{-\frac{\eta}{2}\boldsymbol{\alpha}^\top A_t^\top A_t\boldsymbol{\alpha}+\boldsymbol{\alpha}^\top\boldsymbol{b}_t^n\right\}.$$

■

Appendix B. Proof of Proposition 1

Proposition 1

Let $F:\mathbb{R}^N\to\mathbb{R}$ be a concave function. Let us define $\boldsymbol{\alpha}_*=\operatorname{argmax}_{\boldsymbol{\alpha}\in\Delta}F(\boldsymbol{\alpha})$. Then there exists $c\in\mathbb{R}$ such that:

$$\forall n\in[N]\text{ such that }\alpha_*^n>0,\text{ we have: }\left.\frac{\partial F(\boldsymbol{\alpha})}{\partial\alpha^n}\right|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_*}=c. \tag{26}$$

In other words, the value of the partial derivative is shared among all coordinates of $\boldsymbol{\alpha}_*$ that are non-zero.

Proof: We start form the optimally condition for constrained problems:

$$\langle\nabla F(\boldsymbol{\alpha}_*),\boldsymbol{\alpha}_*-\boldsymbol{\alpha}\rangle\geq 0,\qquad\forall\boldsymbol{\alpha}\in\Delta \tag{27}$$

We consider the point $\hat{\boldsymbol{\alpha}}$. Without loss of generality we assume $\hat{\boldsymbol{\alpha}}$ is equal to $\boldsymbol{\alpha}_*$ for all but two dimensions i and j. We let $\hat{\alpha}_i=0$ and $\hat{\alpha}_j=\alpha_i^*+\alpha_j^*$. Hence we know $\hat{\boldsymbol{\alpha}}\in\Delta$ as we have $\sum_i\hat{\alpha}_i=\sum_i\alpha_i^*=1$ and $\hat{\alpha}_i\geq 0,\ \forall i$. Letting $\boldsymbol{\alpha}=\hat{\boldsymbol{\alpha}}$ in (27) give us:

$$-\frac{\partial F(\boldsymbol{\alpha}_*)}{\partial\alpha_i}\alpha_i^*+\frac{\partial F(\boldsymbol{\alpha}_*)}{\partial\alpha_j}\alpha_i^*\geq 0. \tag{28}$$

Rearranging we have:

$$\frac{\partial F(\boldsymbol{\alpha}_*)}{\partial\alpha_j}\alpha_i^*\geq\frac{\partial F(\boldsymbol{\alpha}_*)}{\partial\alpha_i}\alpha_i^*. \tag{29}$$

Hence for any $\alpha_i^*\neq 0$, we have:

$$\frac{\partial F(\boldsymbol{\alpha}_*)}{\partial\alpha_j}\geq\frac{\partial F(\boldsymbol{\alpha}_*)}{\partial\alpha_i}. \tag{30}$$

Noticing, this results holds for any i and j gives us the following and proves the result.

$$\frac{\partial F(\boldsymbol{\alpha}_*)}{\partial\alpha_j}=\frac{\partial F(\boldsymbol{\alpha}_*)}{\partial\alpha_i}=c \tag{31}$$

■

Appendix C. Proof of Proposition 3

Proposition 3

Algorithm 1 returns a solution $\boldsymbol{\alpha}^*$ that satisfies $\boldsymbol{\alpha}^* \in \operatorname{argmax}_{\boldsymbol{\alpha}\in\Delta_N} D(\boldsymbol{\alpha})$. This is true even when a the dual does not have a unique solution.

Proof: let $\boldsymbol{x}^* \in \operatorname{argmax}_{\boldsymbol{\alpha}\in\Delta_N} D(\boldsymbol{\alpha})$ such that $\boldsymbol{x}^*$ has a maximal number of zero coordinates. x^* exists as we know the solution set is empty. Let I be the set of non-zero coordinates of x^*. We denote by $S^{(I)}$ the set of soultions to the linear system associated with I:

$$\mathcal{S}^{(I)} \triangleq \left\{\boldsymbol{x} \in \mathbb{R}^{|I|} : \tilde{Q}\boldsymbol{x} = \tilde{\boldsymbol{b}}, \boldsymbol{x} \geq 0\right\}, \text{where}, \tilde{Q} \triangleq \begin{bmatrix} Q_{[I\times I]} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} \text{and}, \tilde{\boldsymbol{b}} = \begin{bmatrix} \boldsymbol{b}_{[I]} \\ 1 \end{bmatrix}. \tag{32}$$

$\mathcal{S}^{(I)}$ is a polytope as the intersection intersection between the probability simplex and a linear sub-space. There for it admits and vertex representation:

$$\mathcal{S}^{(I)} = \operatorname{Conv}(\mathcal{V}^{(I)}) \tag{33}$$

such that $\operatorname{Conv}(\cdot)$ denotes the convex hull operation, and $\mathcal{V}^{(I)}$ is a finite set. Since $\mathcal{S}^{(I)}$ contains $\boldsymbol{x}^*[I]$, $\mathcal{S}^{(I)}$ is non-empty and neither is $\mathcal{V}^{(I)}$. let v be and element of $\mathcal{V}^{(I)}$. Note, that if v had one or more zero coordinates, it would be a solution after lifting to $\mathbb{R}^N$ it would also be a optimal solution to the dual as good as $\boldsymbol{x}^*$ while having more zero coordinates than x*, this is impossible by the definition of $\boldsymbol{x}^*$. Thus we can conclude v has exclusively non-zero coordinates.
Since v is a external point of $S^{(I)}$, see section 2.1 of Bertsekas (2009) for a definition. It follows from proposition 2.1.4b of (Bertsekas, 2009) the columns of $\tilde{Q}$ are independent. Therefore the linear system admin a unique solution $\boldsymbol{x}^*$, which is found by Algorithm 1. ■

Appendix D. Convex Results

Lemma 2

Adding additional linear approximations to the bundle of BORAT can never result in a lower maximal dual value. Formally:

$$D^m(\boldsymbol{\alpha}_*) \geq D^l(\boldsymbol{\alpha}_*) \quad \forall m > l. \tag{34}$$

Proof: Any vector of Δ_l can be lifted to Δ_m by appending $m - l$ zeros to it, which does not impact the value of the objective function. The lifted set Δ_l is then a subset of Δ_m, hence the result (maximisation performed performed over a larger space). ■

Theorem 3 (*Convex*)

We assume that Ω is a convex set, and that for every $z \in \mathcal{Z}$, ℓ_z is convex and. Let $\mathbf{w}_\star$ be a solution of ($\mathcal{P}$), and assume that we have perfect interpolation: $\forall z \in \mathcal{Z}$, $\ell_z(\mathbf{w}_\star) = 0$. Then BORAT for $N \geq 2$ applied to f satisfies:

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta \max_{\boldsymbol{\alpha}\in\Delta_N} D(\boldsymbol{\alpha}), \tag{35}$$

where D is defined in (7).

Proof:

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \|\Pi_\Omega(\mathbf{w}_t - \eta A_t \boldsymbol{\alpha}_t) - \mathbf{w}^*\|^2, \tag{36}$$

$$\leq \|\mathbf{w}_t - \eta A_t \boldsymbol{\alpha}_t - \mathbf{w}^*\|^2, \qquad (\Pi_\Omega \text{ projection}) \tag{37}$$

$$\leq \|\mathbf{w}_t - \mathbf{w}^*\|^2 + \eta^2 \|A_t \boldsymbol{\alpha}_t\|^2 - 2\eta \langle A_t \boldsymbol{\alpha}_t, \mathbf{w}_t - \mathbf{w}^* \rangle, \qquad (\text{expanding}) \tag{38}$$

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 - \|\mathbf{w}_t - \mathbf{w}^*\|^2 \leq \eta^2 \|A_t \boldsymbol{\alpha}_t\|^2 - 2\eta \langle A_t \boldsymbol{\alpha}_t, \mathbf{w}_t - \mathbf{w}^* \rangle, \text{ (rearranging)} \tag{39}$$

$$= \eta^2 \|A_t \boldsymbol{\alpha}_t\|^2 - 2\eta \langle A_t \boldsymbol{\alpha}_t, \mathbf{w}_t - \mathbf{w}^* \rangle \tag{40}$$

$$= \eta^2 \|A_t \boldsymbol{\alpha}_t\|^2 - 2\eta \langle A_t \boldsymbol{\alpha}_t, \mathbf{w}_t - \hat{\mathbf{w}}_t^n \rangle - 2\eta \langle A_t \boldsymbol{\alpha}_t, \hat{\mathbf{w}}_t^n - \mathbf{w}^* \rangle, \tag{41}$$

$$= \eta^2 \|A_t \boldsymbol{\alpha}_t\|^2 - 2\eta \langle A_t \boldsymbol{\alpha}_t, \mathbf{w}_t - \hat{\mathbf{w}}_t^n \rangle - 2\eta \sum_{n=1}^{N-1} \alpha^n \nabla \ell_{z_t}(\hat{\mathbf{w}}_t^n)^\top (\hat{\mathbf{w}}_t^n - \mathbf{w}^*), (A_t \alpha_t^N = 0) \tag{42}$$

$$\leq \eta^2 \|A_t \boldsymbol{\alpha}_t\|^2 - 2\eta \langle A_t \boldsymbol{\alpha}_t, \mathbf{w}_t - \hat{\mathbf{w}}_t^n \rangle - 2\eta \sum_{n=1}^{N-1} \alpha_t^n (\ell_{z_t}(\hat{\mathbf{w}}_t^n) - \ell_{z_t}(\mathbf{w}^*)), \qquad (\text{convexity}) \tag{43}$$

$$\leq \eta^2 \|A_t \boldsymbol{\alpha}_t\|^2 - 2\eta \sum_{n=1}^{N-1} \alpha^n \nabla \ell_{z_t}(\hat{\mathbf{w}}_t^n)^\top (\mathbf{w}_t - \hat{\mathbf{w}}_t^n) - 2\eta \sum_{n=1}^{N-1} \alpha_t^n (\ell_{z_t}(\hat{\mathbf{w}}_t^n) - \ell_{z_t}(\mathbf{w}^*)), \tag{44}$$

$$\leq \eta^2 \|A_t \boldsymbol{\alpha}_t\|^2 - 2\eta \sum_{n=1}^{N-1} \alpha^n [\ell_{z_t}(\hat{\mathbf{w}}_t^n) - \nabla \ell_{z_t}(\hat{\mathbf{w}}_t^n)^\top (\hat{\mathbf{w}}_t^n - \mathbf{w}_t)] + 2\eta \sum_{n=1}^{N-1} \alpha_t^n \ell_{z_t}(\mathbf{w}^*), \tag{45}$$

$$\leq \eta^2 \|A_t \boldsymbol{\alpha}_t\|^2 - 2\eta \boldsymbol{\alpha}_t \boldsymbol{b}_t - 2\eta \sum_{n=1}^{N-1} \alpha_t^n \ell_{z_t}(\mathbf{w}^*), \qquad (\boldsymbol{b}_t \text{ definition}) \tag{46}$$

$$\leq -2\eta D(\boldsymbol{\alpha}_t) + 2\eta \sum_{n=1}^{N-1} \alpha_t^n \ell_{z_t}(\mathbf{w}^*), \qquad (D \text{ definition}) \tag{47}$$

$$\leq -2\eta D(\boldsymbol{\alpha}_t) + 2\eta (1 - \alpha_t^N) \ell_{z_t}(\mathbf{w}^*), \qquad (\sum_{n=1}^{N} \alpha_t^n = 1) \tag{48}$$

$$\leq -2\eta D(\boldsymbol{\alpha}_t), \qquad (\ell_{z_t}(\mathbf{w}^*) = 0, \text{ perfect interpolation}) \tag{49}$$

$$\leq -2\eta \max_{\boldsymbol{\alpha} \in \Delta_N} D(\boldsymbol{\alpha}) \qquad (\boldsymbol{\alpha}_t \text{definition}) \tag{50}$$

■

A consequence of Lemma 2 is that the convergence rate given by Theorem 5 improves as N increases.

