
Understanding Progressive Training Through the Framework of Randomized Coordinate Descent

Rafał Szlendak
AI Initiative
KAUST, Saudi Arabia

Elnur Gasanov
AI Initiative
KAUST, Saudi Arabia

Peter Richtárik
AI Initiative
KAUST, Saudi Arabia

Abstract

We propose a Randomized Progressive Training algorithm (RPT) – a stochastic proxy for the well-known Progressive Training method (PT) (Karras et al., 2018). Originally designed to train GANs (Goodfellow et al., 2014), PT was proposed as a heuristic, with no convergence analysis even for the simplest objective functions. On the contrary, to the best of our knowledge, RPT is the first PT-type algorithm with rigorous and sound theoretical guarantees for general smooth objective functions. We cast our method into the established framework of Randomized Coordinate Descent (RCD) (Nesterov, 2012; Richtárik and Takáč, 2014), for which (as a by-product of our investigations) we also propose a novel, simple and general convergence analysis encapsulating strongly-convex, convex and nonconvex objectives. We then use this framework to establish a convergence theory for RPT. Finally, we validate the effectiveness of our method through extensive computational experiments.

1 INTRODUCTION

The modern practice of supervised learning typically relies on training huge-dimensional, overparametrized models using first-order gradient methods. However, in most cases, using the full gradient information in the training procedure is computationally infeasible. The gold-standard approach of bypassing this difficulty is provided by the family of Stochastic Gradient Descent (SGD) algorithms which use only a partial information

about the gradients for the reward of reduced compute time. Progressive Training (PT) (Karras et al., 2018), albeit a non-stochastic method, relies on a similar idea. In its original form, it was developed as a method for training generative adversarial networks (GANs) (Goodfellow et al., 2014) by progressively growing the neural network that we are training. Specifically, for a fixed period of time we only train the first layer of our GAN while keeping the weights corresponding to the other layers intact; then, we perform updates with respect to the first two layers, and so on.

Karras et al. (2018) justified two main advantages of this approach: an increased stability of the training (since the training procedure starts from a simpler, reduced model), and reduced computation cost. The method was proposed as a heuristic, with no convergence guarantees even for the simplest objectives, such as quadratics or convex functions.

Observe that in the PT procedure, the domain of the parameters which we are trying to learn is partitioned into blocks which correspond to the respective layers of the neural network that PT aims to train. Motivated by this, in this work we are going to be concerned with minimizing the general objective functions f whose domain will be written as $\mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_B}$. For example, \mathbb{R}^{d_i} might represent the parameters of the i^{th} layer of the neural network in question.

Algorithm 1 offers a formal description of the PT procedure applied to the general differentiable function $f : \mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_B} \rightarrow \mathbb{R}$. The training procedure is divided into B epochs; in each epoch $b \in [B]$ we want to update the first b coordinate blocks. Let $G_b f(x) \in \mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_B}$ denote the vector

$$(\nabla_1 f(x), \dots, \nabla_b f(x), 0, \dots, 0),$$

where $\nabla_i f$ is the gradient of f with respect to the i^{th} coordinate block. In each epoch b we make the gradient update

$$x \leftarrow x - \gamma_b G_b f(x)$$

for T_b iterations, where $\gamma_b > 0$ is a suitably chosen

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

stepsize which is fixed for the entire epoch b .

Algorithm 1 Progressive Training

- 1: **Input:** initial iterate $x \in \mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_B}$; stepsizes $\gamma_1, \gamma_2, \dots, \gamma_n > 0$; integers $T_1, T_2, \dots, T_n \geq 0$
 - 2: **for** $b = 1, \dots, B$ **do**
 - 3: **for** $t = 0, \dots, T_b - 1$ **do**
 - 4: $x \leftarrow x - \gamma_b G_b f(x)$
 - 5: **end for**
 - 6: **end for**
 - 7: **Output:** x
-

1.1 Existing literature

Wang et al. (2022) attempt to analyze the federated version of PT called ProgFed. Unfortunately, as argued in Appendix A, their analysis is vacuous.

Recently, a novel and related research thread emerged around the idea of *independent subnet training* (IST) which relies on partitioning the neural network in question into sparser subnets, training them in parallel across the devices and periodically aggregating the model. Yuan et al. (2022) introduce IST and provide its analysis. However, the authors make rather strong assumptions on the objective function, such as Lipschitz continuity of the objective function and certain assumptions on the stochastic gradient error. Analogous technique is applied by Dun et al. (2022) for training ResNet (He et al., 2016). It is important to note that the IST framework differs from RPT in several key components. Indeed, IST partitions the neural network in question into sparser submodels which are then trained across the devices *in parallel*. The model is then aggregated and this procedure repeats. In contrast, RPT performs a *single update at a time* of a random subnetwork *chosen in a very particular way* (which simulates the deterministic PT). Moreover, IST is a distributed method; in contrast, in our paper we focus on the single-node setup.

Notably, none of the aforementioned papers contains mathematically sound analysis for general smooth functions.

1.2 Our contributions and organization of the paper

As we have argued, there is still a considerable gap in theoretical understanding of the PT mechanism. *The key objective of this paper is fixing this gap by proposing a randomized proxy for PT which mimics its behavior and has simple and general analysis with mild assumptions.* The contributions of our work can be summarized as follows:

- In Section 2 we develop a simple and general analysis of Randomized Coordinate Descent (RCD) for strongly-convex, convex and nonconvex objectives. Our analysis recovers previous convergence rates of RCD and allows us to fully characterize its iteration complexity in strongly-convex, convex and nonconvex regimes via a single quantity $L_{\mathbf{P}}$.
- In Sections 3 and 4 we propose a method called RPT which mimics the behavior of PT, and, as we argue, is the first PT-type method with sound theoretical guarantees for smooth objectives. Instead of progressively expanding the model which we are trying to optimize, in each iteration RPT makes a choice of the submodel according to a carefully chosen randomness, and then makes a gradient update of this model. We then view RPT as a particular instance of RCD, which immediately establishes its convergence. We finally identify regimes in which RPT can be advantageous over vanilla gradient descent (GD) through a novel and general notion of total computation complexity.
- In Section 5, we present a series of empirical experiments that demonstrate the advantages of our proposed method. These experiments are divided into three sub-sections. The first sub-section employs synthetic data to demonstrate that, when appropriately generated, our proposed Randomized Progressive Training (RPT) method can converge at a faster rate than Gradient Descent while also incurring less computation cost. In the second and third sub-sections, we utilize real-world datasets to evaluate the performance of the RPT method, including testing on a ridge regression problem across various datasets and on an image classification task on the CIFAR10 dataset. The results demonstrate that the RPT method exhibits comparable or even superior performance in comparison to other methods.

1.3 Assumptions

In this section we lay out the assumptions that we are going to make throughout the whole work. In an attempt to make the analysis as general as possible, in this paper we focus on the problem of minimizing an arbitrary differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, where \mathbb{R}^d is decomposed as $\mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_B}$ for some pre-specified d_1, \dots, d_B .

We first recall the notion of *matrix smoothness* (Qu et al., 2016; Safaryan et al., 2021) which generalizes the standard L -smoothness assumption and allows one to encapsulate more curvature information of the objective function.

Definition 1. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ and \mathbf{L} be a positive semidefinite matrix. We say that f is \mathbf{L} -smooth if

$$f(x) - f(y) - \langle \nabla f(y), x - y \rangle \leq \frac{\|x - y\|_{\mathbf{L}}^2}{2}$$

for all $x, y \in \mathbb{R}^d$, where $\|v\|_{\mathbf{L}} \stackrel{\text{def}}{=} \sqrt{v^\top \mathbf{L} v}$.

We make the following assumption throughout the whole paper.

Assumption 1. f is \mathbf{L} -smooth.

With Assumption 1 taken for granted, we analyse RPT in each of the three standard settings:

Assumption 2. f is μ -strongly convex, that is, there exists a positive constant μ such that

$$f(x) - f(y) - \langle \nabla f(y), x - y \rangle \geq \frac{\mu \|x - y\|^2}{2}$$

for all $x, y \in \mathbb{R}^d$.

Assumption 3. f is convex and has a global minimizer x^* .

Assumption 4. f is nonconvex and lower-bounded by $f^* \in \mathbb{R}$.

Under assumptions 2, 3 and 4 respectively we will be interesting in minimizing the following quantities:

$$\mathbb{E} \left[\|x^t - x^*\|^2 \right], \mathbb{E} [f(x^t) - f(x^*)], \mathbb{E} \left[\|\nabla f(x^t)\|^2 \right].$$

2 NEW ANALYSIS OF RANDOMIZED COORDINATE DESCENT

2.1 SkGD

Randomized Coordinate Descent (RCD) is a celebrated technique of minimizing high-dimensional objective functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$, where updating all gradient coordinates is prohibitively expensive. In its original form (Nesterov, 2012; Richtárik and Takáč, 2014) RCD used a *biased* randomized estimator of the gradient; in each iteration it chooses a random sub-block of coordinates and performs a gradient step with respect to that sub-block alone.

