
DROP-MUON: UPDATE LESS, CONVERGE FASTER

Kaja Gruntkowska

KAUST*

Center of Excellence for Generative AI
Thuwal, Saudi Arabia

Yassine Maziane

KAUST*

Center of Excellence for Generative AI
Thuwal, Saudi Arabia

Zheng Qu

Shenzhen University

School of Mathematical Sciences
Shenzhen, China

Peter Richtárik

KAUST*

Center of Excellence for Generative AI
Thuwal, Saudi Arabia

ABSTRACT

Conventional wisdom in deep learning optimization dictates updating all layers at every step—a principle followed by all recent state-of-the-art optimizers such as Muon. In this work, we challenge this assumption, showing that full-network updates can be fundamentally suboptimal, both in theory and in practice. We introduce a non-Euclidean Randomized Progressive Training method—Drop-Muon—a simple yet powerful framework that updates only a subset of layers per step according to a randomized schedule, combining the efficiency of progressive training with layer-specific non-Euclidean updates for top-tier performance. We provide rigorous convergence guarantees under both layer-wise smoothness and layer-wise (L^0, L^1) -smoothness, covering deterministic and stochastic gradient settings, marking the first such results for progressive training in the stochastic and non-smooth regime. Our cost analysis further reveals that full-network updates are not optimal unless a very specific relationship between layer smoothness constants holds. Through controlled CNN experiments, we empirically demonstrate that Drop-Muon consistently outperforms full-network Muon, achieving the same accuracy up to $1.4\times$ faster in wall-clock time. Together, our results suggest a shift in how large-scale models can be efficiently trained, challenging the status quo and offering a highly efficient, theoretically grounded alternative to full-network updates.

1 Introduction

Since their debut, Adam and related methods [Kingma and Ba, 2015, Loshchilov and Hutter, 2019] have dominated deep learning optimization. Yet, the field is now at an inflection point. Recent advances highlight a new generation of algorithms designed to better capture the geometry of modern models, with Muon [Jordan et al., 2024] and its successors—Scion [Pethick et al., 2025] and Gluon [Riabinin et al., 2025]—emerging as promising alternatives. Fueled by state-of-the-art performance in large language models (LLMs) training [Liu et al., 2025, Shah et al., 2025, Thérien et al., 2025, Moonshot AI, 2025, Wen et al., 2025] and emerging theoretical developments [Pethick et al., 2025, Kovalev, 2025, Riabinin et al., 2025], these methods are on track to disrupt entrenched practices.

Central to their design are the layer-specific *linear minimization oracles* (LMOs) over non-Euclidean norm balls, enabling better alignment with the highly anisotropic loss landscapes of neural networks. Concretely, let $X = [X_1, \dots, X_b]$ denote the parameters of a b -layer model, with X_i indexing the parameters of layer $i \in [b] := \{1, \dots, b\}$. Each of the aforementioned optimizers can be viewed as an instance of the general update rule

$$X_i^{k+1} = X_i^k + \text{LMO}_{\mathcal{B}_i(0, t_i^k)}(M_i^k), \quad i \in \{1, \dots, b\} \quad (1)$$

(see Section A.1). Here, $\mathcal{B}_i(X_i, t_i) := \{Z_i \in \mathcal{X}_i : \|X_i - Z_i\|_{(i)} \leq t_i\}$ is a ball of radius t_i centered at X_i in the vector space $\mathcal{X}_i$, where $\|\cdot\|_{(i)}$ is a norm chosen for the i th layer, M_i^k is a momentum term, and the linear minimization

*King Abdullah University of Science and Technology

Algorithm 1 Drop-Muon

```

1: Input: initial iterate  $X^0 = [X_1^0, \dots, X_b^0] \in \mathcal{X}$ ; momentum  $M^0 = [M_1^0, \dots, M_b^0] \in \mathcal{X}$ ; stepsizes  $\gamma_i^k > 0$ ;
   momentum parameters  $\beta^k \in [0, 1)$ 
2: for  $k = 0, \dots, K - 1$  do
3:   Sample  $\xi^k \sim \mathcal{P}$  and the set of active layers  $S^k \sim \mathcal{D}$ 
4:   for  $i \notin S^k$  do ▷ Freeze layers not selected as active
5:      $M_i^k = M_i^{k-1}$ 
6:      $X_i^{k+1} = X_i^k$ 
7:   end for
8:   for  $i \in S^k$  do ▷ Update active layers
9:     Update momentum  $M_i^k = (1 - \beta_i)M_i^{k-1} + \beta_i \nabla_i f(X^k; \xi^k)$ 
10:    Update parameters via
        
$$X_i^{k+1} = \text{LMO}_{\mathcal{B}_i(X_i^k, t_i^k)}(M_i^k) = X_i^k - \gamma_i^k (M_i^k)^\# \quad (3)$$

11:   end for
12: end for
    
```

oracle is defined as $\text{LMO}_{\mathcal{B}_i(X_i, t_i)}(M_i) := \arg \min_{Z_i \in \mathcal{B}_i(X_i, t_i)} \langle M_i, Z_i \rangle$. Different choices of $\|\cdot\|_{(i)}$ produce different algorithms (for example, Muon uses the spectral norm for hidden layers). Crucially, however, all these updates share a common characteristic: **all layers are updated at every iteration**. In this work, we question this design choice. Our central hypothesis is simple yet fundamental:

*Updating the entire network at every step may **not** be optimal.*

In the remainder of this paper, we demonstrate that the default practice of full-network updates is indeed not universally the best choice, from both *theoretical* and *practical* points of view, calling into question a core principle of standard training protocols.