D.1 Convex and C-Lipshits

Theorem 1 (*Convex and Lipschitz*)

We assume that Ω is a convex set, and that for every $z \in \mathcal{Z}$, ℓ_z is convex and C-Lipschitz. Let $\mathbf{w}_\star$ be a solution of ($\mathcal{P}$), and assume that we have perfect interpolation: $\forall z \in \mathcal{Z}$, $\ell_z(\mathbf{w}_\star) = 0$. Then BORAT for $N \geq 2$ applied to f satisfies:

$$f\left(\tfrac{1}{T+1} \sum_{t=0}^{T} \mathbf{w}_t\right) - f_\star \leq C \sqrt{\frac{\|\mathbf{w}_0 - \mathbf{w}_\star\|^2}{(T+1)}} + \frac{\|\mathbf{w}_0 - \mathbf{w}_\star\|^2}{\eta(T+1)}. \tag{51}$$

Proof: We start from (50), hence we have:

$$\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - 2\eta D(\boldsymbol{\alpha}_t) \tag{52}$$

From Lemma 2 we additionally have that $D^2(\boldsymbol{\alpha}_*) \leq D^{N>2}(\boldsymbol{\alpha}_*)$ hence consider the $N = 2$ as this provides the worse rate.

$$\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - 2\eta D^2(\boldsymbol{\alpha}_t) \tag{53}$$

For $N = 2$ we have exactly two sub problems, and hence can write the dual in following compact form:

$$D^2(\boldsymbol{\alpha}_t) = \begin{cases} -\frac{\eta}{2}\|\boldsymbol{g}_{z_t}\|^2 + \ell_{z_t}(\mathbf{w}_t), & \text{if } \eta\|\boldsymbol{g}_{z_t}\|^2 \leq \ell_{z_t}(\mathbf{w}_t) \\ \frac{1}{2\eta}\frac{\ell_{z_t}(\mathbf{w}_t)^2}{\|\boldsymbol{g}_{z_t}\|^2} & \text{if } \eta\|\boldsymbol{g}_{z_t}\|^2 \geq \ell_{z_t}(\mathbf{w}_t) \end{cases} \tag{54}$$

We now introduce $\mathcal{I}_T$ and $\mathcal{J}_T$ as follows:

$$\mathcal{I}_T \triangleq \left\{t \in \{0, ..., T\} : \eta\|\boldsymbol{g}_{z_t}\|^2 \geq \ell_{z_t}(\mathbf{w}_t)\right\}$$
$$\mathcal{J}_T \triangleq \{0, ..., T\} \setminus \mathcal{I}_T$$

Defining $\mathbb{1}$ to be the indicator function we can write:

$$\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 + \mathbb{1}(t \in \mathcal{I}_T)\eta\left(\eta\|\boldsymbol{g}_{z_t}\|^2 - 2\ell_{z_t}(\mathbf{w}_t)\right) - \mathbb{1}(t \in \mathcal{J}_T)\left(\frac{\ell_{z_t}(\mathbf{w}_t)^2}{\|\boldsymbol{g}_{z_t}\|^2}\right). \tag{55}$$

From our definition of $\mathcal{I}_T$ for all $t \in \mathcal{I}_T$ we have $\eta\|\boldsymbol{g}_{z_t}\|^2 \geq \ell_{z_t}(\mathbf{w}_t)$ hence we can write:

$$\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \mathbb{1}(t \in \mathcal{I}_T)\eta\ell_{z_t}(\mathbf{w}_t) - \mathbb{1}(t \in \mathcal{J}_T)\left(\frac{\ell_{z_t}(\mathbf{w}_t)^2}{\|\boldsymbol{g}_{z_t}\|^2}\right). \tag{56}$$

Summing over T:

$$\|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_0 - \mathbf{w}_\star\|^2 - \eta\sum_{t\in\mathcal{I}_T}\ell_{z_t}(\mathbf{w}_t) - \sum_{t\in\mathcal{J}_T}\left(\frac{\ell_{z_t}(\mathbf{w}_t)^2}{\|\boldsymbol{g}_{z_t}\|^2}\right) \tag{57}$$

Using $\|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2 \geq 0$, we obtain:

$$\eta\sum_{t\in\mathcal{I}_T}\ell_{z_t}(\mathbf{w}_t) + \sum_{t\in\mathcal{J}_T}\left(\frac{\ell_{z_t}(\mathbf{w}_t)^2}{\|\boldsymbol{g}_{z_t}\|^2}\right) \leq \|\mathbf{w}_0 - \mathbf{w}_\star\|^2. \tag{58}$$

From $\left(\frac{\ell_{z_t}(\mathbf{w}_t)^2}{\|\boldsymbol{g}_{z_t}\|^2}\right) \geq 0$, we get:

$$\sum_{t\in\mathcal{I}_T}\ell_{z_t}(\mathbf{w}_t) \leq \frac{1}{\eta}\|\mathbf{w}_0 - \mathbf{w}_\star\|^2. \tag{59}$$

Likewise, using the observation that $\ell_z \geq 0$, we can write:

$$\sum_{t\in\mathcal{J}_T}\frac{\ell_{z_t}(\mathbf{w}_t)^2}{C^2} \leq \sum_{t\in\mathcal{J}_T}\frac{\ell_{z_t}(\mathbf{w}_t)^2}{\|\boldsymbol{g}_{z_t}\|^2} \leq \|\mathbf{w}_0 - \mathbf{w}_\star\|^2. \tag{60}$$

Diving by C^2:

$$\sum_{t\in\mathcal{J}_T} \ell_{z_t}(\mathbf{w}_t)^2 \leq C^2\|\mathbf{w}_0 - \mathbf{w}_\star\|^2. \tag{61}$$

Using the Cauchy-Schwarz inequality, we can further write:

$$\left(\sum_{t\in\mathcal{J}_T} \ell_{z_t}(\mathbf{w}_t)\right)^2 \leq |\mathcal{J}_T| \sum_{t\in\mathcal{J}_T} \ell_{z_t}(\mathbf{w}_t)^2. \tag{62}$$

Therefore we have:

$$\sum_{t\in\mathcal{J}_T} \ell_{z_t}(\mathbf{w}_t) \leq C\sqrt{|\mathcal{J}_T|\|\mathbf{w}_0 - \mathbf{w}_\star\|^2}. \tag{63}$$

We can now put together inequalities (59) and (63) by writing:

$$\sum_{t=0}^{T} \ell_{z_t}(\mathbf{w}_t) = \sum_{t\in\mathcal{J}_T} \ell_{z_t}(\mathbf{w}_t) + \sum_{t\in\mathcal{I}_T} \ell_{z_t}(\mathbf{w}_t) \tag{64}$$

$$\sum_{t=0}^{T} \ell_{z_t}(\mathbf{w}_t) \leq \frac{1}{\eta}\|\mathbf{w}_0 - \mathbf{w}_\star\|^2 + C\sqrt{|\mathcal{J}_T|\|\mathbf{w}_0 - \mathbf{w}_\star\|^2} \tag{65}$$

$$\sum_{t=0}^{T} \ell_{z_t}(\mathbf{w}_t) \leq \frac{1}{\eta}\|\mathbf{w}_0 - \mathbf{w}_\star\|^2 + C\sqrt{(T+1)\|\mathbf{w}_0 - \mathbf{w}_\star\|^2} \tag{66}$$

Dividing by $T+1$ and taking the expectation over z_t, we finally get:

$$f\left(\frac{1}{T+1}\sum_{t=0}^{T}\mathbf{w}_t\right) - f^* \leq \frac{1}{T+1}\sum_{t=0}^{T} f(\mathbf{w}_t) - f^*, \quad (f \text{ is convex}) \tag{67}$$

$$f\left(\frac{1}{T+1}\sum_{t=0}^{T}\mathbf{w}_t\right) - f^* \leq C\sqrt{\frac{\|\mathbf{w}_0 - \mathbf{w}_\star\|^2}{(T+1)}} + \frac{\|\mathbf{w}_0 - \mathbf{w}_\star\|^2}{\eta(T+1)}. \tag{68}$$

■

D.2 Convex and Smooth

Lemma 3

Let $z \in \mathcal{Z}$. Assume that ℓ_z is β-smooth and non-negative on $\mathbb{R}^d$. Then we have:

$$\forall(\mathbf{w}) \in \mathbb{R}^d, \;\; \ell_z(\mathbf{w}) \geq \frac{1}{2\beta}\|\nabla\ell_z(\mathbf{w})\|^2$$

Note that we do not assume that ℓ_z is convex.

Proof: Let $\mathbf{w} \in \mathbb{R}^d$. By Lemma 3.4 of Bubeck (2015), we have:

$$\forall\, \boldsymbol{u} \in \mathbb{R}^d, \;\; |\ell_z(\boldsymbol{u}) - \ell_z(\mathbf{w}) - \nabla\ell_z(\mathbf{w})^\intercal(\boldsymbol{u} - \mathbf{w})| \leq \frac{\beta}{2}\|\boldsymbol{u} - \mathbf{w}\|^2.$$

Therefore we can write:

$$\forall\, \boldsymbol{u} \in \mathbb{R}^d, \;\; \ell_z(\boldsymbol{u}) \leq \ell_z(\mathbf{w}) + \nabla\ell_z(\mathbf{w})^\intercal(\boldsymbol{u} - \mathbf{w})| + \frac{\beta}{2}\|\boldsymbol{u} - \mathbf{w}\|^2.$$

And since $\forall\ \boldsymbol{u}, l_z(\boldsymbol{u}) \geq 0$, we have:

$$\forall\ \boldsymbol{u} \in \mathbb{R}^d,\ \ 0 \leq \ell_z(\mathbf{w}) + \nabla \ell_z(\mathbf{w})^\intercal(\boldsymbol{u} - \mathbf{w})| + \frac{\beta}{2}\|\boldsymbol{u} - \mathbf{w}\|^2.$$

We now choose $\boldsymbol{u} = -\frac{1}{\beta}\nabla l_z(\mathbf{w})$, which yeilds:

$$\forall\ \boldsymbol{u} \in \mathbb{R}^d,\ \ 0 \leq \ell_z(\mathbf{w}) - \frac{1}{\beta}\|\nabla \ell_z(\mathbf{w})\|^2 + \frac{\beta}{2}\|\nabla \ell_z(\mathbf{w})\|^2,$$

which gives the desired result. ■

Theorem 3 (*Convex and Smooth - Large Eta*)

We assume that Ω is a convex set, and that for every $z \in \mathcal{Z}$, ℓ_z is convex and β-smooth. Let $\mathbf{w}_\star$ be a solution of $(\mathcal{P})$, and assume that we have prefect interpolation: $\forall z \in \mathcal{Z}$, $\ell_z(\mathbf{w}_\star) = 0$. Then BORAT for $N \geq 2$ applied to f with $\eta \geq \frac{1}{2\beta}$ satisfies:

$$f\left(\tfrac{1}{T+1}\sum_{t=0}^{T}\mathbf{w}_t\right) - f_\star \leq 2\beta\frac{\|\mathbf{w}_0 - \mathbf{w}_\star\|^2}{T+1}. \tag{69}$$

Proof: We start from Equation (57) we have:

$$\|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_0 - \mathbf{w}_\star\|^2 - \eta\sum_{t\in\mathcal{I}_T}\ell_{z_t}(\mathbf{w}_t) - \sum_{t\in\mathcal{J}_T}\left(\frac{\ell_{z_t}(\mathbf{w}_t)^2}{\|\boldsymbol{g}_{z_t}\|^2}\right). \tag{70}$$

Now using Lemma 3 we can write:

$$\|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_0 - \mathbf{w}_\star\|^2 - \eta\sum_{t\in\mathcal{I}_T}\ell_{z_t}(\mathbf{w}_t) - \frac{1}{2\beta}\sum_{t\in\mathcal{J}_T}\ell_{z_t}(\mathbf{w}_t). \tag{71}$$

From our assumption on $\eta \geq \frac{1}{2\beta}$ we can write:

$$\|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_0 - \mathbf{w}_\star\|^2 - \frac{1}{2\beta}\sum_{t\in\mathcal{I}_T}\ell_{z_t}(\mathbf{w}_t) - \frac{1}{2\beta}\sum_{t\in\mathcal{J}_T}\ell_{z_t}(\mathbf{w}_t), \tag{72}$$

$$\|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_0 - \mathbf{w}_\star\|^2 - \frac{1}{2\beta}\sum_{t=0}^{T}\ell_{z_t}(\mathbf{w}_t). \tag{73}$$

using $\|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2 \geq 0$, we obtain:

$$\sum_{t=0}^{T}\ell_{z_t}(\mathbf{w}_t) \leq 2\beta\|\mathbf{w}_0 - \mathbf{w}_\star\|^2. \tag{74}$$

Dividing by $T+1$ and taking the expectation over z_t, we finally get:

$$f\left(\frac{1}{T+1}\sum_{t=0}^{T}\mathbf{w}_t\right) - f^* \leq \frac{1}{T+1}\sum_{t=0}^{T}f(\mathbf{w}_t) - f^*, \qquad (f \text{ is convex})$$

$$f\left(\frac{1}{T+1}\sum_{t=0}^{T}\mathbf{w}_t\right) - f^* \leq 2\beta\frac{\|\mathbf{w}_0 - \mathbf{w}_\star\|^2}{T+1}.$$

■

Theorem 4 (*Convex and Smooth - Small Eta*)