Recently, Safaryan et al. (2021) developed an *unbiased* version of RCD (called SkGD) which updates the iterates through *unbiased sketches*. Specifically, in each iteration t , SkGD starts from choosing a random set of coordinates $S^t \subset [d]$. Denote $p_j = \mathbf{Prob}(j \in S^t)$ and assume $p_j > 0$. Then SkGD performs an update of form

$$x^{t+1} = x^t - \gamma \mathbf{C}^t \nabla f(x^t),$$

where

$$\mathbf{C}^t = \text{Diag}(c_1, \dots, c_d), \quad c_j = \begin{cases} 1/p_j & \text{if } j \in S^t \\ 0 & \text{otherwise.} \end{cases}$$

It is then readily verifiable that $\mathbf{C}^t \nabla f(x^t)$ is an unbiased gradient estimator, that is

$$\mathbb{E} [\mathbf{C}^t \nabla f(x^t)] = \nabla f(x^t).$$

In the remainder of this paper we will always denote

$$\mathbf{P} \stackrel{\text{def}}{=} \text{Diag}(p_1, \dots, p_d).$$

2.2 Key quantity: $L_{\mathbf{P}}$

We now define the most important component of our analysis. In the rest of the paper $\lambda_{\max}(\mathbf{M})$ will denote the largest eigenvalue of matrix \mathbf{M} .

Definition 2. Let \mathbf{C} be an unbiased sketch operator and \mathbf{P} its associated probability matrix. Denote the smoothness matrix of the objective function f by \mathbf{L} . Then

$$L_{\mathbf{P}} \stackrel{\text{def}}{=} \lambda_{\max}(\mathbf{P}^{-1/2} \mathbf{L} \mathbf{P}^{-1/2}).$$

It turns out that this quantity describes the convergence of RCD in *all of the regimes in question* (assumptions 2, 3, 4). Therefore, by estimating it for a particular choice of sampling distribution \mathbf{P} one obtains a unified analysis of RCD in the aforementioned regimes. Moreover, in the following lemma we provide sharp estimates for $L_{\mathbf{P}}$.

Lemma 1. Let \mathbf{L} be any positive semidefinite matrix, \mathbf{C} – any unbiased sketch operator, and \mathbf{P} – its probability matrix. Then

$$\lambda_{\max}(\mathbb{E}[\mathbf{CLC}]) \leq L_{\mathbf{P}} \leq \lambda_{\max}(\mathbf{L}) \lambda_{\max}(\mathbf{P}^{-1}).$$

The first bound will be useful in proving the lemma on the convergence of RCD for nonconvex function. The second one quantifies how large can $L_{\mathbf{P}}$ be in terms of the smoothness constant of f and \mathbf{P} . In practice, however, $L_{\mathbf{P}}$ can be much smaller.

2.3 Convergence results

We begin with a preliminary lemma:

Lemma 2. Let $\mathbf{W} = \text{Diag}(w_1, \dots, w_d)$, where w_1, \dots, w_d are any positive entries and assume that $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is \mathbf{L} -smooth. Then

$$\|\nabla f(x)\|_{\mathbf{W}}^2 \leq 2A(f(x) - f(x^*)),$$

where $A = \lambda_{\max}(\mathbf{W}^{1/2} \mathbf{L} \mathbf{W}^{1/2})$.

The following estimate of a second moment of the gradient estimator is then an easy corollary of Lemma 2.

Lemma 3. Let $\mathbf{C} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be an unbiased sketch operator. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be differentiable and denote $g(x) = \mathbf{C}\nabla f(x)$. Then

$$\mathbb{E} \left[\|g(x)\|^2 \right] = \|\nabla f(x)\|_{\mathbf{P}^{-1}}^2 \leq 2L_{\mathbf{P}}(f(x) - f^*).$$

The convergence result for f satisfying Assumption 2 is now an immediate consequence of Corollary A.1 of (Gorbunov et al., 2020), which provides a unified analysis of SGD under strong convexity.

Theorem 1. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be \mathbf{L} -smooth and μ -strongly convex. Let \mathbf{C} be an unbiased sketch with associated probability matrix \mathbf{P} . Then the iterates x^t of SkGD with stepsize $0 < \gamma \leq 1/L_{\mathbf{P}}$ and update rule $x^{t+1} = x^t - \gamma \mathbf{C}\nabla f(x^t)$ satisfy

$$\mathbb{E} \left[\|x^t - x^*\|^2 \right] \leq (1 - \gamma\mu)^t \|x^0 - x^*\|^2.$$

Under assumptions 3 and 4 there also exist unified analyses of SGD (respectively Khaled et al. (2023) and Khaled and Richtárik (2023)) where Lemma 2 is applicable. However, as we argue in Appendix D, applying these generic results to our setting leads to *suboptimal* convergence rates. Nevertheless, in Mishchenko et al. (2023), the analysis of the DIANA algorithm with $n = 1$ and treating the compressor as a sketch results in an asymptotic convergence rate for the non-convex setting that coincides with the rate proposed in our research. However, this analysis is based on the assumption of constant smoothness, in contrast to the matrix smoothness adopted in our study. We instead develop a novel analysis of SkGD in convex and non-convex regimes and present it as Theorems 2 and 3 respectively.

Theorem 2. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be \mathbf{L} -smooth, convex, and lower-bounded. Furthermore, suppose that f attains its minimum at x^* . Let $(x^t)_{t=0}^{\infty}$ be the iterates of the SkGD with unbiased sketch \mathbf{C} and stepsize

$$0 < \gamma \leq \frac{1}{2L_{\mathbf{P}}}.$$

Denote $\bar{x}^T = \frac{1}{T+1} \sum_{t=0}^T x^t$. Then

$$\mathbb{E} [f(\bar{x}^T) - f(x^*)] \leq \frac{\|x^0 - x^*\|^2}{\gamma(T+1)}.$$

In particular, for $\gamma = 1/(2L_{\mathbf{P}})$ and $T + 1 \geq \frac{2L_{\mathbf{P}}\|x^0 - x^*\|^2}{\varepsilon}$, we have that

$$\mathbb{E} [f(\bar{x}^T) - f(x^*)] \leq \varepsilon.$$

Theorem 3. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be \mathbf{L} -smooth and lower bounded by f^* . Let $(x^t)_{t=0}^{\infty}$ be the iterates of SkGD with unbiased sketch \mathbf{C} and stepsize

$$0 < \gamma \leq \frac{1}{L_{\mathbf{P}}}.$$

Denote $\hat{x}^T = \operatorname{argmin}_{t=0, \dots, T} \|\nabla f(x^t)\|^2$. Then

$$\mathbb{E} \left[\|\nabla f(\hat{x}^T)\|^2 \right] \leq \frac{2\delta^0}{\gamma(T+1)},$$

where $\delta^0 = f(x^0) - f^*$. In particular, for $\gamma = 1/L_{\mathbf{P}}$ and $T + 1 \geq \frac{2L_{\mathbf{P}}\delta^0}{\varepsilon}$, we have that

$$\mathbb{E} \left[\|\nabla f(\hat{x}^T)\|^2 \right] \leq \varepsilon.$$

3 RANDOMIZED PROGRESSIVE TRAINING

3.1 Problem formulation

For the sake of consistency with the PT framework, we shall decompose the domain of the objective function f into B disjoint subblocks. Thus, we will view the domain of f as $\mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_B}$, where $\sum_{i=1}^B d_i = d$. We will further assume that f is \mathbf{L} -smooth, where \mathbf{L} is a symmetric, positive definite matrix of size $d \times d$. We let \mathbf{L}_i be the i^{th} diagonal subblock of \mathbf{L} of size $d_i \times d_i$. We put $L_i \stackrel{\text{def}}{=} \lambda_{\max}(\mathbf{L}_i)$. We will be interested in minimizing this function under assumptions 2, 3 or 4, i.e. finding

$$\operatorname{argmin}_{x \in \mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_B}} f(x).$$

3.2 Randomized Progressive Training

We now introduce the *PT-sketch operator*, the key building block of the Randomized Progressive Training (RPT) algorithm.

Definition 3. Let $1 = p_1 \geq p_2 \geq \dots \geq p_B > 0$. We additionally put $p_{B+1} = 0$. Let $\sigma = (\sigma_1, \dots, \sigma_B)$ be any permutation of set $[B]$. For each $i \in [B]$ define the matrix $\mathbf{C}_i \in \mathbb{R}^{d \times d}$ as follows:

$$\mathbf{C}_i = \operatorname{Diag} \left((\xi_{\sigma_1}^i)_{d_1}, \dots, (\xi_{\sigma_B}^i)_{d_B} \right),$$

where $\operatorname{Diag}(v)$ denotes the diagonal matrix with v as its diagonal, $(x)_m \stackrel{\text{def}}{=} (x, \dots, x) \in \mathbb{R}^m$, and

$$\xi_j^i = \begin{cases} 1/p_j & \text{if } j \leq i \\ 0 & \text{otherwise.} \end{cases}$$

We finally define a random matrix \mathbf{C} which is equal to \mathbf{C}_i with probability $p_i - p_{i+1}$ and call it a *PT-sketch*.

With this definition, we are now ready to formulate the RPT algorithm.

Algorithm 2 Randomized Progressive Training

- 1: **Input:** initial iterates $x^0 \in \mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_B}$; step-size γ ; integer $T \geq 0$
 - 2: **for** $t = 1, \dots, T$ **do**
 - 3: Draw the PT-sketch \mathbf{C}^t
 - 4: $x^{t+1} = x^t - \gamma \mathbf{C}^t \nabla f(x^t)$
 - 5: **end for**
 - 6: **Output:** x^T
-

RPT is an instance of SkGD with probability matrix

$$\mathbf{P} = \text{Diag}((p_{\sigma_1})_{d_1}, \dots, (p_{\sigma_B})_{d_B}).$$

Hence, theorems 1, 2, 3 immediately apply to it. Our task is now to choose \mathbf{P} in a way that minimizes the total computation complexity.

Remark 1. There are two main differences between RPT and classical approaches to progressive training (Karras et al., 2018; Wang et al., 2022). Firstly, our method chooses the sizes of coordinate updates in a randomized fashion. PT, however, is a deterministic method which grows the coordinate updates according to a pre-set schedule.