We assume that Ω is a convex set, and that for every $z \in \mathcal{Z}$, ℓ_z is convex and β-smooth. Let $\mathbf{w}_\star$ be a solution of $(\mathcal{P})$, and assume that we have prefect interpolation: $\forall z \in \mathcal{Z}$, $\ell_z(\mathbf{w}_\star) = 0$. Then BORAT for $N \geq 2$ applied to f with $\eta \leq \frac{1}{2\beta}$ satisfies:

$$f\left(\tfrac{1}{T+1}\sum_{t=0}^{T}\mathbf{w}_t\right) - f_\star \leq 2\beta\frac{\|\mathbf{w}_0 - \mathbf{w}_\star\|^2}{T+1}. \tag{75}$$

Proof: now considering the $\eta \leq \frac{1}{2\beta}$ starting from (76)

$$\|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_0 - \mathbf{w}_\star\|^2 - \eta\sum_{t\in\mathcal{I}_T}\ell_{z_t}(\mathbf{w}_t) - \frac{1}{2\beta}\sum_{t\in\mathcal{J}_T}\ell_{z_t}(\mathbf{w}_t), \tag{76}$$

$$\|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_0 - \mathbf{w}_\star\|^2 - \eta\sum_{t\in\mathcal{I}_T}\ell_{z_t}(\mathbf{w}_t) - \eta\sum_{t\in\mathcal{J}_T}\ell_{z_t}(\mathbf{w}_t), \tag{77}$$

$$\|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_0 - \mathbf{w}_\star\|^2 - \eta\sum_{t=0}^{T}\ell_{z_t}(\mathbf{w}_t). \tag{78}$$

using $\|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2 \geq 0$, we obtain:

$$\sum_{t=0}^{T}\ell_{z_t}(\mathbf{w}_t) \leq \frac{1}{\eta}\|\mathbf{w}_0 - \mathbf{w}_\star\|^2. \tag{79}$$

Dividing by $T+1$ and taking the expectation over z_t, we finally get:

$$f\left(\frac{1}{T+1}\sum_{t=0}^{T}\mathbf{w}_t\right) - f^* \leq \frac{1}{T+1}\sum_{t=0}^{T}f(\mathbf{w}_t) - f^*, \quad (f \text{ is convex})$$

$$f\left(\frac{1}{T+1}\sum_{t=0}^{T}\mathbf{w}_t\right) - f^* \leq \frac{\|\mathbf{w}_0 - \mathbf{w}_\star\|^2}{\eta(T+1)}.$$

■

D.3 Strongly Convex

Finally, we consider the α-strongly convex and β-smooth case.

Lemma 4

Let $z \in \mathcal{Z}$. Assume that ℓ_z is α-strongly convex, non-negative on $\mathbb{R}^d$, and such that $\inf \ell_z = 0$. Then we have:

$$\forall\, \mathbf{w} \in \mathbb{R}^d,\ \frac{\ell_z(\mathbf{w})}{\|\nabla\ell_z(\mathbf{w})\|^2} \leq \frac{1}{2\alpha}. \tag{80}$$

Proof: Let $\mathbf{w} \in \mathbb{R}^d$ and suppose that ℓ_z reaches its minimum at $\underline{\mathbf{w}} \in \mathbb{R}^d$ (this minimum exists because of strong convexity). By definition of strong convexity, we have that:

$$\forall\, \hat{\mathbf{w}} \in \mathbb{R}^d,\ \ell_z(\hat{\mathbf{w}}) \geq \ell_z(\mathbf{w}) + \nabla\ell_z(\mathbf{w})^\top(\hat{\mathbf{w}} - \mathbf{w}) + \frac{\alpha}{2}\|\hat{\mathbf{w}} - \mathbf{w}\|^2 \tag{81}$$

We minimize the right hand-side over $\hat{\mathbf{w}}$, which gives:

$$\begin{aligned}\forall \hat{\mathbf{w}} \in \mathbb{R}^d,\ \ell_z(\hat{\mathbf{w}}) &\geq \ell_z(\mathbf{w}) + \nabla \ell_z(\mathbf{w})^\top (\hat{\mathbf{w}} - \mathbf{w}) + \frac{\alpha}{2}\|\hat{\mathbf{w}} - \mathbf{w}\|^2 \\ &\geq \ell_z(\mathbf{w}) - \frac{1}{2\alpha}\|\nabla \ell_z(\mathbf{w})\|^2\end{aligned} \tag{82}$$

Thus by choosing $\hat{\mathbf{w}} = \underline{\mathbf{w}}$ and re-ordering, we obtain the following result (a.k.a. the Polyak-Lojasiewicz inequality):

$$\ell_z(\mathbf{w}) - \ell_z(\underline{\mathbf{w}}) \leq \frac{1}{2\alpha}\|\nabla \ell_z(\mathbf{w})\|^2 \tag{83}$$

However we have $\ell_z(\underline{\mathbf{w}}) = 0$ which concludes the proof. ■

Lemma 5

For any $a, b \in \mathbb{R}^d$, we have that:

$$\|a\|^2 + \|b\|^2 \geq \frac{1}{2}\|a - b\|^2. \tag{84}$$

Proof: This is a simple application of the parallelogram law, but we give the proof here for completeness.

$$\begin{aligned}\|a\|^2 + \|b\|^2 - \frac{1}{2}\|a - b\|^2 &= \|a\|^2 + \|b\|^2 - \frac{1}{2}\|a\|^2 - \frac{1}{2}\|b\|^2 + a^\top b \\ &= \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2 + a^\top b \\ &= \frac{1}{2}\|a + b\|^2 \\ &\geq 0\end{aligned}$$

■

Lemma 6

Let $z \in \mathcal{Z}$. Assume that ℓ_z is α-strongly convex and achieves its (possibly constrained) minimum at $\mathbf{w}_\star \in \Omega$. Then we have:

$$\forall\, \mathbf{w} \in \Omega,\ \ell_z(\mathbf{w}) - \ell_z(\mathbf{w}_\star) \geq \frac{\alpha}{2}\|\mathbf{w} - \mathbf{w}_\star\|^2 \tag{85}$$

Proof: By definition of strong-convexity Bubeck (2015), we have:

$$\forall\, \mathbf{w} \in \Omega,\ \ell_z(\mathbf{w}) - \ell_z(\mathbf{w}_\star) - \nabla \ell_z(\mathbf{w}_\star)^\top (\mathbf{w} - \mathbf{w}_\star) \geq \frac{\alpha}{2}\|\mathbf{w} - \mathbf{w}_\star\|^2. \tag{86}$$

In addition, since $\mathbf{w}_\star$ minimizes ℓ_z, then necessarily:

$$\forall\, \mathbf{w} \in \Omega,\ \nabla \ell_z(\mathbf{w}_\star)^\top (\mathbf{w} - \mathbf{w}_\star) \geq 0. \tag{87}$$

Combining the two equations gives the desired result. ■

Finally, we consider the α-strongly convex and β-smooth case. Again, our proof yields a natural separation between$\eta \geq \frac{1}{2\beta}$ and $\eta \leq \frac{1}{2\beta}$.

Theorem 5 (*Strongly Convex - Large Eta*)

We assume that Ω is a convex set, and that for every $z \in \mathcal{Z}$, ℓ_z is α-strongly convex and β-smooth. Let $\mathbf{w}_\star$ be a solution of $(\mathcal{P})$, and assume that we have prefect interpolation: $\forall z \in \mathcal{Z}$, $\ell_z(\mathbf{w}_\star) = 0$. Then BORAT for $N \geq 2$ and applied to f with a $\eta \geq \frac{1}{2\beta}$satisfies:

$$\mathbb{E}[f(\mathbf{w}_{t+1})] - f(\mathbf{w}_\star) \leq \frac{\beta}{2} \exp\left(-\frac{\alpha t}{4\beta}\right) \|\mathbf{w}_0 - \mathbf{w}_\star\|^2. \tag{88}$$

Proof: We start from (56):

$$\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \mathbb{1}(t \in \mathcal{I}_T)\eta \ell_{z_t}(\mathbf{w}_t) - \mathbb{1}(t \in \mathcal{J}_T)\left(\frac{\ell_{z_t}(\mathbf{w}_t)^2}{\|\boldsymbol{g}_{z_t}\|^2}\right). \tag{89}$$

We next use Lemma 3 write:

$$\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \mathbb{1}(t \in \mathcal{I}_T)\eta \ell_{z_t}(\mathbf{w}_t) - \mathbb{1}(t \in \mathcal{J}_T)\frac{1}{2\beta} \tag{90}$$

$$ell_{z_t}(\mathbf{w}_t). \tag{91}$$

Now we use our assumption on η to give:

$$\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \frac{1}{2\beta}\ell_{z_t}(\mathbf{w}_t) \tag{92}$$

taking expectations:

$$\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2] \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \frac{1}{2\beta}f(\mathbf{w}_t) \tag{93}$$

Using Lemma 6

$$\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2] \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \frac{\alpha}{4\beta}\|\mathbf{w}_t - \mathbf{w}_\star\|^2 \tag{94}$$

We use a trivial induction over t and write:

$$\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2] \leq \left(1 - \frac{\alpha}{4\beta}\right)\|\mathbf{w}_t - \mathbf{w}_\star\|^2, \tag{95}$$

$$\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2] \leq \left(1 - \frac{\alpha}{4\beta}\right)^t \|\mathbf{w}_0 - \mathbf{w}_\star\|^2, \tag{96}$$

$$\tag{97}$$

Given an arbitrary $\mathbf{w} \in \mathbb{R}^d$, we now wish to relate the distance $\|\mathbf{w} - \mathbf{w}_\star\|^2$ to the function values $f(\mathbf{w}) - f(\mathbf{w}_\star)$. From smoothness, we have:

$$f(\mathbf{w}t + 1) - f(\mathbf{w}_\star) \leq \nabla f(\mathbf{w}_\star)^\top(\mathbf{w}t + 1 - \mathbf{w}_\star) + \frac{\beta}{2}\|\mathbf{w}t + 1 - \mathbf{w}_\star\|^2. \tag{98}$$

However we know by definition $\nabla f(\mathbf{w}_\star) = 0$ hence:

$$f(\mathbf{w}_{t+1}) - f(\mathbf{w}_\star) \leq \frac{\beta}{2}\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2. \tag{99}$$

Taking expectations:

$$\mathbb{E}[f(\mathbf{w}_{t+1})] - f(\mathbf{w}_\star) \leq \frac{\beta}{2}\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2]. \tag{100}$$

Hence we recover the final rate:

$$\mathbb{E}[f(\mathbf{w}_{t+1})] - f(\mathbf{w}_\star) \leq \frac{\beta}{2}\left(1 - \frac{\alpha}{4\beta}\right)^t \|\mathbf{w}_0 - \mathbf{w}_\star\|^2, \tag{101}$$

$$\mathbb{E}[f(\mathbf{w}_{t+1})] - f(\mathbf{w}_\star) \leq \frac{\beta}{2}\exp\left(-\frac{\alpha t}{4\beta}\right)\|\mathbf{w}_0 - \mathbf{w}_\star\|^2. \tag{102}$$

■

Theorem 6 (*Strongly Convex - Small Eta*)

We assume that Ω is a convex set, and that for every $z \in \mathcal{Z}$, ℓ_z is α-strongly convex and β-smooth. Let $\mathbf{w}_\star$ be a solution of $(\mathcal{P})$, and assume that we have prefect interpolation: $\forall z \in \mathcal{Z}$, $\ell_z(\mathbf{w}_\star) = 0$. Then BORAT for $N \geq 2$ and applied to f with a $\eta \leq \frac{1}{2\beta}$satisfies:

$$\mathbb{E}[f(\mathbf{w}_{t+1})] - f(\mathbf{w}_\star) \leq \frac{\beta}{2}\exp\left(-\frac{\alpha\eta t}{2}\right)\|\mathbf{w}_0 - \mathbf{w}_\star\|^2. \tag{103}$$

Proof: We start from (56):

$$\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \mathbb{1}(t \in \mathcal{I}_T)\eta\ell_{z_t}(\mathbf{w}_t) - \mathbb{1}(t \in \mathcal{J}_T)\left(\frac{\ell_{z_t}(\mathbf{w}_t)^2}{\|\boldsymbol{g}_{z_t}\|^2}\right). \tag{104}$$

We next use Lemma 3 write:

$$\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \mathbb{1}(t \in \mathcal{I}_T)\eta\ell_{z_t}(\mathbf{w}_t) - \mathbb{1}(t \in \mathcal{J}_T)\frac{1}{2\beta}\ell_{z_t}(\mathbf{w}_t). \tag{105}$$

Now we use our assumption on η to give:

$$\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2 \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \eta\ell_{z_t}(\mathbf{w}_t) \tag{106}$$

taking expectations:

$$\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2] \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \eta\ell_{z_t}(\mathbf{w}_t) \tag{107}$$

Using Lemma 6

$$\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2] \leq \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \frac{\alpha\eta}{2}\|\mathbf{w}_t - \mathbf{w}_\star\|^2 \tag{108}$$