Moreover, PT assumes a particular order of importance of the coordinate blocks – it *always* updates the first coordinate block, then includes the second block, and so on. However, we note that in general there is no particular reason to prioritize the first coordinate block, and RPT allows us to decide on the order of importance by choosing the permutation σ in Definition 3.

Remark 2. The formulation of RPT is stemming from the problem formulation described in Section 3.1, which assumes a particular division of the domain elements into B coordinate blocks. While our analysis holds for any such division, we do not specify how to decide on it, noting that it is often problem-specific (for instance, a coordinate block might be a layer of a neural network (Karras et al., 2018; Wang et al., 2022)).

4 ANALYSIS OF RPT

4.1 Cost model

In this section we will define the computation cost of optimizing the objective function from Section 3.1. By construction, in the RPT procedure the variable $x^t \in \mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_B}$ is divided into B coordinate blocks and updated with respect to a certain subset of these blocks. In our discussion we will assume that by

carrying out an update with respect to block i the algorithm suffers a penalty c_i . Hence, if $\mathbf{C}^t = \mathbf{C}_i$, then cost_t , the cost incurred in the iteration t is equal to $\sum_{j=1}^i c_j$. Moreover, without loss of generality we make a simplifying normalizing assumption on the costs c_i for the sake of presentation.

Assumption 5. $\sum_{i=1}^B c_i = 1$.

Now, recall that $\mathbf{C} = \mathbf{C}_i$ with probability $p_i - p_{i+1}$. Therefore, the expected cost of the entire optimization procedure is

$$\begin{aligned} \mathbb{E}[\text{cost}(T)] &\stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{t=1}^T \text{cost}_t\right] \\ &= T \sum_{i=1}^B (p_i - p_{i+1}) \sum_{j=1}^i c_{\sigma_j} \\ &= T \sum_{i=1}^B p_i c_{\sigma_i}. \end{aligned}$$

Previous convergence results show that in order to reach the ε -accurate solution in all regimes described by assumptions 2, 3, and 4, one needs T to be proportional to $L_{\mathbf{P}}$. Thus,

$$\mathbb{E}[\text{cost}(T)] \propto c(\mathbf{P}) \stackrel{\text{def}}{=} L_{\mathbf{P}} \sum_{i=1}^B p_i c_{\sigma_i}$$

regardless of the regime in question. Thus, choosing the optimal RPT sketch is equivalent to solving the minimization problem

$$\min_{\sigma} \min_{1=p_1 \geq \dots \geq p_B > 0} L_{\mathbf{P}} \sum_{i=1}^B p_i c_{\sigma_i}. \quad (1)$$

4.2 Choosing the probabilities p_i

Finding the global minimum of the quantity $c(\mathbf{P})$ in general is a challenging task when \mathbf{L} is simply any positive definite matrix; $c(\mathbf{P})$ is a non-convex function that can intricately depend on the structure of matrix \mathbf{L} . However, note that taking $p_i = 1$ for all $i \in [B]$ recovers gradient descent, i.e. the optimal $c(\mathbf{P})$ is at most L . Thus, the solution of Problem (1) is not worse than gradient descent. Furthermore, one can find a closed-form minimizer for the *tight upper bound* for $L_{\mathbf{P}}$. We first invoke Lemma 1 from Nesterov (2012).

Lemma 4. *Consider an arbitrary positive semidefinite matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ with diagonal subblocks $\mathbf{M}_1, \dots, \mathbf{M}_B$, where each $\mathbf{M}_i \in \mathbb{R}^{d_i \times d_i}$. Put $M_i = \lambda_{\max}(\mathbf{M}_i)$ and $M = \lambda_{\max}(\mathbf{M})$. Then*

$$\max_i M_i \leq M \leq \sum_{i=1}^B M_i.$$

In Appendix C we argue that these bounds are tight. We then have

$$c(\mathbf{P}) \leq \sum_{i=1}^B \frac{L_{\sigma_i}}{p_i} \cdot \sum_{i=1}^B p_i c_{\sigma_i}.$$

The below lemma describes the global minimizer of this upper bound.

Lemma 5.

$$\sum_{i=1}^B \frac{L_{\sigma_i}}{p_i} \cdot \sum_{i=1}^B p_i c_{\sigma_i} \geq \left(\sum_{i=1}^B \sqrt{L_i c_i} \right)^2.$$

Moreover, this bound is met by taking σ to be such that

$$\sqrt{\frac{L_{\sigma_1}}{c_{\sigma_1}}} \geq \dots \geq \sqrt{\frac{L_{\sigma_B}}{c_{\sigma_B}}},$$

and

$$p_i = \frac{\sqrt{L_{\sigma_i}/c_{\sigma_i}}}{\max_j \sqrt{L_j/c_j}}.$$

This choice of probabilities is similar to the one proposed by Zhu et al. (2016) (which, contrary to our paper, treats *accelerated version* of RCD) where the probabilities p_i are proportional to $\sqrt{L_i}$. However, Zhu et al. (2016) consider the setup of non-overlapping blocks and analyse iteration complexity only. On the contrary, our work, to the best of our knowledge, is the first one to consider such choice of probabilities in a non-accelerated setting and motivates this choice through the total complexity analysis.

We summarize the discussion with the corollary:

Corollary 1. *Let \mathbf{P} be the optimal solution of Equation (1). Then,*

$$c(\mathbf{P}) \leq \min \left\{ L, \left(\sum_{i=1}^B \sqrt{L_i c_i} \right)^2 \right\}.$$

5 EXPERIMENTS

We conduct several experiments to evaluate the performance of Randomized Progressive Training in comparison to other methods on various target functions and datasets.

5.1 Quadratic functions on synthetic data

In the first experiment, we show how fast Randomized Progressive Training can be compared to vanilla Gradient Descent. When running RPT with probabilities proposed in Lemma 5, theory suggests that total

computation cost of

$$O \left(\frac{\left(\sum_{i=1}^B \sqrt{L_i c_i} \right)^2}{\varepsilon} \right)$$

is required to achieve ε -accuracy. At the same time, standard theory for GD under Assumption 5 (meaning that a cost of one GD iteration is one) suggests $O\left(\frac{L}{\varepsilon}\right)$ total cumulative cost. We vary block smoothness constants L_i and computation costs c_i so that the ratio $(\sum_{i=1}^B \sqrt{L_i c_i})^2/L$ is relatively large. In this vein, we generate nine quadratic functions $f(x) = \frac{1}{2}x^\top \mathbf{A}x$ with varying dimensions and properties of the *diagonal* matrix \mathbf{A} . The number of blocks is three in all experiments. We consider the computation cost c_i of block i to be proportional to its size. As shown on Figure 1, when the size of the block with the smallest smoothness constant is much larger than one of the largest, RPT shows significant speedup over GD. Table 1 compares speedups predicted by theory with actual ones.

Setup				Theory speedup	Actual speedup
dims	10	10	10	1.1	1.9
Ls	272	53.3	11		
dims	10	50	250	3.4	19.8
Ls	270.5	55	11		
dims	10	100	1000	5.7	26.9
Ls	256.7	54.8	11		
dims	10	500	25000	12.7	36.6
Ls	274.8	55	11		
dims	10	10	10	1.5	2.2
Ls	1066.5	108.4	11		
dims	10	50	250	6.4	17.6
Ls	1080.4	109.6	11		
dims	10	100	1000	12.3	64.9
Ls	1093.9	109.7	11		
dims	10	500	25000	36.6	128.1
Ls	1066.7	110	11		
dims	10	10	10	2.2	2.8
Ls	27414	547	11		
dims	10	50	250	15.4	20
Ls	27166	550	11		
dims	10	100	1000	40.1	62.4
Ls	26020	549	11		
dims	10	500	25000	282.5	407.8
Ls	27363	550	11		

Table 1: Comparison of theoretical and actual speedup of Randomized Progressive Training versus Gradient Descent with quadratic functions. In the "Setup" section, "dims" refers to the dimension of a block and "Ls" denotes the corresponding smoothness constant.