We use a trivial induction over t and write:

$$\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2] \leq \left(1 - \frac{\alpha\eta}{2}\right)\|\mathbf{w}_t - \mathbf{w}_\star\|^2, \tag{109}$$

$$\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2] \leq \left(1 - \frac{\alpha\eta}{2}\right)^t\|\mathbf{w}_0 - \mathbf{w}_\star\|^2, \tag{110}$$

$$\tag{111}$$

Given an arbitrary $\mathbf{w} \in \mathbb{R}^d$, we now wish to relate the distance $\|\mathbf{w} - \mathbf{w}_\star\|^2$ to the function values $f(\mathbf{w}) - f(\mathbf{w}_\star)$. From smoothness, we have:

$$f(\mathbf{w}t + 1) - f(\mathbf{w}_\star) \leq \nabla f(\mathbf{w}_\star)^\top(\mathbf{w}t + 1 - \mathbf{w}_\star) + \frac{\beta}{2}\|\mathbf{w}t + 1 - \mathbf{w}_\star\|^2. \tag{112}$$

However we know by definition $\nabla f(\mathbf{w}_\star) = 0$ hence:

$$f(\mathbf{w}_{t+1}) - f(\mathbf{w}_\star) \leq \frac{\beta}{2}\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2. \tag{113}$$

Taking expectations:

$$\mathbb{E}[f(\mathbf{w}_{t+1})] - f(\mathbf{w}_\star) \leq \frac{\beta}{2}\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2]. \tag{114}$$

Hence we recover the final rate:

$$\mathbb{E}[f(\mathbf{w}_{t+1})] - f(\mathbf{w}_\star) \leq \frac{\beta}{2}\left(1 - \frac{\alpha\eta}{2}\right)^t \|\mathbf{w}_0 - \mathbf{w}_\star\|^2, \tag{115}$$

$$\mathbb{E}[f(\mathbf{w}_{t+1})] - f(\mathbf{w}_\star) \leq \frac{\beta}{2}\exp\left(\frac{-\alpha\eta t}{2}\right) \|\mathbf{w}_0 - \mathbf{w}_\star\|^2. \tag{116}$$

■

Appendix E. None Convex Results

Here we provide the proof of theorem 2, which we restate for clarity. To simplify our analysis, we consider the BORAT algorithm with $N = 3$. We prove these result for BORAT with the minor modification, that is, all linear approximations are formed using the same mini-batch of data, $\ell_{z_t^n} = \ell_{z_t}$ for all $n \in \{2, ..., N-1\}$.

Theorem 2 (*RSI*)

We consider problems of type (1), see main paper. We assume l_z satisfies RSI with constant μ, smoothness constant β and perfect interpolation e.g. $l_z(\mathbf{w}^*) = 0$, $\forall z \in \mathcal{Z}$. Then if set $\eta \leq \hat{\eta} = \min\{\frac{1}{4\beta}, \frac{1}{4\mu}, \frac{\mu}{2\beta^2}\}$ then in the worst case we have:

$$f(\mathbf{w}_{T+1}) - f^* \leq exp\left(\left(-\frac{3}{8}\hat{\eta}\mu\right)T\right)||\mathbf{w}_0 - \mathbf{w}^*||^2. \tag{117}$$

In order to derive the proof for Theorem 2 we first give a brief overview of BORAT with $N = 3$. We detail the $(2^N - 1)$ possible subproblems (7 in this case), and the resulting values of $\boldsymbol{\alpha}_t$ for each. We show for $\eta \leq \frac{1}{2\beta}$, only a sub-set of the subproblems result in valid solutions with optimal points within the simplex. Finally we derive a rate assuming that for the each of the remaining subproblems is optimal for all t. Lastly by taking the minimum of these rates we construct the worst case rate.

E.0.1 BORAT ($N = 3$)

With ($N = 3$) the bundle is made of three linear approximations. These are, the lower bound on the loss and linear approximations made at the current point and at the optimum of the bundle of size two. Hence this third linear approximation is made at the location one would reach after taking a ALI-G step. Note, here we use γ_t to denote the optimal value of α^1 as given by 12. As some of this proofs get quite involved, we make use of the following abbreviations to save space:

$$\hat{\mathbf{w}}_t^1 = \mathbf{w}_t, \qquad \boldsymbol{g}_{z_t} = \nabla\ell_{z_t^n}(\mathbf{w}_t), \qquad b_t^1 = \ell_{z_t^n}(\mathbf{w}_t),$$

$$\hat{\mathbf{w}}_t^2 = \mathbf{w}'_t, \qquad \boldsymbol{g}'_{z_t} = \nabla \ell_{z_t^n}(\mathbf{w}'_t), \qquad b_t^2 = \ell_{z_t^n}(\mathbf{w}'_t) + \eta\gamma_t \langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t} \rangle.$$

Where $\mathbf{w}'_t$ is defined as $\mathbf{w}'_t = \mathbf{w}_t - \eta\gamma_t \nabla \ell_{z_t^n}(\mathbf{w}_t)$, where $\gamma_t = \min\{1, \frac{\ell_{z_t^n}(\mathbf{w}_t)}{\eta \|\nabla \ell_{z_t^n}(\mathbf{w}_t)\|^2}\eta$. With this notation defined, the BORAT primal problem for this special case can be simplified to:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\mathbb{R}^d}{\operatorname{argmin}} \left\{ \frac{1}{2\eta} ||\mathbf{w} - \mathbf{w}_t||^2 + \max\left\{ \langle \boldsymbol{g}_{z_t}, \mathbf{w} - \mathbf{w}_t \rangle + b_t^1, \langle \boldsymbol{g}'_{z_t}, \mathbf{w} - \mathbf{w}_t \rangle + b_t^2, 0 \right\} \right\}. \quad (118)$$

The dual of (118) be written an:

$$\boldsymbol{\alpha}_t = \underset{\boldsymbol{\alpha}\in\Delta_3}{\operatorname{argmax}} \left\{ -\frac{\eta}{2} ||\alpha^1 \boldsymbol{g}_{z_t} + \alpha^{2t} \boldsymbol{g}'_{z_t}||^2 + \alpha^1 \ell_{z_t^n}(\mathbf{w}_t) + \alpha^2 \ell_{z_t^n}(\mathbf{w}'_t) + \alpha^2 \eta\gamma_t \langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t} \rangle \right\}. \quad (119)$$

For $N = 3$ we have the following parameter update:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha^{1t} \eta \nabla \ell_{z_t^n}(\mathbf{w}_t) - \alpha^{2t} \eta \nabla \ell_{z_t^n}(\mathbf{w}'_t), \quad (120)$$

E.0.2 subproblems

Algorithm 1 solves $(2^N - 1)$ sub problem. Each of these linear systems corresponds to a loci of the simplex, defining the feasible set. We refer to each of these $(2^N - 1)$ options as subproblems. However, each subproblem can also be interpreted as a sub-system of $Q\boldsymbol{\alpha} = \boldsymbol{b}$, (see line 2 of Algorithm 1 for definitions). For $(N = 3)$ the form of $Q\boldsymbol{\alpha} = \boldsymbol{b}$ is detailed in (121).

$$\begin{bmatrix} \eta||\boldsymbol{g}_{z_t}||^2 & \eta\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t} \rangle & 0 & 1 \\ \eta\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t} \rangle & \eta||\boldsymbol{g}'_{z_t}||^2 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha^1 \\ \alpha^2 \\ \alpha^3 \\ c \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ 1 \end{bmatrix}, \quad (121)$$

where c is defined in Theorem 1. Each subproblem is defined by setting a unique subset of the dual variables α^n to zero, before solving for the remaining variables. For $(N = 3)$ we have exactly seven subproblems, which we each give a unique label, see table 5. For clarity, we detail a few specific subproblems. The SGD subproblem corresponds to setting $\alpha_2, \alpha_3 = 0$, and recovers the SGD update. Likewise, the ESGD step corresponds to setting $\alpha^1, \alpha_3 = 0$ and recovers an update similar to the extra gradient method. Finally, by setting $\alpha_2 = 0$ before solving the system we recover the a ALI-G like step, hence we label this subproblem as the ALI-G step. If a subproblem results in a $\boldsymbol{\alpha} \in \Delta_3$ we refer to that subproblem as feasible, conversely, if it results in a $\boldsymbol{\alpha} \notin \Delta_3$ we refer to that subproblem as infeasible. Algorithm 2 computes the dual value for all feasible subproblems and selects the largest. This subproblem's $\boldsymbol{\alpha}$ is then used in (120) to update the parameters. The closed form solutions for $\boldsymbol{\alpha}$ for each of the 7 subproblems are listed in table 5. We use a subscript to show that $\boldsymbol{\alpha}$ belongs to a certain subproblem. For example $\boldsymbol{\alpha}_{SGD} = [1, 0, 0]$.

α^1	α^2	α^3	label
1	0	0	SGD
0	1	0	SEGD
0	0	1	ZERO
0	$\frac{b_2}{\eta\|\|\boldsymbol{g}'_{z_t}\|\|^2}$	$1-\frac{b_2}{\eta\|\|\boldsymbol{g}'_{z_t}\|\|^2}$	EALIG
$\frac{b_1}{\eta\|\|\boldsymbol{g}_{z_t}\|\|^2}$	0	$1-\frac{b_1}{\eta\|\|\boldsymbol{g}_{z_t}\|\|^2}$	ALIG
$\frac{\eta\|\|\boldsymbol{g}'_{z_t}\|\|^2-2\eta\langle\boldsymbol{g}_{z_t},\boldsymbol{g}'_{z_t}\rangle+b_1-b_2}{\eta\|\|\boldsymbol{g}_{z_t}-\boldsymbol{g}'_{z_t}\|\|^2}$	$\frac{\eta\|\|\boldsymbol{g}_{z_t}\|\|^2+b_2-b_1}{\eta\|\|\boldsymbol{g}_{z_t}-\boldsymbol{g}'_{z_t}\|\|^2}$	0	MAX2
$\frac{b_1\|\|\boldsymbol{g}'_{z_t}\|\|^2-b_2\boldsymbol{g}_{z_t}^\top\boldsymbol{g}'_{z_t}}{\eta\|\|\boldsymbol{g}_{z_t}\|\|^2\|\|\boldsymbol{g}'_{z_t}\|\|^2-\eta\|\|\boldsymbol{g}_{z_t}\boldsymbol{g}'_{z_t}\|\|^2}$	$\frac{b_2\|\|\boldsymbol{g}_{z_t}\|\|^2-b_1\boldsymbol{g}_{z_t}^\top a_2}{\eta\|\|\boldsymbol{g}_{z_t}\|\|^2\|\|\boldsymbol{g}'_{z_t}\|\|^2-\eta\|\|\boldsymbol{g}_{z_t}\boldsymbol{g}'_{z_t}\|\|^2}$	$1-\alpha^1-\alpha^2$	MAX3

Table 5: subproblems for $N=3$.

E.0.3 Dual Values

The corresponding expressions for the dual values for the seven different subproblems are detailed below:

$$
\begin{aligned}
D_{ZERO}(\boldsymbol{\alpha}) &= 0,\\
D_{SGD}(\boldsymbol{\alpha}) &= -\frac{\eta}{2}||\boldsymbol{g}_{z_t}||^2 + \ell_{z_t^n}(\mathbf{w}_t),\\
D_{ESGD}(\boldsymbol{\alpha}) &= -\frac{\eta}{2}||\boldsymbol{g}'_{z_t}||^2 + \ell_{z_t^n}(\mathbf{w}'_t) + \eta\gamma_t\langle\boldsymbol{g}_{z_t},\boldsymbol{g}'_{z_t}\rangle,\\
D_{ALIG}(\boldsymbol{\alpha}) &= \frac{1}{2\eta}\frac{\ell_{z_t^n}(\mathbf{w}_t)^2}{||\boldsymbol{g}_{z_t}||^2},\\
D_{EALIG}(\boldsymbol{\alpha}) &= \frac{1}{2\eta}\frac{\left(\ell_{z_t^n}(\mathbf{w}'_t) + \eta\gamma_t\langle\boldsymbol{g}_{z_t},\boldsymbol{g}'_{z_t}\rangle\right)^2}{||\boldsymbol{g}'_{z_t}||^2},\\
D_{MAX2}(\boldsymbol{\alpha}) &= \frac{1}{2\eta||\boldsymbol{g}_{z_t}-\boldsymbol{g}'_{z_t}||^2}\Big((\ell_{z_t^n}(\mathbf{w}'_t)-\ell_{z_t^n}(\mathbf{w}_t))^2 + 2\eta\left(\ell_{z_t^n}(\mathbf{w}'_t)||\boldsymbol{g}_{z_t}||^2 + \ell_{z_t^n}(\mathbf{w}_t)||\boldsymbol{g}'_{z_t}||^2\right)\\
&\qquad - 4\eta\ell_{z_t^n}(\mathbf{w}_t)\langle\boldsymbol{g}_{z_t},\boldsymbol{g}'_{z_t}\rangle + 2\eta^2||\boldsymbol{g}_{z_t}||^2\langle\boldsymbol{g}_{z_t},\boldsymbol{g}'_{z_t}\rangle - \eta^2||\boldsymbol{g}_{z_t}||^2||\boldsymbol{g}'_{z_t}||^2\Big),\\
D_{MAX3}(\boldsymbol{\alpha}) &= \frac{1}{2}\frac{\left(\ell_{z_t^n}(\mathbf{w}'_t)\boldsymbol{g}_{z_t} + \eta\gamma_t\langle\boldsymbol{g}_{z_t},\boldsymbol{g}'_{z_t}\rangle\boldsymbol{g}_{z_t} - \ell_{z_t^n}(\mathbf{w}_t)\boldsymbol{g}'_{z_t}\right)^2}{\eta||\boldsymbol{g}_{z_t}||^2||\boldsymbol{g}'_{z_t}||^2 - \eta\langle\boldsymbol{g}_{z_t},\boldsymbol{g}'_{z_t}\rangle^2}.
\end{aligned}
$$

Due to spatial constraints we state these results without proof. The dual value for each subproblem is simply derived by inserting the closed form expression for $\boldsymbol{\alpha}$ for each subproblem detailed in table 5 into (7). These dual values observe a tree like hierarchy,

$$
\begin{aligned}
D_{MAX3} &\geq D_{MAX2}, D_{ALIG}, D_{EALIG},\\
D_{MAX2} &\geq D_{SGD}, D_{ESGD},\\
D_{ALIG} &\geq D_{SGD}, D_{ZERO},\\
D_{EALIG} &\geq D_{ESGD}, D_{ZERO}.
\end{aligned}
$$

This hierarchy is a consequence of the subproblems maximising the dual over progressively larger spaces, $\Delta^{n-1} \subset \Delta^n$.

E.0.4 Feasible Subproblems

To give a worst case rate on the convergence of BORAT with $N = 3$, we first prove for smooth functions and small η, only certain subproblems will be feasible. We start with a useful lemma, before proving this result.

Lemma 7

Let $z \in \mathcal{Z}$. Assume that ℓ_z is β-smooth. If we define $\mathbf{w}' = \mathbf{w} - \eta\nabla\ell_z(\mathbf{w})$ and $\eta \leq \frac{1}{\beta}$ then we have:

$$\forall(\mathbf{w}) \in \mathbb{R}^d, \;\; \langle \nabla\ell_z(\mathbf{w}), \nabla\ell_z(\mathbf{w}')\rangle \geq 0.$$

Note that we do not assume that ℓ_z is convex.

Proof:

$$\begin{aligned}
2\langle \nabla\ell_z(\mathbf{w}), \nabla\ell_z(\mathbf{w}')\rangle &= -||\nabla\ell_z(\mathbf{w}) - \nabla\ell_z(\mathbf{w}')||^2 + ||\nabla\ell_z(\mathbf{w})||^2 + ||\nabla\ell_z(\mathbf{w}')||^2 \\
2\langle \nabla\ell_z(\mathbf{w}), \nabla\ell_z(\mathbf{w}')\rangle &\geq -\beta^2||\mathbf{w} - \mathbf{w}'||^2 + ||\nabla\ell_z(\mathbf{w})||^2 + ||\nabla\ell_z(\mathbf{w}')||^2, \quad \text{(smoothness)} \\
2\langle \nabla\ell_z(\mathbf{w}), \nabla\ell_z(\mathbf{w}')\rangle &\geq -\beta^2\eta^2||\nabla\ell_z(\mathbf{w})||^2 + ||\nabla\ell_z(\mathbf{w})||^2 + ||\nabla\ell_z(\mathbf{w}')||^2, \quad (\mathbf{w}' \text{ definition}) \\
\langle \nabla\ell_z(\mathbf{w}), \nabla\ell_z(\mathbf{w}')\rangle &\geq \frac{1}{2}(1 - \beta^2\eta^2)||\nabla\ell_z(\mathbf{w})||^2 + \frac{1}{2}||\nabla\ell_z(\mathbf{w}')||^2, \\
\langle \nabla\ell_z(\mathbf{w}), \nabla\ell_z(\mathbf{w}')\rangle &\geq 0. \quad (\frac{1}{\beta} \geq \eta)
\end{aligned}$$

■

Lemma 8 (*Feasible Subproblems*)

If $\ell_{z_t^n}$ has smoothness constant β and we set $\eta \leq \frac{1}{2\beta}$ for the BORAT algorithm with ($N = 3$) detailed in Section E.0.1, the ALIG, EALIG and MAX3 subproblems will always be infeasible.

Proof: We start by showing the ALIG step is infeasible. From Lemma 3 we have:

$$\frac{\ell_z(\mathbf{w})}{||\boldsymbol{g}_{z_t}||^2} \geq \frac{1}{2\beta}.$$

For the ALI-G step to be feasible we require $\alpha^3_{ALIG} > 0$ or $1 - \frac{\ell_{z_t^n}(\mathbf{w}_t)}{\eta||\boldsymbol{g}_{z_t}||^2} > 0$. Rearranging, we have:

$$\eta > \frac{\ell_{z_t^n}(\mathbf{w}_t)}{||\boldsymbol{g}_{z_t}||^2}.$$

Combining these two inequalities gives:

$$\eta > \frac{\ell_z(\mathbf{w})}{||\boldsymbol{g}_{z_t}||^2} \geq \frac{1}{2\beta}.$$

Hence for any $\frac{1}{2\beta} \geq \eta$, $\eta > \frac{\ell_{z_t^n}(\mathbf{w}_t)}{||\boldsymbol{g}_{z_t}||^2}$ cannot hold. We now use a similar argument to show that the EALIG subproblem is infeasible. For EALIG to be feasible we require $\alpha^{3t}_{EALIG} > 0$,

plugging in the closed form solution for α^3_{EALIG} gives:

$$\eta > \frac{\ell_{z_t^n}(\mathbf{w}'_t) + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle}{||\boldsymbol{g}'_{z_t}||^2} = \frac{\ell_{z_t^n}(\mathbf{w}'_t) + \eta\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle}{||\boldsymbol{g}'_{z_t}||^2} = \frac{\ell_{z_t^n}(\mathbf{w}'_t)}{||\boldsymbol{g}'_{z_t}||^2} + \frac{\eta\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle}{||\boldsymbol{g}'_{z_t}||^2} \geq \frac{\ell_{z_t^n}(\mathbf{w}'_t)}{||\boldsymbol{g}'_{z_t}||^2} \geq \frac{1}{2\beta},$$

where the penultimate inequality makes use of $\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle \geq 0$, which is a direct application of Lemma 7. The final inequality is a direct application of Lemma 3. Again, it is clear that the condition $\eta > \frac{\ell_{z_t^n}(\mathbf{w}'_t)}{||\boldsymbol{g}'_{z_t}||^2}$ cannot be satisfied for $\eta \leq \frac{1}{2\beta}$.

We show that the $MAX3$ step is never taken for $\eta \leq \frac{1}{2\beta}$. First, we show that the dual value for the $MAX3$ step can be written as $D_{MAX3}(\boldsymbol{\alpha}) = \frac{1}{2}\left(\ell_{z_t^n}(\mathbf{w}_t)\alpha^{1t} + \ell_{z_t^n}(\mathbf{w}'_t)\alpha^{2t} + \eta\gamma_t\alpha^{2t}\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle\right)$. We start from the dual value stated in Section E.0.3.

$$D_{MAX3}(\boldsymbol{\alpha}) = \frac{1}{2}\frac{\left(\ell_{z_t^n}(\mathbf{w}'_t)\boldsymbol{g}_{z_t} + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle \boldsymbol{g}_{z_t} - \ell_{z_t^n}(\mathbf{w}_t)\boldsymbol{g}'_{z_t}\right)^2}{\eta||\boldsymbol{g}_{z_t}||^2||\boldsymbol{g}'_{z_t}||^2 - \eta\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle^2},$$

expanding,

$$\begin{aligned}
D_{MAX3}(\boldsymbol{\alpha}) = \frac{1}{2}\ell_{z_t^n}(\mathbf{w}'_t) &\underbrace{\frac{\left(\ell_{z_t^n}(\mathbf{w}'_t)\boldsymbol{g}_{z_t} + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle \boldsymbol{g}_{z_t} - \ell_{z_t^n}(\mathbf{w}_t)\boldsymbol{g}'_{z_t}\right)}{\eta||\boldsymbol{g}_{z_t}||^2||\boldsymbol{g}'_{z_t}||^2 - \eta\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle^2}\boldsymbol{g}_{z_t}}_{=\alpha^2} \\
+ \frac{1}{2}\eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle &\underbrace{\frac{\left(\ell_{z_t^n}(\mathbf{w}'_t)\boldsymbol{g}_{z_t} + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle \boldsymbol{g}_{z_t} - \ell_{z_t^n}(\mathbf{w}_t)\boldsymbol{g}'_{z_t}\right)}{\eta||\boldsymbol{g}_{z_t}||^2||\boldsymbol{g}'_{z_t}||^2 - \eta\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle^2}\boldsymbol{g}_{z_t}}_{=\alpha^2} \\
- \frac{1}{2}\ell_{z_t^n}(\mathbf{w}_t) &\underbrace{\frac{\left(\ell_{z_t^n}(\mathbf{w}'_t)\boldsymbol{g}_{z_t} + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle \boldsymbol{g}_{z_t} - \ell_{z_t^n}(\mathbf{w}_t)\boldsymbol{g}'_{z_t}\right)}{\eta||\boldsymbol{g}_{z_t}||^2||\boldsymbol{g}'_{z_t}||^2 - \eta\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle^2}\boldsymbol{g}'_{z_t}}_{=-\alpha^1}.
\end{aligned}$$

Using the definitions of α^1_{MAX3} and α^2_{MAX3} we recover the following expression for the $MAX3$ subproblem's dual function:

$$D_{MAX3}(\boldsymbol{\alpha}) = \frac{1}{2}\left(\ell_{z_t^n}(\mathbf{w}_t)\alpha^{1t} + \ell_{z_t^n}(\mathbf{w}'_t)\alpha^{2t} + \eta\gamma_t\alpha^{2t}\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle\right).$$

With the dual function in this form it is easy to upper bound the feasible dual value for the $MAX3$ subproblem as $D_{MAX3} \leq \frac{1}{2}\max\left\{\ell_{z_t^n}(\mathbf{w}_t), \ell_{z_t^n}(\mathbf{w}'_t) + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle\right\}$. This is a consequence of the fact that $\boldsymbol{\alpha} \in \Delta$ must hold for feasible steps. However, from the hierarchy of dual values we have the lower bounds $D_{MAX3} \geq D_{SGD}$ and $D_{MAX3} \geq D_{ESGD}$, on the $MAX3$ dual value, see Section E.0.3. If either of these lower bounds have larger value than the feasible dual value's upper bound, the $MAX3$ step will not be feasible. We now show that this is always the case for $\eta \leq \frac{1}{2\beta}$. In order to do this we consider two cases.

We first assume $\ell_{z_t^n}(\mathbf{w}_t) \geq \ell_{z_t^n}(\mathbf{w}'_t) + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle$. Hence we know the maximum feasible value for $D^{MAX3} = \frac{1}{2}\ell_{z_t^n}(\mathbf{w}_t)$, if either D^{SGD} or D^{ESGD} have larger dual value we can conclude the $MAX3$ step is unfeasible.

$$D_{SGD}(\boldsymbol{\alpha}) = -\frac{\eta}{2}||\boldsymbol{g}_{z_t}||^2 + \ell_{z_t^n}(\mathbf{w}_t),$$

Hence if the following condition holds we know the $MAX3$ step will be unfeasible:

$$\frac{1}{2}\ell_{z_t^n}(\mathbf{w}_t) \leq -\frac{\eta}{2}||\boldsymbol{g}_{z_t}||^2 + \ell_{z_t^n}(\mathbf{w}_t).$$

Thus, the converse must hold for the $MAX3$ step to be feasible:

$$\frac{1}{2}\ell_{z_t^n}(\mathbf{w}_t) \geq -\frac{\eta}{2}||\boldsymbol{g}_{z_t}||^2 + \ell_{z_t^n}(\mathbf{w}_t),$$

which is equivalent to,

$$\eta \geq \frac{\ell_{z_t^n}(\mathbf{w}_t)}{||\boldsymbol{g}_{z_t}||^2}.$$

Using the same logic as stated for the ALI-G step we know this condition is never satisfied for $\eta \leq \frac{1}{2\beta}$.