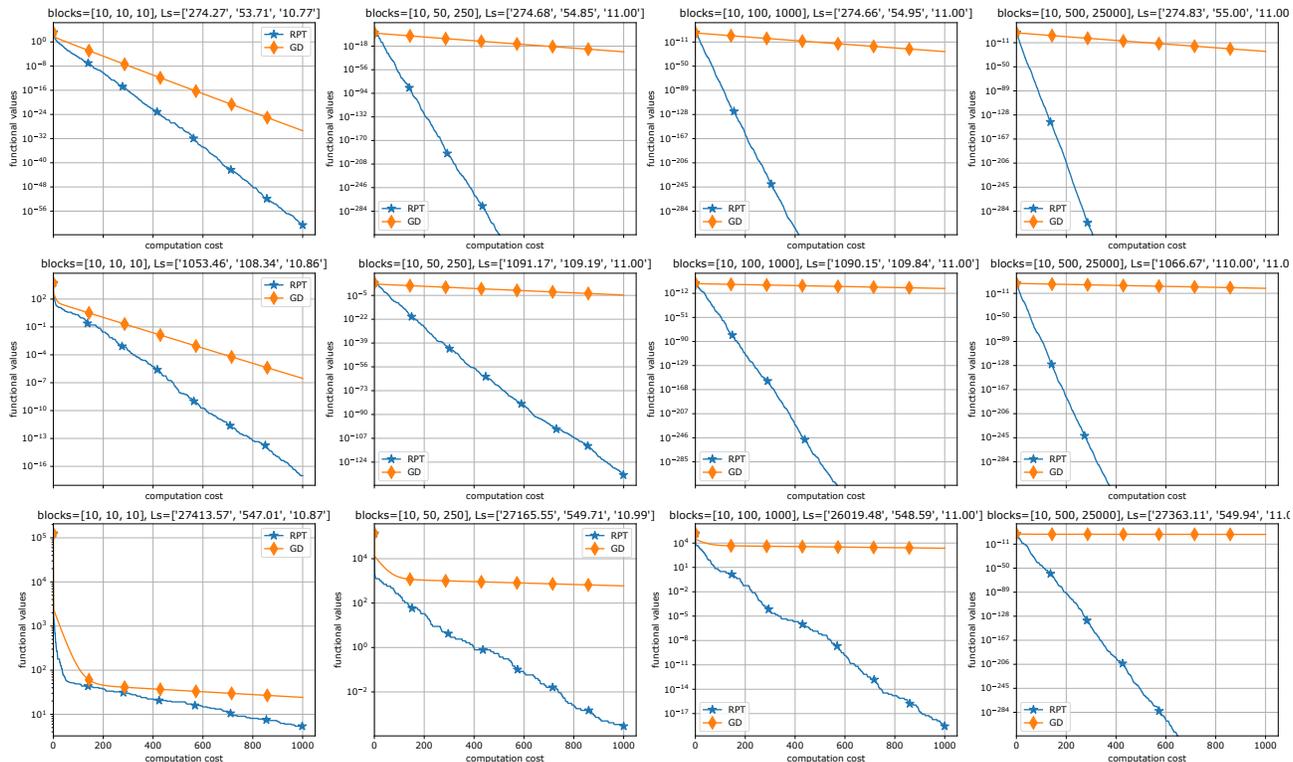


Figure 1: Comparison of Randomized Progressive Training (RPT) Gradient Descent (GD) on quadratic functions. Target function is $f(x) = \frac{1}{2}x^\top \mathbf{A}x$, where the *diagonal* matrix $\mathbf{A} \in \mathbb{R}^{l \times l}$ and its dimension l differ in each plot. For RPT, the iterate is separated into three blocks. 'blocks' in the plot title indicates the block sizes, 'Ls' indicates smoothness constants within the corresponding block. Numbers in 'blocks' sum up to l . The block computation cost c_i is the size of block i divided by l , and the computation cost of a single GD iteration is one. Probabilities for RPT are set as stated in Lemma 5.

5.2 Ridge regression on real datasets

In our second set of experiments, we assess the convergence of Randomized Progressive Training, Gradient Descent, and Cyclic Block Coordinate Descent (CBCD) (Sun and Hong, 2015) on the ridge regression problem

$$\min f(x) = \frac{1}{n} \sum_{i=1}^n (\langle a_i, x \rangle - b_i)^2 + \lambda \|x\|_2^2,$$

where $a_i \in \mathbb{R}^d$, $b_i \in \mathbb{R}$ and λ is a regularization parameter. We consider four regression datasets: California housing (Kelley Pace and Barry, 1997), prostate cancer, Los Angeles ozone (Hastie et al., 2009), and white wine quality (Cortez et al., 2009). Each dataset feature is normalized, and a bias vector is added. Then, heuristically, features are permuted so that the diagonal of the Hessian matrix is sorted in descending order. As in the previous experiment, we consider c_i proportional to the corresponding block size.

As can be seen from Figure 2, although the theoretical speedup of RPT over GD, according to Table 2, is close

Dataset	Theory speedup	Actual speedup
california housing	1	1.3
prostate cancer	1	2.1
los angeles ozone	1	2.4
white wine quality	1.1	1.3

Table 2: Comparison of theoretical and actual speedup of Randomized Progressive Training versus Gradient Descent for ridge regression problems.

to 1, the actual speedup is still noteworthy. However, both algorithms are slower than Cyclic Block Coordinate Descent.

5.3 Image classification on CIFAR10

In our final computational experiment, we trained a Deep Residual Neural Network (He et al., 2016) with 18 layers (ResNet18) for the task of image classification on the CIFAR10 dataset (Krizhevsky, 2009).

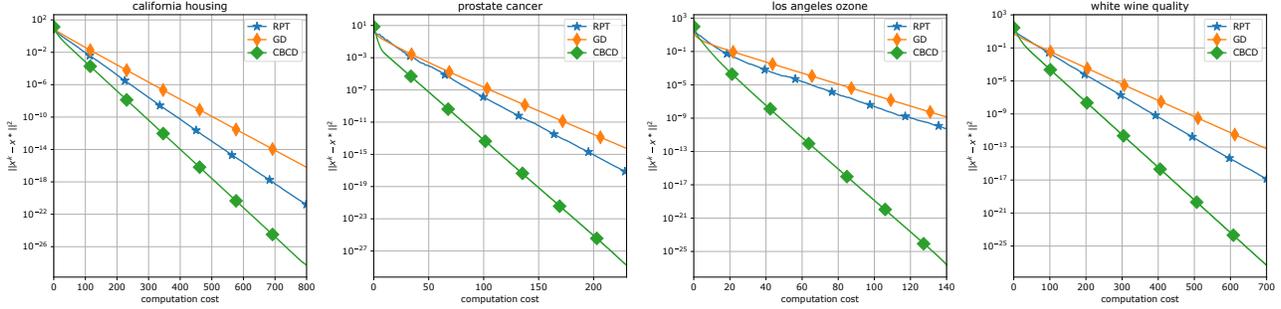


Figure 2: Comparison of Randomized Progressive Training (RPT) vs. Gradient Descent (GD) and Cyclic Block Coordinate Descent (CBCD) on the ridge regression problem. The block computation cost c_i is the size of block i divided by the dimension of the problem, and the computation cost of one GD iteration is 1. Probabilities of block sampling for RPT are set as stated in Lemma 5.

Implementation. The CIFAR10 dataset comprises 50’000 training samples and 10’000 test samples, which are distributed evenly among 10 classes.. The samples are RGB images of size 32×32 . To facilitate hyperparameter tuning, we further divided the training dataset into training and validation datasets with a ratio 9 : 1. During the training process, we applied a padding of 4, randomly cropped to size 32×32 , and randomly flipped images horizontally. We trained the neural network with three algorithms: Stochastic Gradient Descent, Progressive Training (Wang et al., 2022), and our own method. The batch size remains fixed at 128 for each method. We implemented all settings using the Python packages JAX (Bradbury et al., 2018), Flax (Heek et al., 2020), and Optax.

Computation cost. The computation cost associated with a convolutional layer in ResNet18 is estimated by calculating the ratio of the total number of elements in the kernel to the total number of parameters present in ResNet18. Similarly, the computation cost attributed to the fully connected layer is determined by dividing the total number of weights within that layer by the overall number of parameters. It is noted for simplicity that the computation cost associated with the batch normalization layer is disregarded and considered as zero. These computations reflect the costs incurred for processing the entire batch.

Fine-tuning. Fine-tuning is a crucial step in the process of training neural networks, as the lack of smoothness and convexity properties inherent in these models often renders the application of theoretical results ineffective (Szegedy et al., 2014). To address this, a common approach for hyperparameter tuning is to utilize a grid search method. In our case, for the Randomized Progressive Training method, we fine-tuned the learning rate and sampling probabilities. A grid search was conducted for the learning rate over the values $[1, 0.5, 0.1, 0.05, 0.01, 0.001]$, and for sampling

probabilities, we evaluated two different strategies. The first strategy used exponentially decaying probabilities with rates $[0.999, 0.995, 0.99, 0.95, 0.9, 0.85]$, *e.g.*, for a rate of 0.9, the probabilities were calculated as $(p_1, \dots, p_{22}) = (1, 0.9, 0.9^2, \dots, 0.9^{21})$. The second strategy used evenly spaced probabilities between $p_1 = 1$ and p_{22} , where p_{22} ran over the list $[0.7, 0.6, 0.5, 0.4, 0.3, 0.2]$. The best results were obtained with a learning rate of 1 and exponentially decaying sampling probabilities with a rate of 0.9. For Progressive Training (PT), we only fine-tuned the learning rate and utilized the round sizes T_1, \dots, T_{22} suggested in (Wang et al., 2022) as the best strategy. Specifically, we set $T_1 = \dots = T_{21} = \frac{T}{44}$, and $T_{22} = \frac{23T}{44}$, where T is the total number of steps. The grid search for the learning rate was the same as for RPT. Additionally, Stochastic Gradient Descent parameters such as the learning rate, weight decay, and momentum were set according to the guidelines in (Moreau et al., 2022). The batch size in all three settings was 128.

Results. The results of our image classification task are summarized in Figure 3. While it may appear that RPT converges more slowly and has a lower test accuracy than SGD, based on Figures 3c and 3d, it is important to note that this comparison is not entirely fair as one RPT step can take significantly less time than one SGD step. To more accurately compare the performance of RPT to its competitors, we have evaluated the performance versus computation cost, as shown in Figures 3a and 3b. From these figures, it is clear that RPT outperforms both competitors.

Remark 3. In our practical deployment of Algorithms 1 and 2, at every iteration, we initially compute the entire gradient, retaining only those elements as dictated by the algorithm. This procedure accurately emulates algorithms but proves to be impractical for real-world application due to the necessity of comput-

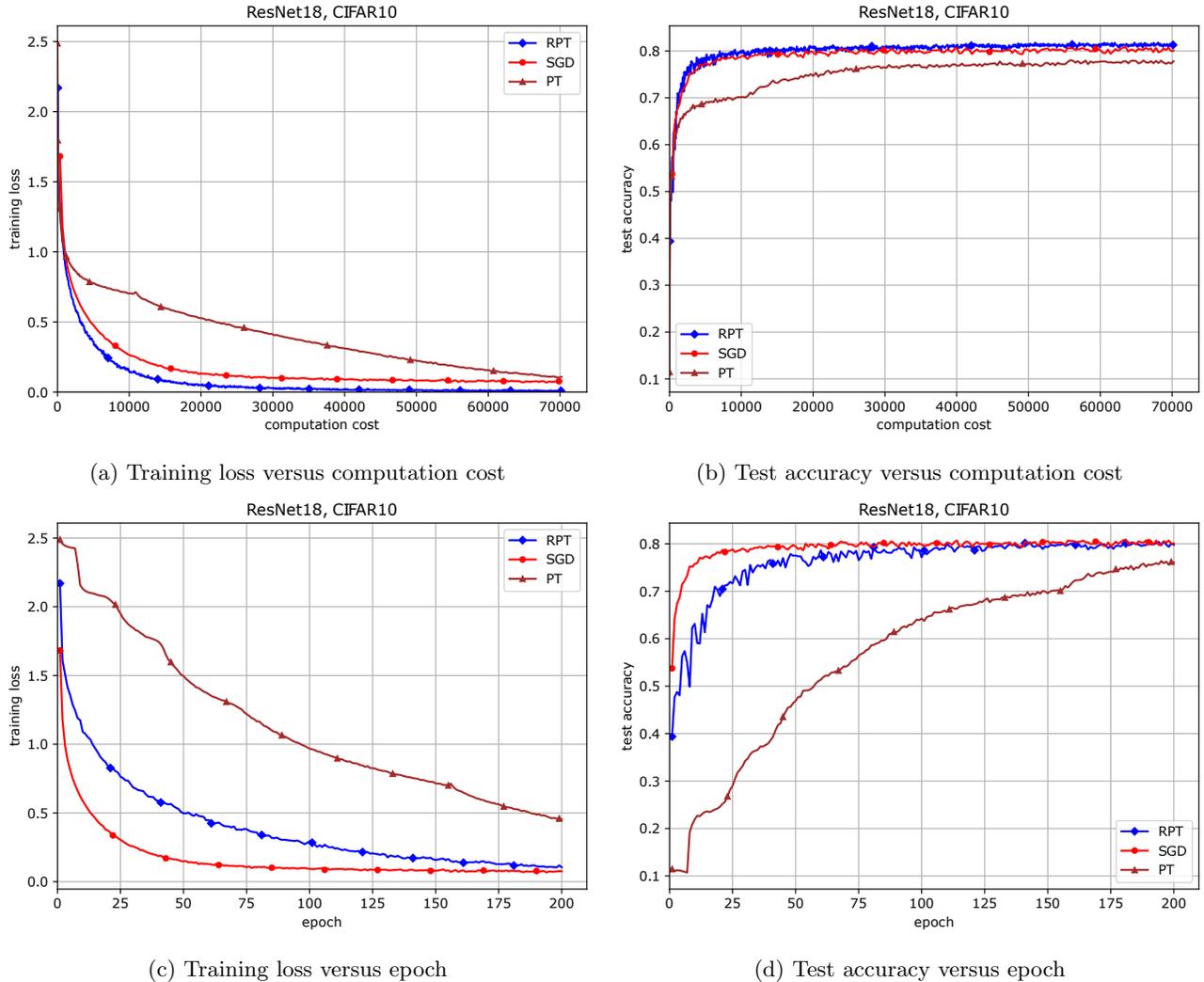


Figure 3: Comparison of Randomized Progressive Training (RPT), Stochastic Gradient Descent (SGD), and Progressive Training (PT) on an image classification task using a ResNet18 model. The block computation cost c_i is proportional to the number of elements in the kernel for convolution layers, the number of weights in the fully connected layer, and is disregarded (assigned a value of zero) for the batch normalization layer. Normalization is implemented in such a manner that the cumulative computation cost across all blocks for a single batch equals 1. Consequently, the computation cost for one iteration of Stochastic Gradient Descent over a batch is also one.

ing the full gradient, effectively nullifying any theoretical advantages. Consequently, comparing the efficiency of Randomized Progressive Training against a robustly implemented Stochastic Gradient Descent, as included in the standard Optax package, particularly regarding *time* efficiency, proves to be an inequitable evaluation.

6 Limitations

The primary constraint of this study resides in the computational challenges associated with the determi-

nation of the constants denoted as L_i within a real-world context, which forms the foundation for the theoretical derivations presented. Furthermore, a critical assumption underpinning this study pertains to the smoothness of the objective function, a premise that may not inherently hold true for neural networks.

References

- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.
- Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553. Smart Business Networks: Concepts and Empirical Evidence.
- Dun, C., Wolfe, C. R., Jermaine, C. M., and Kyrillidis, A. (2022). ResIST: layer-wise decomposition of ResNets for distributed training. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Conference on Neural Information Processing Systems (NeurIPS)*.
- Gorbunov, E., Hanzely, F., and Richtarik, P. (2020). A unified theory of SGD: variance reduction, sampling, quantization and coordinate descent. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.
- Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. (2020). Flax: A neural network library and ecosystem for JAX.
- Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation.
- Kelley Pace, R. and Barry, R. (1997). Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297.
- Khaled, A. and Richtárik, P. (2023). Better theory for SGD in the nonconvex world. *Transactions on Machine Learning Research*.
- Khaled, A., Sebbouh, O., Loizou, N., Gower, R. M., and Richtárik, P. (2023). Unified analysis of stochastic gradient methods for composite convex and smooth optimization. *Journal of Optimization Theory and Applications*.
- Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto.
- Mishchenko, K., Gorbunov, E., Takáč, M., and Richtárik, P. (2023). Distributed learning with compressed gradient differences.
- Moreau, T., Massias, M., Gramfort, A., Ablin, P., Bannier, P.-A., Charlier, B., Dagréou, M., Dupré la Tour, T., Durif, G., F. Dantas, C., Klopfenstein, Q., Larsson, J., Lai, E., Lefort, T., Malézieux, B., Moufad, B., T. Nguyen, B., Rakotomamonjy, A., Ramzi, Z., Salmon, J., and Vaiter, S. (2022). Benchopt: Reproducible, efficient and collaborative optimization benchmarks. In *Conference on Neural Information Processing Systems (NeurIPS)*.
- Nesterov, Y. (2012). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362.
- Qu, Z., Richtarik, P., Takac, M., and Fercoq, O. (2016). SDNA: stochastic dual newton ascent for empirical risk minimization. In *International Conference on Machine Learning (ICML)*.
- Richtárik, P. and Takáč, M. (2014). Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1):1–38.
- Safaryan, M., Hanzely, F., and Richtarik, P. (2021). Smoothness matrices beat smoothness constants: better communication compression techniques for distributed optimization. In *Conference on Neural Information Processing Systems (NeurIPS)*.
- Sun, R. and Hong, M. (2015). Improved iteration complexity bounds of cyclic block coordinate descent for convex problems. In *Conference on Neural Information Processing Systems (NeurIPS)*.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*.
- Wang, H.-P., Stich, S., He, Y., and Fritz, M. (2022). ProgFed: effective, communication, and computation efficient federated learning by progressive training. In *International Conference on Machine Learning (ICML)*.
- Yuan, B., Wolfe, C. R., Dun, C., Tang, Y., Kyrillidis, A., and Jermaine, C. (2022). Distributed learning of fully connected neural networks using independent subnet training. *Proceedings of the VLDB Endowment*.
- Zhu, Z. A., Qu, Z., Richtárik, P., and Yuan, Y. (2016). Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning (ICML)*.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes]
 - (b) Complete proofs of all theoretical results. [Yes]
 - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes]
 - (b) The license information of the assets, if applicable. [Not Applicable]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
 - (d) Information about consent from data providers/curators. [Not Applicable]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. [Not Applicable]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A ISSUES WITH THE THEORY FOR PROGFED

We will first briefly outline the notation that Wang et al. (2022) use in their paper.

We denote by \mathcal{M} the neural network that we aim to train. Put \mathcal{M}^s to be network \mathcal{M} truncated to its first s layers. Let \mathcal{L} denote loss function of interest and put f and f^s to be $\mathcal{L} \circ \mathcal{M}$ and $\mathcal{L} \circ \mathcal{M}^s$ respectively. Denote \mathbf{x}^s to be the parameter vector of \mathcal{M}^s . Finally, for $s \leq i \leq S$ we put $\mathbf{x}_{E_s}^i$ and $\nabla f^i(\mathbf{x}^i)|_{E_s}$ to be the projections of vectors \mathbf{x}^i and $\nabla f^i(\mathbf{x}^i)$ onto the parameter space of \mathcal{M}^s .

The first result by Wang et al. (2022) proves that

$$\frac{1}{T} \sum_{t=0}^{T-1} \alpha_t^2 \|\nabla f^s(\mathbf{x}_t^s)|_{E_s}\| < \varepsilon$$

after $\mathcal{O}\left(\frac{\sigma^2}{\varepsilon^2} + \frac{1}{\varepsilon}\right) \cdot LF_0$ iterations. Here \mathbf{x}_t^s is the t^{th} iterate of ProgFed,

$$\alpha_t = \min \left\{ 1, \frac{\langle \nabla f(\mathbf{x}_t)|_{E_s}, \nabla f^s(\mathbf{x}_t^s)|_{E_s} \rangle}{\|\nabla f^s(\mathbf{x}_t^s)\|^2} \right\},$$

L is the smoothness constant of f , and F_0 is the function suboptimality. Thus, the numbers α_t depend on the iterates x_t . No bounds on α_t are provided. In particular, the theory does not exclude the possibility that α_t^2 are extremely close to 0. This renders the result meaningless.