We now assume $\ell_{z_t^n}(\mathbf{w}_t) \leq \ell_{z_t^n}(\mathbf{w}'_t) + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle$ and thus we know the max feasible value of $D^{MAX3} \leq \frac{1}{2}\ell_{z_t^n}(\mathbf{w}'_t) + \frac{1}{2}\eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle$, again if either D^{SGD} or D^{ESGD} have larger values, we know the $MAX3$ subproblem is unfeasible:

$$D_{ESGD}(\boldsymbol{\alpha}) = -\frac{\eta}{2}||\boldsymbol{g}'_{z_t}||^2 + \ell_{z_t^n}(\mathbf{w}'_t) + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle.$$

Hence for the $MAX3$ step to be valid we must have:

$$\frac{1}{2}\ell_{z_t^n}(\mathbf{w}'_t) + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle \leq -\frac{\eta}{2}||\boldsymbol{g}'_{z_t}||^2 + \ell_{z_t^n}(\mathbf{w}'_t) + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle,$$

which is equivalent to,

$$\eta \leq \frac{\ell_{z_t^n}(\mathbf{w}'_t) + \eta\gamma_t\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle}{||\boldsymbol{g}'_{z_t}||^2}.$$

Again, we have the same condition as for the EALIG step, which we have already proven can never be feasible for $\eta \leq \frac{1}{2\beta}$. Hence the $MAX3$ subproblem is never feasible for $\eta \leq \frac{1}{2\beta}$. ■

Lemma 9

For any set of vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ then, the following inequality holds:

$$-2||\boldsymbol{a} - \boldsymbol{b}||^2 \leq -||\boldsymbol{a} - \boldsymbol{c}||^2 + 2||\boldsymbol{b} - \boldsymbol{c}||^2.$$

Proof: First consider two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$.

$$\begin{aligned}
0 &\leq ||\boldsymbol{x} - \boldsymbol{y}||^2, \\
0 &\leq ||\boldsymbol{x}||^2 + ||\boldsymbol{y}||^2 - 2\langle \boldsymbol{x}, \boldsymbol{y}\rangle \\
2\langle \boldsymbol{x}, \boldsymbol{y}\rangle &\leq ||\boldsymbol{x}||^2 + ||\boldsymbol{y}||^2, \\
||\boldsymbol{x} + \boldsymbol{y}||^2 &= ||\boldsymbol{x}||^2 + ||\boldsymbol{y}||^2 + 2\langle \boldsymbol{x}, \boldsymbol{y}\rangle, \\
||\boldsymbol{x} + \boldsymbol{y}||^2 &\leq 2||\boldsymbol{x}||^2 + 2||\boldsymbol{y}||^2, \\
-2||\boldsymbol{x}||^2 &\leq 2||\boldsymbol{y}||^2 - ||\boldsymbol{x} + \boldsymbol{y}||^2.
\end{aligned}$$

Setting $\boldsymbol{x} = \boldsymbol{a} - \boldsymbol{b}$ and $\boldsymbol{y} = \boldsymbol{b} - \boldsymbol{c}$ gives the desired result. ■

Lemma 10

Let $z \in \mathcal{Z}$. Assume that ℓ_z is β-smooth and non-negative on $\mathbb{R}^d$. Then we have:

$$\forall(\mathbf{w}) \in \mathbb{R}^d, \;\; \ell_z(\mathbf{w}) \geq \frac{1}{2\beta}||\nabla \ell_z(\mathbf{w})||^2$$

Note that we do not assume that ℓ_z is convex.

Proof: Let $\mathbf{w} \in \mathbb{R}^d$. By Lemma 3.4 of Bubeck (2015), we have:

$$\forall\, \boldsymbol{u} \in \mathbb{R}^d, \;\; |\ell_z(\boldsymbol{u}) - \ell_z(\mathbf{w}) - \nabla \ell_z(\mathbf{w})^\intercal(\boldsymbol{u} - \mathbf{w})| \leq \frac{\beta}{2}||\boldsymbol{u} - \mathbf{w}||^2.$$

Therefore we can write:

$$\forall\, \boldsymbol{u} \in \mathbb{R}^d, \;\; \ell_z(\boldsymbol{u}) \leq \ell_z(\mathbf{w}) + \nabla \ell_z(\mathbf{w})^\intercal(\boldsymbol{u} - \mathbf{w})| + \frac{\beta}{2}||\boldsymbol{u} - \mathbf{w}||^2.$$

And since $\forall\, \boldsymbol{u}, l_z(\boldsymbol{u}) \geq 0$, we have:

$$\forall\, \boldsymbol{u} \in \mathbb{R}^d, \;\; 0 \leq \ell_z(\mathbf{w}) + \nabla \ell_z(\mathbf{w})^\intercal(\boldsymbol{u} - \mathbf{w})| + \frac{\beta}{2}||\boldsymbol{u} - \mathbf{w}||^2.$$

We now choose $\boldsymbol{u} = -\frac{1}{\beta}\nabla l_z(\mathbf{w})$, which yeilds:

$$\forall\, \boldsymbol{u} \in \mathbb{R}^d, \;\; 0 \leq \ell_z(\mathbf{w}) - \frac{1}{\beta}||\nabla \ell_z(\mathbf{w})||^2 + \frac{\beta}{2}||\nabla \ell_z(\mathbf{w})||^2,$$

which gives the desired result. ■

In this section we derive the rate for each of the remaining feasible steps, that is, *SGD*, *ESGD* and $MAX2$.

E.1 SGD Subproblem

Lemma 11

We assume that $\Omega = \mathbb{R}^d$, for every $z \in \mathcal{Z}$, $\ell_z(w)$ is β and satisfies the RSI condition with constant μ. Let $\mathbf{w}^*$ be a solution of $f(\mathbf{w})$. We assume $\forall z \in \mathcal{Z}, \ell_{z_t^n}(\mathbf{w}^*) = 0$. Then, if we apply BORAT with $\eta \leq \hat{\eta} = \min\{\frac{1}{4\beta}, \frac{1}{4\mu}, \frac{\mu}{\beta^2}\}$ and we take the step resulting from the SGD subproblem for all t we have:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] \leq (1 - \hat{\eta}\mu)||\mathbf{w}_t - \mathbf{w}^*||^2.$$

Proof:

$$\begin{aligned}
||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 &\leq ||\Pi_\Omega(\mathbf{w}_t - \eta \boldsymbol{g}'_{z_t}) - \mathbf{w}^*||^2, && (122)\\
&\leq ||\mathbf{w}_t - \eta \boldsymbol{g}_{z_t} - \mathbf{w}^*||^2, && (123)\\
&= ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\boldsymbol{g}_{z_t}||^2 - 2\eta\langle \boldsymbol{g}_{z_t}, \mathbf{w}_t - \mathbf{w}^*\rangle, && (124)\\
&\leq ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\boldsymbol{g}_{z_t}||^2 - 2\eta\mu||\mathbf{w}_t - \mathbf{w}^*||^2. && (125)
\end{aligned}$$

We have $||\boldsymbol{g}_{z_t}||^2 \leq 2\beta\ell_{z_t^n}(\mathbf{w}_t)$ from Lemma 3 and $\ell_{z_t^n}(\mathbf{w}_t) \leq \frac{\beta}{2}||\mathbf{w}_t - \mathbf{w}^*||^2$ from smoothness giving $||\boldsymbol{g}_{z_t}||^2 \leq \beta^2||\mathbf{w}_t - \mathbf{w}^*||^2$. We can now upper bound the r.h.s producing:

$$||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \leq ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2\beta^2||\mathbf{w}_t - \mathbf{w}^*||^2 - 2\eta\mu||\mathbf{w}_t - \mathbf{w}^*||^2, \tag{126}$$

$$||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \leq (1 - 2\eta\mu + \eta^2\beta^2)||\mathbf{w}_t - \mathbf{w}^*||^2, \tag{127}$$

$$||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \leq \left(1 - \eta(2\mu - \eta\beta^2)\right)||\mathbf{w}_t - \mathbf{w}^*||^2. \tag{128}$$

Now, if we select $\eta \leq \hat{\eta} = \min\{\frac{1}{2\beta}, \frac{1}{4\mu}, \frac{\mu}{\beta^2}\}$ in the worst case we get:

$$||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \leq (1 - \hat{\eta}\mu)||\mathbf{w}_t - \mathbf{w}^*||^2. \tag{129}$$

Taking expectations with respect to z_t:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] \leq \mathbb{E}[(1 - \hat{\eta}\mu)||\mathbf{w}_t - \mathbf{w}^*||^2]. \tag{130}$$

Noting that $\mathbf{w}_t$ does not depend on z_t, and neither does $\mathbf{w}^*$ due to the interpolation property:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] \leq (1 - \hat{\eta}\mu)||\mathbf{w}_t - \mathbf{w}^*||^2. \tag{131}$$

■

E.2 ESGD Subproblem

Lemma 12

Let $z \in \mathcal{Z}$. We assume that ℓ_z is β-smooth and non-negative on $\mathbb{R}^d$. Additionally we assume that $\eta \leq \frac{1}{2\beta}$. If we define $\gamma_t \doteq \min\{1, \frac{\ell_{z_t^n}(\mathbf{w}_t)}{\eta||\boldsymbol{g}_{zt}||^2}\}$, then we have:

$$\gamma_t = 1, \ \forall t$$

Proof: From our assumption on η and Lemma 3 we have:

$$\eta \leq \frac{1}{2\beta} \leq \frac{\ell_{z_t^n}(\mathbf{w}_t)}{||\boldsymbol{g}_{zt}||^2}.$$

Rearranging gives:

$$1 \leq \frac{\ell_{z_t^n}(\mathbf{w}_t)}{\eta||\boldsymbol{g}_{zt}||^2}.$$

Plugging in to the the definition of γ_t, gives the desired result. ■

Lemma 13

We assume that $\Omega = \mathbb{R}^d$, for every $z \in \mathcal{Z}$, $\ell_z(w)$ is β and satisfies the RSI condition with constant μ. Let $\mathbf{w}^*$ be a solution of $f(\mathbf{w})$. We assume $\forall z \in \mathcal{Z}, \ell_{z_t^n}(\mathbf{w}^*) = 0$. Then, if we apply BORAT with $\eta \leq \hat{\eta} = \min\{\frac{1}{4\beta}, \frac{1}{4\mu}, \frac{\mu}{\beta^2}\}$ and we take the step resulting from the ESGD subproblem for all t we have:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] \leq (1 - \hat{\eta}\mu)||\mathbf{w}_t - \mathbf{w}^*||^2.$$

Proof: *This proof loosely follows work by Vaswani et al. (2019a).*

$$||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \leq ||\Pi_\Omega(\mathbf{w}_t - \eta \boldsymbol{g}'_{z_t}) - \mathbf{w}^*||^2, \tag{132}$$
$$\leq ||\mathbf{w}_t - \eta \boldsymbol{g}'_{z_t} - \mathbf{w}^*||^2, \tag{133}$$
$$= ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 ||\boldsymbol{g}'_{z_t}||^2 - 2\eta \langle \boldsymbol{g}'_{z_t}, \mathbf{w}_t - \mathbf{w}^* \rangle, \tag{134}$$
$$= ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 ||\boldsymbol{g}'_{z_t}||^2 - 2\eta \langle \boldsymbol{g}'_{z_t}, \mathbf{w}'_t + \eta\gamma_t \boldsymbol{g}_{z_t} - \mathbf{w}^* \rangle, \tag{135}$$
$$= ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 ||\boldsymbol{g}'_{z_t}||^2 - 2\eta \langle \boldsymbol{g}'_{z_t}, \mathbf{w}'_t + \eta \boldsymbol{g}_{z_t} - \mathbf{w}^* \rangle, \text{ (lemma 12)} \tag{136}$$
$$= ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 ||\boldsymbol{g}'_{z_t}||^2 - 2\eta \langle \boldsymbol{g}'_{z_t}, \mathbf{w}'_t - \mathbf{w}^* \rangle - 2\eta^2 \langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t} \rangle. \tag{137}$$

Using the RSI condition:

$$||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \leq ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 ||\boldsymbol{g}'_{z_t}||^2 - 2\eta\mu ||\mathbf{w}'_t - \mathbf{w}^*||^2 - 2\eta^2 \langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t} \rangle. \tag{138}$$