The second result that Wang et al. (2022) attempts to bound $\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla f(\mathbf{x}_t)\|^2$. Unfortunately, it suffers from a similar issue. The authors claim that $\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla f(\mathbf{x}_t)\|^2$ after

$$\mathcal{O}\left(\frac{q^4 \sigma^2}{\varepsilon^2} + \frac{q^2}{\varepsilon}\right) \cdot LF_0,$$

where $q := \max_{t \in [T]} \left(q_t := \frac{\|\nabla f(\mathbf{x}_t)\|}{\alpha_t \|\nabla f^s(\mathbf{x}_t^s)|_{E_s}\|} \right)$. Wang et al. (2022) do not provide any bound on q , which does not depend on the iterates. Thus, it is not possible to decide *a priori* how many iterations do we need in order to reach the ε -accurate solution, because the theory is dependent on the iterates *produced during the run of the method*. In other words, their second convergence result (in its current state) does not exclude an infinite runtime, so in fact it does not show the convergence at all.

B PROOFS

B.1 Proof of Lemma 1

Lemma 1. *Let \mathbf{L} be any positive semidefinite matrix, \mathbf{C} – any unbiased sketch operator, and \mathbf{P} – its probability matrix. Then*

$$\lambda_{\max}(\mathbb{E}[\mathbf{CLC}]) \leq L_{\mathbf{P}} \leq \lambda_{\max}(\mathbf{L}) \lambda_{\max}(\mathbf{P}^{-1}).$$

Proof. We start from proving the first inequality. Let $x \in \mathbb{R}^d$. Then

$$\begin{aligned} x^\top \mathbf{P}^{1/2} \mathbb{E}[\mathbf{CLC}] \mathbf{P}^{1/2} x &= \mathbb{E} \left[\left\| \mathbf{L}^{1/2} \mathbf{C} \mathbf{P}^{1/2} x \right\|^2 \right] \\ &\leq \lambda_{\max}(\mathbf{L}) \mathbb{E} \left[\left\| \mathbf{C} \mathbf{P}^{1/2} x \right\|^2 \right] \\ &= \lambda_{\max}(\mathbf{L}) \mathbb{E} \left[\sum_{i=1}^n \frac{1}{p_i} x_i^2 \chi_{\{\mathbf{C}=\mathbf{C}_i\}} \right] \\ &= \lambda_{\max}(\mathbf{L}) \sum_{i=1}^n x_i^2 \frac{1}{p_i} p_i \\ &= \lambda_{\max}(\mathbf{L}) \|x\|^2. \end{aligned}$$

Thus,

$$\lambda_{\max}(\mathbf{P}^{1/2}\mathbb{E}[\mathbf{CLC}]\mathbf{P}^{1/2}) \leq \lambda_{\max}(\mathbf{L}),$$

i.e.

$$\lambda_{\max}(\mathbb{E}[\mathbf{CLC}]) \leq \lambda_{\max}(\mathbf{P}^{-1/2}\mathbf{LP}^{-1/2}).$$

For the second inequality, similarly,

$$\begin{aligned} x^\top \mathbf{P}^{-1/2}\mathbf{LP}^{-1/2}x &= \left\| \mathbf{L}^{1/2}\mathbf{P}^{-1/2}x \right\|^2 \\ &\leq \lambda_{\max}(\mathbf{L}) \left\| \mathbf{P}^{-1/2}x \right\|^2 \\ &\leq \lambda_{\max}(\mathbf{L})\lambda_{\max}(\mathbf{P}^{-1}) \|x\|^2. \end{aligned}$$

Thus,

$$\lambda_{\max}(\mathbf{P}^{-1/2}\mathbf{LP}^{-1/2}) \leq \lambda_{\max}(\mathbf{L})\lambda_{\max}(\mathbf{P}^{-1}).$$

□

B.2 Proof of Lemma 2

Lemma 2. Let $\mathbf{W} = \text{Diag}(w_1, \dots, w_d)$, where w_1, \dots, w_d are any positive entries and assume that $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is \mathbf{L} -smooth. Then

$$\|\nabla f(x)\|_{\mathbf{W}}^2 \leq 2A(f(x) - f(x^*)),$$

where $A = \lambda_{\max}(\mathbf{W}^{1/2}\mathbf{LW}^{1/2})$.

Proof. Fix $\alpha \in \mathbb{R}$ and $x \in \mathbb{R}^d$. Since x^* is a minimizer of f and since f is \mathbf{L} -smooth, we have

$$\begin{aligned} f(x^*) &\leq f(x - \alpha\mathbf{W}\nabla f(x)) \\ &\leq f(x) - \alpha\langle \nabla f(x), \mathbf{W}\nabla f(x) \rangle + \frac{\alpha^2}{2} \|\mathbf{W}\nabla f(x)\|_{\mathbf{L}}^2 \\ &= f(x) - \alpha\langle \nabla f(x), \mathbf{W}\nabla f(x) \rangle + \frac{\alpha^2}{2} \langle \nabla f(x), \mathbf{WLW}\nabla f(x) \rangle \end{aligned}$$

By definition of A , we have $\mathbf{W}^{1/2}\mathbf{LW}^{1/2} \preceq A\mathbf{I}_d$, which implies that $\mathbf{WLW} \preceq A\mathbf{W}$. Plugging this estimate into the above inequality, we get

$$\begin{aligned} f(x^*) &\leq f(x) - \alpha\|\nabla f(x)\|_{\mathbf{W}}^2 + \frac{\alpha^2 A}{2} \|\nabla f(x)\|_{\mathbf{W}}^2 \\ &= f(x) - \left(\alpha - \frac{\alpha^2 A}{2} \right) \|\nabla f(x)\|_{\mathbf{W}}^2 \end{aligned}$$

Since the above inequality holds for any $\alpha \in \mathbb{R}$, it also holds for $\alpha^* = \frac{1}{A}$; this α maximizes the expression $\alpha - \frac{\alpha^2 A}{2}$. It only remains to plug α^* into the above expression and rearrange the inequality. □

B.3 Proof of Lemma 3

Lemma 3. Let $\mathbf{C} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be an unbiased sketch operator. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be differentiable and denote $g(x) = \mathbf{C}\nabla f(x)$. Then

$$\mathbb{E} \left[\|g(x)\|^2 \right] = \|\nabla f(x)\|_{\mathbf{P}^{-1}}^2 \leq 2L_{\mathbf{P}}(f(x) - f^*).$$

Proof.

$$\begin{aligned} \mathbb{E} \left[\|g(x)\|^2 \right] &= \mathbb{E} \left[\|\mathbf{C}\nabla f(x)\|^2 \right] \\ &= \mathbb{E} \left[\nabla f(x)^\top \mathbf{C}^2 \nabla f(x) \right] \\ &= \nabla f(x)^\top \mathbb{E} [\mathbf{C}^2] \nabla f(x) \\ &= \nabla f(x)^\top \mathbf{P}^{-1} \nabla f(x) \\ &= \|\nabla f(x)\|_{\mathbf{P}^{-1}}^2. \end{aligned}$$

The last inequality follows directly from Lemma 2. \square

B.4 Proof of Theorem 2

Theorem 2. *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be \mathbf{L} -smooth, convex, and lower-bounded. Furthermore, suppose that f attains its minimum at x^* . Let $(x^t)_{t=0}^\infty$ be the iterates of the SkGD with unbiased sketch \mathbf{C} and stepsize*

$$0 < \gamma \leq \frac{1}{2L_{\mathbf{P}}}.$$

Denote $\bar{x}^T = \frac{1}{T+1} \sum_{t=0}^T x^t$. Then

$$\mathbb{E} [f(\bar{x}^T) - f(x^*)] \leq \frac{\|x^0 - x^*\|^2}{\gamma(T+1)}.$$

In particular, for $\gamma = 1/(2L_{\mathbf{P}})$ and $T+1 \geq \frac{2L_{\mathbf{P}}\|x^0 - x^*\|^2}{\varepsilon}$, we have that

$$\mathbb{E} [f(\bar{x}^T) - f(x^*)] \leq \varepsilon.$$

Proof. Taking conditional expectations and expanding the brackets yields

$$\begin{aligned} \mathbb{E} [\|x^{t+1} - x^*\|^2 | x^t] &= \|x^t - x^*\|^2 - 2\gamma \mathbb{E} [\langle \mathbf{C}\nabla f(x^t), x^t - x^* \rangle | x^t] + \gamma^2 \mathbb{E} [\|\mathbf{C}\nabla f(x^t)\|^2 | x^t] \\ &\leq \|x^t - x^*\|^2 - 2\gamma \langle \nabla f(x^t), x^t - x^* \rangle + 2\gamma^2 L_{\mathbf{P}} (f(x^t) - f(x^*)) \\ &\leq \|x^t - x^*\|^2 - 2\gamma (f(x^t) - f(x^*)) + \gamma (f(x^t) - f(x^*)) \\ &= \|x^t - x^*\|^2 - \gamma (f(x^t) - f(x^*)), \end{aligned}$$

where the first inequality follows from Lemma 3, and the second one from convexity of f and $0 < \gamma \leq 1/(2L_{\mathbf{P}})$. Rearranging the above inequality, taking the expectation of both sides and using the tower property gives

$$\gamma \mathbb{E} [f(x^t) - f(x^*)] \leq \mathbb{E} [\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2].$$

Thus, by convexity of f , we finally arrive at

$$\begin{aligned} \mathbb{E} [f(\bar{x}^T) - f(x^*)] &\leq \frac{1}{\gamma(T+1)} \sum_{t=0}^T (f(x^t) - f(x^*)) \\ &\leq \frac{1}{\gamma(T+1)} \left(\|x^0 - x^*\|^2 - \mathbb{E} [\|x^{T+1} - x^*\|^2] \right) \\ &\leq \frac{\|x^0 - x^*\|^2}{\gamma(T+1)}. \end{aligned}$$

\square

B.5 Proof of Theorem 3

Theorem 3. *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be \mathbf{L} -smooth and lower bounded by f^* . Let $(x^t)_{t=0}^\infty$ be the iterates of SkGD with unbiased sketch \mathbf{C} and stepsize*

$$0 < \gamma \leq \frac{1}{L_{\mathbf{P}}}.$$

Denote $\hat{x}^T = \operatorname{argmin}_{t=0, \dots, T} \|\nabla f(x^t)\|^2$. Then

$$\mathbb{E} [\|\nabla f(\hat{x}^T)\|^2] \leq \frac{2\delta^0}{\gamma(T+1)},$$

where $\delta^0 = f(x^0) - f^*$. In particular, for $\gamma = 1/L_{\mathbf{P}}$ and $T+1 \geq \frac{2L_{\mathbf{P}}\delta^0}{\varepsilon}$, we have that

$$\mathbb{E} [\|\nabla f(\hat{x}^T)\|^2] \leq \varepsilon.$$

Before carrying out the proof of Theorem 3 we note that in this theorem, one can replace $L_{\mathbf{P}}$ by

$$L_{\mathbf{M}} \stackrel{\text{def}}{=} \lambda_{\max}(\mathbb{E}[\mathbf{CLC}])$$

which is *at most* $L_{\mathbf{P}}$ by Lemma 1. However, the constant $L_{\mathbf{M}}$ is difficult to interpret qualitatively. For the sake of unified presentation of the RPT analysis in all regimes in question, we therefore chose to replace it by $L_{\mathbf{P}}$.

Proof. By \mathbf{L} -smoothness we have

$$\begin{aligned} f(x^{t+1}) &\leq f(x^t) + \langle \nabla f(x^t), x^{t+1} - x^t \rangle + \frac{\|x^{t+1} - x^t\|_{\mathbf{L}}^2}{2} \\ &= f(x^t) - \gamma \langle \nabla f(x^t), \mathbf{C}\nabla f(x^t) \rangle + \frac{\gamma^2}{2} \|\mathbf{C}\nabla f(x^t)\|_{\mathbf{L}}^2 \\ &= f(x^t) - \gamma \langle \nabla f(x^t), \mathbf{C}\nabla f(x^t) \rangle + \frac{\gamma^2}{2} \langle \nabla f(x^t), \mathbf{CLC}\nabla f(x^t) \rangle. \end{aligned}$$

Thus, taking expectation conditional on x^t yields

$$\begin{aligned} \mathbb{E}[f(x^{t+1})|x^t] &\leq f(x^t) - \gamma \|\nabla f(x^t)\|^2 + \frac{\gamma^2}{2} \nabla f(x^t)^\top \mathbb{E}[\mathbf{CLC}] \nabla f(x^t) \\ &\leq f(x^t) - \gamma \|\nabla f(x^t)\|^2 + \frac{\gamma^2}{2} L_{\mathbf{M}} \|\nabla f(x^t)\|^2 \\ &\leq f(x^t) - \gamma \|\nabla f(x^t)\|^2 + \frac{\gamma^2}{2} L_{\mathbf{P}} \|\nabla f(x^t)\|^2, \end{aligned}$$

i.e.

$$\left(\gamma - \frac{\gamma^2 L_{\mathbf{P}}}{2}\right) \|\nabla f(x^t)\|^2 \leq f(x^t) - \mathbb{E}[f(x^{t+1})|x^t].$$

By taking expectations of both sides, using tower property and taking the average over t , we get

$$\begin{aligned} \frac{\gamma}{2} \min_{t=0, \dots, T} \|\nabla f(x^t)\|^2 &\leq \frac{\gamma}{2(T+1)} \sum_{t=0}^T \|\nabla f(x^t)\|^2 \\ &\leq \left(\frac{\gamma}{2} + \frac{\gamma}{2}(1 - \gamma L_{\mathbf{P}})\right) \frac{1}{T+1} \sum_{t=0}^T \|\nabla f(x^t)\|^2 \\ &= \left(\gamma - \frac{\gamma^2 L_{\mathbf{P}}}{2}\right) \frac{1}{T+1} \sum_{t=0}^T \|\nabla f(x^t)\|^2 \\ &\leq \frac{1}{T+1} \mathbb{E}[f(x^0) - f(x^{T+1})] \\ &\leq \frac{1}{T+1} \delta^0. \end{aligned}$$

Rearranging gives

$$\mathbb{E}[\|\nabla f(\hat{x}^T)\|^2] \leq \frac{2\delta^0}{\gamma(T+1)}.$$

□

B.6 Proof of Lemma 4

Lemma 4. Consider an arbitrary positive semidefinite matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ with diagonal subblocks $\mathbf{M}_1, \dots, \mathbf{M}_B$, where each $\mathbf{M}_i \in \mathbb{R}^{d_i \times d_i}$. Put $M_i = \lambda_{\max}(\mathbf{M}_i)$ and $M = \lambda_{\max}(\mathbf{M})$. Then

$$\max_i M_i \leq M \leq \sum_{i=1}^B M_i.$$

Proof. The second inequality is just Lemma 1 from Nesterov (2012). Let us now prove the first inequality. Without loss of generality we may assume that $\max_i L_i = L_1$. Denote e_i to be the i^{th} standard unit vector of \mathbb{R}^d . Let $S = \text{Span}(e_1, \dots, e_{d_1})$. Now,

$$\begin{aligned} L_1 &= \sup_{\substack{x \in \mathbb{R}^{d_1}, \\ \|x\|=1}} x^\top \mathbf{L}_1 x \\ &= \sup_{\substack{x \in S, \\ \|x\|=1}} x^\top \mathbf{L} x \\ &\leq \sup_{\substack{x \in \mathbb{R}^d, \\ \|x\|=1}} x^\top \mathbf{L} x. \end{aligned}$$

This concludes the proof. □

B.7 Proof of Lemma 5

Lemma 5.

$$\sum_{i=1}^B \frac{L_{\sigma_i}}{p_i} \cdot \sum_{i=1}^B p_i c_{\sigma_i} \geq \left(\sum_{i=1}^B \sqrt{L_i c_i} \right)^2.$$

Moreover, this bound is met by taking σ to be such that

$$\sqrt{\frac{L_{\sigma_1}}{c_{\sigma_1}}} \geq \dots \geq \sqrt{\frac{L_{\sigma_B}}{c_{\sigma_B}}},$$

and

$$p_i = \frac{\sqrt{L_{\sigma_i}/c_{\sigma_i}}}{\max_j \sqrt{L_j/c_j}}.$$

Proof. By Cauchy-Schwarz inequality,

$$\begin{aligned} \sum_{i=1}^B \frac{L_{\sigma_i}}{p_i} \cdot \sum_{i=1}^B p_i c_{\sigma_i} &\geq \left(\sum_{i=1}^B \sqrt{L_{\sigma_i}/p_i} \cdot \sqrt{p_i c_{\sigma_i}} \right)^2 \\ &= \left(\sum_{i=1}^B \sqrt{L_i c_i} \right)^2. \end{aligned}$$

Since we are free to choose permutation σ , we may clearly choose it in such a way that

$$\sqrt{L_{\sigma_1}/c_{\sigma_1}} \geq \dots \geq \sqrt{L_{\sigma_B}/c_{\sigma_B}}.$$

Plugging in this choice of permutation σ and probabilities p_1, \dots, p_B verifies that the lower bound from Lemma 5 is attainable. □

C DISCUSSION ABOUT TIGHTNESS OF LEMMA 4

Here we consider the 2-block case argue that the maximum eigenvalue $\lambda_{\max}(\mathbf{M})$ can be arbitrarily close to $\lambda_{\max}(\mathbf{M}_1) + \lambda_{\max}(\mathbf{M}_2)$. Extension of the below result to an arbitrary number of blocks is very straightforward.

Proposition 1. Let $\mathbf{M}_1 \in \mathbb{S}_{++}^{d_1}, \mathbf{M}_2 \in \mathbb{S}_{++}^{d_2}$, and $\mathbf{C} = \sqrt{\lambda_{\max}(\mathbf{M}_1)(\lambda_{\max}(\mathbf{M}_2) - \varepsilon)} \cdot vu^\top$, where $\varepsilon \in (0, \lambda_{\max}(\mathbf{M}_2))$, $u \in \mathbb{R}^{d_1}$ and $v \in \mathbb{R}^{d_2}$ are unit eigenvectors corresponding to maximum eigenvalues of matrices \mathbf{M}_1 and \mathbf{M}_2 , respectively. Then the matrix $\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{C}^\top \\ \mathbf{C} & \mathbf{M}_2 \end{bmatrix}$ is positive definite, and it holds that

$$\lambda_{\max}(\mathbf{M}) \geq \lambda_{\max}(\mathbf{M}_1) + \lambda_{\max}(\mathbf{M}_2) - \varepsilon.$$

Proof. First, let us check the matrix $\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{C}^\top \\ \mathbf{C} & \mathbf{M}_2 \end{bmatrix}$ is positive definite. Since \mathbf{M}_1 is positive definite, it suffices to check that $\mathbf{M}_2 - \mathbf{C}\mathbf{M}_1^{-1}\mathbf{C}^\top$ is positive definite. The second term can be rewritten as

$$\begin{aligned} \mathbf{C}\mathbf{M}_1^{-1}\mathbf{C}^\top &= \lambda_{\max}(\mathbf{M}_1)(\lambda_{\max}(\mathbf{M}_2) - \varepsilon) \underbrace{v u^\top \mathbf{M}_1^{-1} u v^\top}_{= \frac{1}{\lambda_{\max}(\mathbf{M}_1)}} = (\lambda_{\max}(\mathbf{M}_2) - \varepsilon) v v^\top. \end{aligned}$$

Now, through eigenvalue decomposition of \mathbf{M}_2 , we see that eigenvalues of $\mathbf{M}_2 - \mathbf{C}\mathbf{M}_1^{-1}\mathbf{C}^\top$ are the same as eigenvalues of \mathbf{M}_2 except for the maximum one, which is replaced by $\varepsilon > 0$. Thus, $\mathbf{M}_2 - \mathbf{C}\mathbf{M}_1^{-1}\mathbf{C}^\top$ is positive definite.