Using lemma (9) to upper bound $-||\mathbf{w}'_t - \mathbf{w}^*||^2$,

$$= ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 ||\boldsymbol{g}'_{z_t}||^2 - \eta\mu ||\mathbf{w}^* - \mathbf{w}_t||^2 + 2\eta\mu ||\mathbf{w}_t - \mathbf{w}'_t||^2 - 2\eta^2 \langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t} \rangle, \tag{139}$$
$$= (1 - \eta\mu) ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 ||\boldsymbol{g}'_{z_t}||^2 + 2\eta\mu ||\mathbf{w}_t - \mathbf{w}'_t||^2 - 2\eta^2 \langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t} \rangle, \tag{140}$$
$$= (1 - \eta\mu) ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 ||\boldsymbol{g}'_{z_t} - \boldsymbol{g}_{z_t}||^2 - \eta^2 ||\boldsymbol{g}_{z_t}||^2 + 2\eta\mu ||\mathbf{w}_t - \mathbf{w}'_t||^2, \tag{141}$$
$$= (1 - \eta\mu) ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 ||\boldsymbol{g}'_{z_t} - \boldsymbol{g}_{z_t}||^2 - \eta^2 ||\boldsymbol{g}_{z_t}||^2 + 2\eta^3\mu ||\boldsymbol{g}_{z_t}||^2, \tag{142}$$
$$\leq (1 - \eta\mu) ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2\beta^2 ||\mathbf{w}'_t - \mathbf{w}_t||^2 - \eta^2 ||\boldsymbol{g}_{z_t}||^2 + 2\eta^3\mu ||\boldsymbol{g}_{z_t}||^2, \text{(smoothness)} \tag{143}$$
$$= (1 - \eta\mu) ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^4\beta^2 ||\boldsymbol{g}_{z_t}||^2 - \eta^2 ||\boldsymbol{g}_{z_t}||^2 + 2\eta^3\mu ||\boldsymbol{g}_{z_t}||^2, \tag{144}$$
$$= (1 - \eta\mu) ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 \left(\eta^2\beta^2 - 1 + 2\eta\mu\right) ||\boldsymbol{g}_{z_t}||^2, \tag{145}$$

Taking expectations with respect to z_t:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] = \mathbb{E}[(1 - \eta\mu) ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 \left(\eta^2\beta^2 - 1 + 2\eta\mu\right) ||\boldsymbol{g}_{z_t}||^2, \tag{146}$$

Noting that $\mathbf{w}_t$ does not depend on z_t, and neither does $\mathbf{w}^*$ due to the interpolation property:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] = (1 - \eta\mu) ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2 \left(\eta^2\beta^2 - 1 + 2\eta\mu\right) \mathbb{E}[||\boldsymbol{g}_{z_t}||^2], \tag{147}$$

If we set $\eta \leq \hat{\eta} = \min\{\frac{1}{2\beta}, \frac{1}{4\mu}, \frac{\mu}{\beta^2}\}$ then we have $\eta^2\beta^2 \leq \frac{1}{4}$, $2\eta\mu \leq \frac{1}{2}$,hence $\left(\eta^2\beta^2 - 1 + 2\eta\mu\right) \leq 0$.

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] \leq (1 - \eta\mu) ||\mathbf{w}_t - \mathbf{w}^*||^2. \tag{148}$$

Hence if we insert the chosen value for η then we have:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] \leq (1 - \hat{\eta}\mu) ||\mathbf{w}_t - \mathbf{w}^*||^2. \tag{149}$$

■

E.3 MAX2 Subproblem

Lemma 14

We assume that $\Omega = \mathbb{R}^d$, for every $z \in \mathcal{Z}$, $l_z(w)$ is β and satisfies the RSI condition with constant μ. Let $\mathbf{w}^*$ be a solution of $f(\mathbf{w})$. We assume $\forall z \in \mathcal{Z}, l_{z_t}(\mathbf{w}^*) = 0$. Then, if we apply BORAT with $\eta \leq \hat{\eta} = \min\{\frac{1}{4\beta}, \frac{1}{4\mu}, \frac{\mu}{\beta^2}\}$ and we take the step resulting from the

MAX2 subproblem for all t we have:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] \leq \left(1 - \frac{3}{8}\hat{\eta}\mu\right)^t ||\mathbf{w}_0 - \mathbf{w}^*||^2.$$

Proof: Note we assume $\gamma_t = 1$ as proved in Lemma 12.

$$\begin{aligned}
||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 &\leq ||\Pi_\Omega(\mathbf{w}_t - \eta\alpha^{1t}\boldsymbol{g}_{z_t} - \eta\alpha^{2t}\boldsymbol{g}'_{z_t}) - \mathbf{w}^*||^2, && (150)\\
&\leq ||\mathbf{w}_t - \eta\alpha^{1t}\boldsymbol{g}_{z_t} - \eta\alpha^{2t}\boldsymbol{g}'_{z_t} - \mathbf{w}^*||^2, && (151)\\
&= ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 \\
&\quad - 2\eta\langle\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}, \mathbf{w}_t - \mathbf{w}^*\rangle, && (152)\\
&= ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 \\
&\quad - 2\eta\alpha^{1t}\langle\boldsymbol{g}_{z_t}, \mathbf{w}_t - \mathbf{w}^*\rangle - 2\eta\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \mathbf{w}_t - \mathbf{w}^*\rangle, && (153)\\
&= ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 - 2\eta\alpha^{1t}\langle\boldsymbol{g}_{z_t}, \mathbf{w}_t - \mathbf{w}^*\rangle \\
&\quad - 2\eta\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \mathbf{w}'_t + \eta\boldsymbol{g}_{z_t} - \mathbf{w}^*\rangle, && (154)\\
&= ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 - 2\eta\alpha^{1t}\langle\boldsymbol{g}_{z_t}, \mathbf{w}_t - \mathbf{w}^*\rangle \\
&\quad - 2\eta\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \mathbf{w}'_t - \mathbf{w}^*\rangle - 2\eta^2\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle, && (155)\\
&= ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 - 2\eta^2\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle \\
&\quad - 2\eta\alpha^{1t}\langle\boldsymbol{g}_{z_t}, \mathbf{w}_t - \mathbf{w}^*\rangle - 2\eta\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \mathbf{w}'_t - \mathbf{w}^*\rangle. && (156)
\end{aligned}$$

We now make use of $-\langle\boldsymbol{g}_{z_t}, \mathbf{w}_t - \mathbf{w}^*\rangle \leq -\mu||\mathbf{w}^* - \mathbf{w}_t||^2$ (RSI condition),

$$\begin{aligned}
||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 &\leq ||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 \\
&\quad - 2\eta^2\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle - 2\eta\alpha^{1t}\mu||\mathbf{w}_t - \mathbf{w}^*||^2 - 2\eta\langle\alpha^{2t}\boldsymbol{g}'_{z_t}, \mathbf{w}'_t - \mathbf{w}^*\rangle, && (157)\\
||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 &\leq (1 - 2\eta\alpha^{1t}\mu)||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 \\
&\quad - 2\eta^2\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle - 2\eta\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \mathbf{w}'_t - \mathbf{w}^*\rangle. && (158)
\end{aligned}$$

Similarly using $-\langle\boldsymbol{g}'_{z_t}, \mathbf{w}'_t - \mathbf{w}^*\rangle \leq -\mu||\mathbf{w}^* - \mathbf{w}'_t||^2$ (RSI condition),

$$\begin{aligned}
||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 &\leq (1 - 2\eta\alpha^{1t}\mu)||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 \\
&\quad - 2\eta^2\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle - 2\eta\alpha^{2t}\mu||\mathbf{w}'_t - \mathbf{w}^*||^2. && (159)
\end{aligned}$$

We now upper bound $-||\mathbf{w}'_t - \mathbf{w}^*||^2$, using lemma (9):

$$\begin{aligned}
||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 &\leq (1 - 2\eta\alpha^{1t}\mu)||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 \\
&\quad - 2\eta^2\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle - \alpha^{2t}\eta\mu||\mathbf{w}_t - \mathbf{w}^*||^2 + 2\alpha^{2t}\eta\mu||\mathbf{w}_t - \mathbf{w}'_t||^2. && (160)
\end{aligned}$$

This gives the following general form:

$$\begin{aligned}
||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 &\leq (1 - 2\eta\alpha^{1t}\mu - \alpha^{2t}\eta\mu)||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 \\
&\quad - 2\eta^2\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle + 2\alpha^{2t}\eta\mu||\mathbf{w}_t - \mathbf{w}'_t||^2. && (161)
\end{aligned}$$

We now use the inequality $||\mathbf{w}_t - \mathbf{w}'_t|| = \eta^2||\boldsymbol{g}_{z_t}|| \leq \eta^2\beta^2||\mathbf{w}_t - \mathbf{w}^*||$ to upper bound the final term, see SGD proof for derivation of inequality:

$$\begin{aligned}
||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 &\leq (1 - 2\eta\alpha^{1t}\mu - \alpha^{2t}\eta\mu)||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 \\
&\quad - 2\eta^2\alpha^{2t}\langle\boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle + 2\eta^3\beta^2\mu\alpha^{2t}||\mathbf{w}_t - \mathbf{w}^*||, && (162)
\end{aligned}$$

$$||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \leq (1 - 2\eta\alpha^{1t}\mu - \alpha^{2t}\eta\mu + 2\eta^3\alpha^{2t}\beta^2\mu)||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 - 2\eta^2\alpha^{2t}\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle. \tag{163}$$

We now simplify the last two terms, starting with the first:

$$\eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 = \eta^2((\alpha^{1t})^2||\boldsymbol{g}_{z_t}||^2 + 2\alpha^{1t}\alpha^{2t}\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle + (\alpha^{2t})^2||\boldsymbol{g}'_{z_t}||^2). \tag{164}$$

Plugging in the expressions for α^{1t}, α^{2t},$\gamma_t = 1$, grouping like terms and simplifying gives the following. Note, we have excluded a few steps due to spatial constraints:

$$\eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 = \frac{(l_{z_t}(\mathbf{w}'_t) - l_{z_t}(\mathbf{w}_t))^2 + 2\eta(l_{z_t}(\mathbf{w}'_t) - l_{z_t}(\mathbf{w}_t))\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle + \eta^2||\boldsymbol{g}_{z_t}||^2||\boldsymbol{g}'_{z_t}||^2}{||\boldsymbol{g}_{z_t} - \boldsymbol{g}'_{z_t}||^2} \tag{165}$$

Plugging in α^{2t} into the remaining term gives the following expressions:

$$-2\eta^2\alpha^{2t}\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle = \frac{-2\eta^2||\boldsymbol{g}_{z_t}||^2\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle - 2\eta(l_{z_t}(\mathbf{w}'_t) - l_{z_t}(\mathbf{w}_t))\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle}{||\boldsymbol{g}_{z_t} - \boldsymbol{g}'_{z_t}||^2}. \tag{166}$$

Putting these together,

$$\begin{aligned}&\eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 - 2\eta^2\alpha^{2t}\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle \\ &= \frac{(l_{z_t}(\mathbf{w}'_t) - l_{z_t}(\mathbf{w}_t))^2 + 2\eta(l_{z_t}(\mathbf{w}'_t) - l_{z_t}(\mathbf{w}_t))\langle \boldsymbol{g}_{z_t}, \boldsymbol{g}'_{z_t}\rangle}{||\boldsymbol{g}_{z_t} - \boldsymbol{g}'_{z_t}||^2} \\ &+ \frac{\eta^2||\boldsymbol{g}_{z_t}||^2||\boldsymbol{g}'_{z_t}||^2 - 2\eta^2||\boldsymbol{g}_{z_t}||^2\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle - 2\eta(l_{z_t}(\mathbf{w}'_t) - l_{z_t}(\mathbf{w}_t))\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle}{||\boldsymbol{g}_{z_t} - \boldsymbol{g}'_{z_t}||^2}.\end{aligned} \tag{167}$$

Cancelling terms gives,

$$\begin{aligned}&\eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 - 2\eta^2\alpha^{2t}\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle \\ &= \frac{(l_{z_t}(\mathbf{w}'_t) - l_{z_t}(\mathbf{w}_t))^2 + \eta^2||\boldsymbol{g}_{z_t}||^2||\boldsymbol{g}'_{z_t}||^2 - 2\eta^2||\boldsymbol{g}_{z_t}||^2\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle}{||\boldsymbol{g}_{z_t} - \boldsymbol{g}'_{z_t}||^2}.\end{aligned} \tag{168}$$

From $\alpha^{2t} \geq 0$ we have $\eta||\boldsymbol{g}_{z_t}||^2 \geq l_{z_t}(\mathbf{w}_t) - l_{z_t}(\mathbf{w}'_t)$ hence we can upper bound $(l_{z_t}(\mathbf{w}'_t) - l_{z_t}(\mathbf{w}_t))^2$ by $\eta^2||\boldsymbol{g}'_{z_t}||^4$:

$$\begin{aligned}&\eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 - 2\eta^2\alpha^{2t}\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle \\ &\leq \frac{\eta^2||\boldsymbol{g}'_{z_t}||^4 + \eta^2||\boldsymbol{g}_{z_t}||^2||\boldsymbol{g}'_{z_t}||^2 - 2\eta^2||\boldsymbol{g}_{z_t}||^2\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle}{||\boldsymbol{g}_{z_t} - \boldsymbol{g}'_{z_t}||^2} \leq \eta^2||\boldsymbol{g}_{z_t}||^2.\end{aligned} \tag{169}$$