Let $\mathbf{M}_1 = \mathbf{\Lambda}_1 \mathbf{\Sigma}_1 \mathbf{\Lambda}_1^\top$ and $\mathbf{M}_2 = \mathbf{\Lambda}_2 \mathbf{\Sigma}_2 \mathbf{\Lambda}_2^\top$ be eigenvalue decompositions of the matrices, and without the loss of generality assume that the eigenvalues are sorted in a descending order. We also set $a \stackrel{\text{def}}{=} \sqrt{\lambda_{\max}(\mathbf{M}_1)(\lambda_{\max}(\mathbf{M}_2) - \varepsilon)}$. Then

$$\mathbf{M}' \stackrel{\text{def}}{=} \text{Diag}(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2)^\top \mathbf{M} \text{Diag}(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2) = \begin{bmatrix} \mathbf{\Sigma}_A & \text{Diag}(a, 0, \dots, 0)^\top \\ \text{Diag}(a, 0, \dots, 0) & \mathbf{\Sigma}_2 \end{bmatrix}$$

has the same set of eigenvalues as \mathbf{M} . We now have

$$\begin{aligned} \lambda_{\max}(\mathbf{M}') &= \max_{\|u'\|^2 + \|v'\|^2 = 1} (u', v')^\top \mathbf{M}' (u', v') \\ &= \max_{\|u'\|^2 + \|v'\|^2 = 1} u'^\top \mathbf{\Sigma}_1 u' + v'^\top \mathbf{\Sigma}_2 v' + 2u'^\top \text{Diag}(a, 0, \dots, 0)^\top v' \\ &\geq \max_{x^2 + y^2 = 1} x^2 \lambda_{\max}(\mathbf{M}_1) + y^2 \lambda_{\max}(\mathbf{M}_2) + 2xy \sqrt{\lambda_{\max}(\mathbf{M}_1)(\lambda_{\max}(\mathbf{M}_2) - \varepsilon)} \\ &\geq \max_{x^2 + y^2 = 1} x^2 \lambda_{\max}(\mathbf{M}_1) + y^2 (\lambda_{\max}(\mathbf{M}_2) - \varepsilon) + 2xy \sqrt{\lambda_{\max}(\mathbf{M}_1)(\lambda_{\max}(\mathbf{M}_2) - \varepsilon)} \\ &= \max_{x^2 + y^2 = 1} \left(x \sqrt{\lambda_{\max}(\mathbf{M}_1)} + y \sqrt{\lambda_{\max}(\mathbf{M}_2) - \varepsilon} \right)^2 \\ &\geq \left(x \sqrt{\lambda_{\max}(\mathbf{M}_1)} + y \sqrt{\lambda_{\max}(\mathbf{M}_2) - \varepsilon} \right)^2 \Big|_{x = \sqrt{\frac{\lambda_{\max}(\mathbf{M}_1)}{\lambda_{\max}(\mathbf{M}_1) + \lambda_{\max}(\mathbf{M}_2) - \varepsilon}}, y = \sqrt{\frac{\lambda_{\max}(\mathbf{M}_2) - \varepsilon}{\lambda_{\max}(\mathbf{M}_1) + \lambda_{\max}(\mathbf{M}_2) - \varepsilon}}} \\ &= \lambda_{\max}(\mathbf{M}_1) + \lambda_{\max}(\mathbf{M}_2) - \varepsilon. \end{aligned}$$

It remains to note that $\lambda_{\max}(\mathbf{M}') = \lambda_{\max}(\mathbf{M})$, since the sets of eigenvalues coincide. \square

D DISCUSSION ABOUT THE UNIFIED ANALYSIS IN NONCONVEX AND CONVEX SETTINGS

D.1 Nonconvex setting

Recent theoretical breakthrough of Khaled and Richtárik (2023) offers generic result on the convergence of SGD under very mild assumptions. However, in this section we argue that the rates promised by this result is suboptimal for the setting of our paper, which is why we derived a separate analysis of RCD in this regime.

Assumption 6 (Assumption 2, Khaled and Richtárik (2023)). Let $g(x)$ be an unbiased estimator of $\nabla f(x)$ for all $x \in \mathbb{R}^d$. We say that g satisfies an *ABC assumption* if

$$\mathbb{E} \left[\|g(x)\|^2 \right] \leq 2A (f(x) - f^{\text{inf}}) + B \|\nabla f(x)\|^2 + C.$$

for all $x \in \mathbb{R}^d$.

Theorem 4. *Suppose that f is lower-bounded and L -smooth. Let g be an unbiased gradient estimator satisfying Assumption 6 with $B = C = 0$. Let $0 < \gamma \leq \frac{1}{\sqrt{LAK}}$. Denote $f(x^0) - f^{\text{inf}}$ by δ_0 . Then, the iterates x^t of SGD using the estimator g and stepsize γ satisfy*

$$\min_{0 \leq t \leq T-1} \mathbb{E} \left[\|\nabla f(x^t)\|^2 \right] \leq \frac{2(1 + L\gamma^2 A)^T \delta_0}{\gamma T}.$$

In particular, for

$$T \geq \frac{144\delta_0^2 LA}{\varepsilon^2}$$

we have $\min_{0 \leq t \leq T-1} \mathbb{E} \left[\|\nabla f(x^t)\|^2 \right] \leq \varepsilon$.

Notice that for nonconvex smooth problems, by Lemma 3, Assumption 6 is satisfied for $A = L_{\mathbf{P}}$ and $B = C = 0$. Moreover,

$$\mathbb{E} \left[\|g(x)\|^2 \right] = \|\nabla f(x)\|_{\mathbf{P}^{-1}}^2 \leq \lambda_{\max}(\mathbf{P}^{-1}) \|\nabla f(x)\|^2,$$

so Assumption 6 is also satisfied for $B = \lambda_{\max}(\mathbf{P}^{-1})$ and $A = C = 0$. There seem to be no other ways in which the RPT estimator can satisfy Assumption 6.

Plugging in both of these cases into Theorem 4 yields suboptimal convergence rates.

Indeed, for $A = L_{\mathbf{P}}$ and $B = C = 0$ the rate of RPT guaranteed by Theorem 4 is $O\left(\frac{1}{\sqrt{T}}\right)$, which is suboptimal (we have proved that it is in fact $O(1/T)$).

For $B = \lambda_{\max}(\mathbf{P}^{-1})$ and $A = C = 0$ the rate is $O(1/T)$ but it is proportional to

$$L\lambda_{\max}(\mathbf{P}^{-1}) = \lambda_{\max}(\mathbf{L})\lambda_{\max}(\mathbf{P}^{-1}) \geq L_{\mathbf{P}},$$

by Lemma 4.

We manage to obtain faster rate by exploiting \mathbf{L} -smoothness and the fact that $\mathbb{E} \left[\|g(x)\|_{\mathbf{L}}^2 \right]$ can be expressed as a suitable matrix norm.

D.2 Convex setting

Applying generic result of Khaled et al. (2023) yields somewhat good convergence guarantee ($O(1/T)$ rate is recovered); however, it still suffers from an unnecessary dependency on function suboptimality.

Theorem 5 (Adapted from Khaled et al. (2023)). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be convex, differentiable and lower bounded with global minimizer x^* . Suppose that an unbiased gradient estimator g satisfies*

$$\mathbb{E} \left[\|g(x)\|^2 \right] \leq 2A(f(x) - f(x^*)).$$

Choose stepsize

$$\gamma \leq \min \left\{ \frac{1}{4A}, \frac{1}{2L} \right\}.$$

Then the SGD procedure with gradient estimator g and stepsize γ satisfies

$$\mathbb{E} [f(\bar{x}^T) - f(x^*)] \leq \frac{2\gamma\delta^0 + \|x^0 - x^*\|^2}{\gamma T},$$

where $\bar{x}^T \stackrel{\text{def}}{=} \frac{1}{T} \sum_{i=0}^{T-1} x^i$ and $\delta^0 \stackrel{\text{def}}{=} f(x^0) - f(x^*)$.

As discussed previously, the assumption from Theorem 5 is satisfied for $A = L_{\mathbf{P}} \geq L$. Thus, the optimal theoretical stepsize is $\gamma = 1/(4L_{\mathbf{P}})$. Plugging in this stepsize yields the rate

$$O\left(\frac{\delta^0 + L_{\mathbf{P}} \|x^0 - x^*\|^2}{T}\right),$$

whereas we obtained the rate of

$$O\left(\frac{L_{\mathbf{P}} \|x^0 - x^*\|^2}{T}\right).$$