Again we use the inequality $||\mathbf{w}_t - \mathbf{w}'_t|| = \eta^2||\boldsymbol{g}_{z_t}|| \leq \eta^2\beta^2||\mathbf{w}_t - \mathbf{w}^*||$:

$$\eta^2||\alpha^{1t}\boldsymbol{g}_{z_t} + \alpha^{2t}\boldsymbol{g}'_{z_t}||^2 - 2\eta^2\alpha^{2t}\langle \boldsymbol{g}'_{z_t}, \boldsymbol{g}_{z_t}\rangle \leq \eta^2||\boldsymbol{g}_{z_t}||^2 \leq \eta^2\beta^2||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \tag{170}$$

Hence we get the following expression:

$$||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \leq (1 - 2\eta\alpha^{1t}\mu - \alpha^{2t}\eta\mu + 2\eta^3\alpha^{2t}\beta^2\mu)||\mathbf{w}_t - \mathbf{w}^*||^2 + \eta^2\beta^2||\mathbf{w}_{t+1} - \mathbf{w}^*||^2\rangle, \tag{171}$$

$$||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \leq (1 - 2\eta\alpha^{1t}\mu - \alpha^{2t}\eta\mu + 2\eta^3\alpha^{2t}\beta^2\mu + \eta^2\beta^2)||\mathbf{w}_t - \mathbf{w}^*||^2, \tag{172}$$

$$||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \leq (1 - \eta\mu - \alpha^{1t}\eta\mu + 2\eta^3\beta^2\mu - 2\eta^3\alpha^{1t}\beta^2\mu + \eta^2\beta^2)||\mathbf{w}_t - \mathbf{w}^*||^2. \tag{173}$$

Upper bounding $-\alpha^{1t}$ by 0,

$$||\mathbf{w}_{t+1} - \mathbf{w}^*||^2 \leq (1 - \eta\mu + 2\eta^3\beta^2\mu + \eta^2\beta^2)||\mathbf{w}_t - \mathbf{w}^*||^2. \tag{174}$$

Taking expectations with respect to z_t:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] \leq \mathbb{E}[(1 - \eta\mu + 2\eta^3\beta^2\mu + \eta^2\beta^2)||\mathbf{w}_t - \mathbf{w}^*||^2]. \tag{175}$$

Noting that $\mathbf{w}_k$ does not depend on z_t, and neither does $\mathbf{w}^*$ due to the interpolation property:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] \leq (1 - \eta\mu + 2\eta^3\beta^2\mu + \eta^2\beta^2)||\mathbf{w}_t - \mathbf{w}^*||^2. \tag{176}$$

For the this step to be convergent we need the following condition to hold $2\eta^3\beta^2\mu+\eta^2\beta^2-\eta\mu \leq 0$. However, for $\eta \leq \hat{\eta} = \min\{\frac{1}{4\beta}, \frac{1}{4\mu}, \frac{\mu}{2\beta^2}\}$ we have:

$$2\eta^3\beta^2\mu + \eta^2\beta^2 - \eta\mu \leq \frac{1}{8}\eta\mu + \frac{1}{2}\eta\mu - \eta\mu \leq -\frac{3}{8}\eta\mu. \tag{177}$$

Hence, we recover the rate:

$$\mathbb{E}[||\mathbf{w}_{t+1} - \mathbf{w}^*||^2] \leq \left(1 - \frac{3}{8}\hat{\eta}\mu\right)^t ||\mathbf{w}_0 - \mathbf{w}^*||^2. \tag{178}$$

■

E.4 Worst Case Rate

It is clear by inspection that the worst case rate derived corresponds to the MAX2 subproblem. Hence in the worst case this step is taken for all t, and thus a trivial induction gives the result of Theorem 2.

Appendix F. Additional Results

F.1 Cifar Hyperparameters and Variance

Here we detail the hyperparameters and variance for the ALI-G and BORAT results reported in table 6. For other optimisation methods please refer to Appendix E of Berrada et al. (2019b).

F.2 Empirical Run Time

Here we detail the effect of increasing N on the run time of BORAT. Due to each update requiring $N-1$ gradient evaluations BORAT with $N \geq 2$ take significantly longer between updates than other methods. However, BORAT achieves good empirical convergence rates taking $N-1$ fewer parameter updates than other methods, as shown in the results section. Hence we consider the epoch time and show for each pass through the data BORAT has a similar run time to SGD.

Increasing N both increases the run time of Algorithm 1 but additionally BORAT must compute extra dot products when calculating Q. A naive implementation of Algorithm 1 has a time complexity of $\mathcal{O}\left(\sum_{k=1}^{N} \frac{N!}{k!(N-k)!}k^3\right)$. However if we exploit the parallel nature of this algorithm where the sub problems are solved simultaneously, the time complexity reduces to

Data Set	Model	Hyperparameters				Test Accuracy	
		N	η	r	batchzise	Mean	std
CIFAR10	WRN	2	0.1	50	128	95.4	0.13
		3	1	100	128	95.4	0.05
		5	1	75	128	95.0	0.08
	DN	2	0.1	100	64	94.5	0.09
		3	1	75	256	94.9	0.13
		5	1	75	128	94.9	0.13
CIFAR100	WRN	2	0.1	50	512	76.1	0.21
		3	0.1	50	256	76.0	0.16
		5	0.1	50	128	75.8	0.22
	DN	2	0.1	75	256	76.2	0.14
		3	0.1	75	128	76.5	0.38
		5	0.1	75	64	75.7	0.03

Table 6: Cifar Hyperparameters (BORAT)

$\mathcal{O}\left(N^3\right)$ as discussed in Section 3.6. Additionally we need only run Algorithm 1 once every $N-1$ batches so the per epoch time complexity is $\mathcal{O}\left(N^2\right)$. Of course in practice Algorithm 1 is only responsible for a fraction of the run time, where its contribution is determined by the relative size of the model and N.
Table 7 show with a parallel implementation the effect of the this extra computation on the training epoch time isn't significant and the time complexity scales approximately linearly with N. Moreover, Table 7 shows for large scale learning problems, such as ImageNet the extra run time when increasing N is negligible.

Optimiser	SGD	BORAT	BORAT	BORAT	BORAT
N	1	2	3	4	5
Time (s)	51.0	55.6	68.2	69.2	74.3
Optimiser	BORAT	BORAT	BORAT	BORAT	BORAT
N	6	7	8	9	10
Time (s)	77.8	82.8	88.7	94.1	99.5

Table 7: Average BORAT training epoch time for CIFAR100 data set, shown for varying N. Time quoted using a batch size of 128, CIFAR100, CE loss, a Wide ResNet 40-4, and a parallel implantation of BORAT. All Optimiser had access to 3 CPU cores, and one TITAN Xp GPU.

Optimiser	BORAT	BORAT	BORAT
N	2	3	5
Time (s)	885.50	910.49	934.79

Table 8: Average BORAT training epoch time for ImageNet data set, shown for varying N. Time quoted using a batch size of 1024, ImageNet, CE loss, a ResNet18, and a parallel implantation of BORAT. All Optimiser had access to 12 CPU cores, and 4 TITAN Xp GPUs.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, GregS. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelioné Man, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernandaégas Vi, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015. Software available from tensorflow.org.

Hilal Asi and John C Duchi. Stochastic (approximate) proximal point methods: Convergence, optimality, and adaptivity. *SIAM Journal on Optimization*, 2019.

Alfred Auslender. Bundle methods for machine learning. *Numerical Methods for Nondifferentiable Convex Optimization. Mathematical Programming Study*, 2009.

Francis R. Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *CoRR*, 2011.

Atilim Gunes Baydin, Robert Cornish, David Martinez Rubio, Mark Schmidt, and Frank Wood. Online learning rate adaptation with hypergradient descent. *International Conference on Learning Representations*, 2018.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd: Compressed optimisation for non-convex problems. *International Conference on Machine Learning*, 2018.

Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Deep Frank-Wolfe for neural network optimization. *International Conference on Learning Representations*, 2019a.

Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Training neural networks for and by interpolation. *International Conference on Machine Learning*, 2019b.

Dimitri P. Bertsekas. *Convex Optimization Theory*. Athena Scientific, 2009.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *Conference on Empirical Methods in Natural Language Processing*, 2015.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 2015.

Jinghui Chen and Quanquan Gu. Padam: Closing the generalization gap of adaptive gradient methods in training deep neural networks. *arXiv preprint*, 2018.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. *International Conference on Learning Representations*, 2019.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *Conference on Empirical Methods in Natural Language Processing*, 2017.

Alexandre Défossez and Francis Bach. Adabatch: Efficient gradient aggregation rules for sequential and parallel stochastic gradient methods. *arXiv preprint*, 2017.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 2011.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 1956.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Conference on Computer Vision and Pattern Recognition*, 2016.

João F Henriques, Sebastien Ehrhardt, Samuel Albanie, and Andrea Vedaldi. Small steps and giant leaps: Minimal newton solvers for deep learning. *International Conference on Computer Vision*, 2019.

Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *Conference on Computer Vision and Pattern Recognition*, 2017.

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

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations*, 2014.

Simon Lacoste-Julien and Martin Jaggi. Block-coordinate Frank-Wolfe optimization for structural SVMs. *International Conference on Machine Learning*, 2013.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Loss functions for top-k error: Analysis and insights. *Conference on Computer Vision and Pattern Recognition*, 2016.

Claude Lemaréchal, Arkadi Nemirovski, and Yurii Nesterov. New variants of bundle methods. *Math. Program*, 1995.

Kfir Levy. Online to offline conversions, universality and adaptive minibatch sizes. *Neural Information Processing Systems*, 2017.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. *Journal of Machine Learning Research*, 2020.

Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. *International Conference on Artificial Intelligence and Statistics*, 2019.

Changliu Liu, Tomer Arnon, Christopher Lazarus, Clark Barrett, and Mykel J Kochenderfer. Algorithms for verifying deep neural networks. *arXiv:1903.06758*, 2019a.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint*, 2019b.

Francesco Locatello, Rajiv Khanna, Michael Tschannen, and Martin Jaggi. A unified optimization view on generalized matching pursuit and frank-wolfe. *International Conference on Artificial Intelligence and Statistics*, 2017.

Nicolas Loizou, Sharan Vaswani, Issam Laradji, and Simon Lacoste-Julien. Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence. *arXiv preprint*, 2020.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations*, 2017.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *International Conference on Learning Representations*, 2019.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *International Conference on Learning Representations*, 2019.

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning. *International Conference on Machine Learning*, 2018a.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *International Conference on Learning Representations*, 2018b.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. *International Conference on Machine Learning*, 2015.

Mahesh Chandra Mukkamala and Matthias Hein. Variants of rmsprop and adagrad with logarithmic regret bounds. *International Conference on Machine Learning*, 2017.

Adam M Oberman and Mariana Prazeres. Stochastic gradient descent with polyak's learning rate. *arXiv preprint*, 2019.

Francesco Orabona and Dávid Pál. Scale-free algorithms for online linear optimization. *International Conference on Algorithmic Learning Theory*, 2015.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. *NIPS Autodiff Workshop*, 2017.

Boris Teodorovich Polyak. Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics*, 1969.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *International Conference on Learning Representations*, 2018.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, 1951.

Michal Rolinek and Georg Martius. L4: Practical loss-based stepsize adaptation for deep learning. *Neural Information Processing Systems*, 2018.

Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. *International Conference on Machine Learning*, 2013.

Frank Schneider, Lukas Balles, and Philipp Hennig. DeepOBS: A deep learning optimizer benchmark suite. *International Conference on Learning Representations*, 2019.

Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 2016.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. *International Conference on Machine Learning*, 2018.

Alexander J. Smola, S. V. N. Vishwanathan, and Quoc V. Le. Bundle methods for machine learning. *Neural Information Processing Systems*, 2007.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Conference on Computer Vision and Pattern Recognition*, 2015.

Conghui Tan, Shiqian Ma, Yu-Hong Dai, and Yuqiu Qian. Barzilai-borwein step size for stochastic gradient descent. *Neural Information Processing Systems*, 2016.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 2012.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron. *International Conference on Artificial Intelligence and Statistics*, 2019a.

Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. *arXiv preprint*, 2019b.

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *Neural Information Processing Systems*, 2017.

Xiaoxia Wu, Rachel Ward, and Léon Bottou. WNGrad: Learn the learning rate in gradient descent. *arXiv preprint*, 2018.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *British Machine Vision Conference*, 2016.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. *Neural Information Processing Systems*, 2018.

Matthew Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint*, 2012.

Ziming Zhang, Yuanwei Wu, and Guanghui Wang. Bpgrad: Towards global optimality in deep learning via branch and pruning. *Conference on Computer Vision and Pattern Recognition*, 2017.

Shuai Zheng and James T Kwok. Follow the moving leader in deep learning. *International Conference on Machine Learning*, 2017.

Yi Zhou, Junjie Yang, Huishuai Zhang, Yingbin Liang, and Vahid Tarokh. Sgd converges to global minimum in deep learning via star-convex path. *International Conference on Learning Representations*, 2019.