1.1 Background

Let us begin by formalizing the setup. We consider the optimization problem

$$\min_{X \in \mathcal{X}} \{f(X) := \mathbb{E}_{\xi \sim \mathcal{P}} [f(X; \xi)]\}, \quad (2)$$

where $X \in \mathcal{X}$ represents the collection of trainable parameters of a neural network. Specifically, X is composed of block variables $X_i \in \mathcal{X}_i := \mathbb{R}^{m_i \times n_i}$ corresponding to layer $i \in [b]$; we write $X = [X_1, \dots, X_b]$. In this context, $\mathcal{X}$ is the d -dimensional product space

$$\mathcal{X} := \bigotimes_{i=1}^b \mathcal{X}_i \equiv \mathcal{X}_1 \otimes \dots \otimes \mathcal{X}_b,$$

where $d := \sum_{i=1}^b m_i n_i$. Each function $f(\cdot; \xi) : \mathcal{X} \rightarrow \mathbb{R}$ is continuously differentiable, potentially nonconvex and non-smooth, and represents the loss of the model evaluated at a data point ξ sampled from the probability distribution $\mathcal{P}$. We denote by $\nabla_i f(X) \in \mathcal{X}_i$ the gradient component corresponding to the i th layer, so that $\nabla f(X) = [\nabla_1 f(X), \dots, \nabla_b f(X)] \in \mathcal{X}$. Each space $\mathcal{X}_i$ is equipped with the trace inner product, defined as $\langle X_i, Y_i \rangle_{(i)} := \text{tr}(X_i^\top Y_i)$ for $X_i, Y_i \in \mathcal{X}_i$, which induces the standard Euclidean norm, denoted by $\|\cdot\|_2$. In addition, each space is endowed with an arbitrary norm $\|\cdot\|_{(i)}$ (which need not be induced by this inner product). We let $\|\cdot\|_{(i)*}$ be the dual norm associated with $\|\cdot\|_{(i)}$ (i.e., $\|X_i\|_{(i)*} := \sup_{\|Z_i\|_{(i)} \leq 1} \langle X_i, Z_i \rangle_{(i)}$ for any $X_i \in \mathcal{X}_i$).

2 The Algorithm

Before summarizing our main contributions (see the end of Section 3), we dive directly into presenting our method. Motivated by the considerations in Section 1, we propose Drop-Muon (Algorithm 1)—a non-Euclidean layer-wise optimizer for deep learning based on the idea of sub-network training. At each iteration k , instead of updating the entire network as in standard Muon, Drop-Muon samples a random subset $S^k \subseteq [b]$ of layers according to a user-defined distribution $\mathcal{D}$ and updates only the parameters of layers in S^k , keeping all other layers frozen. As

Algorithm 2 Drop-Muon (deterministic gradient variant)

- 1: **Input:** initial iterate $X^0 = [X_1^0, \dots, X_b^0] \in \mathcal{X}$; stepsizes $\gamma_i^k > 0, i \in [b], k \geq 0$
- 2: **for** $k = 0, \dots, K - 1$ **do**
- 3: Sample $S^k \sim \mathcal{D}$
- 4: Freeze the layers $\{X_i\}_{i \notin S^k}$ ($X_i^{k+1} = X_i^k$ for $i \notin S^k$)
- 5: Update the layers $\{X_i\}_{i \in S^k}$ via

$$X_i^{k+1} = X_i^k - \gamma_i^k (\nabla_i f(X^k))^\sharp \quad (5)$$

6: **end for**

the reader may have noticed, the main LMO update step (3) admits two equivalent formulations. The alternative representation of (1) uses the *sharp operator* [Nesterov, 2012, Kelner et al., 2014], defined for any $M \in \mathcal{X}$ by $M^\sharp := \arg \max_{X \in \mathcal{X}} \{\langle M, X \rangle - \frac{1}{2} \|X\|^2\}$. It is well known that $M^\sharp$ relates to the LMO via $M^\sharp = -\|M\|_\star \text{LMO}_{B(0,1)}(M)$, and hence

$$X_i^{k+1} = X_i^k + t_i^k \text{LMO}_{B_i(0,1)}(M_i^k) = X_i^k - \frac{t_i^k}{\|M_i^k\|_{(i)\star}} (M_i^k)^\sharp, \quad (4)$$

which corresponds to a layer-wise normalized *steepest descent* step with stepsize $\gamma_i^k := t_i^k / \|M_i^k\|_\star$. When $\|\cdot\|_{(i)} = \|\cdot\|_2$ is the standard Euclidean norm, the sharp operator reduces to the identity mapping, so that $M^\sharp = M$, and the update coincides with Stochastic Gradient Descent with momentum (SGDM) [Cutkosky and Mehta, 2020], though here performed layer-wise. These equivalent formulations will be repeatedly invoked in the proofs of the results from Sections 4.1 and 4.2.

3 Cost Model

Our key theoretical contribution is that strategically skipping some layer updates may lead to performance gains. To isolate the core phenomenon, we first study a simplified variant of the method without stochasticity in the gradients and momentum, described in Algorithm 2. This deterministic version follows the same fundamental principles as Algorithm 1, with the difference that momentum terms M_i^k are replaced by the components of the full gradient $\nabla_i f(X^k)$. When $\|\cdot\|_{(i)} = \|\cdot\|_2$ is the Euclidean norm, the update (5) coincides with that of layer-wise Gradient Descent (GD).

How expensive is one step of Algorithm 2? The answer is governed by the sampling distribution $\mathcal{D}$. Consider iteration k and denote $s^k := \min S^k$ (the smallest index of an active layer). The operations performed by the algorithm can be summarized as follows:

- (i) **Backward pass:** Backpropagate gradients through layers $[X_{s^k}^k, \dots, X_b^k]$. Since layers $1, \dots, s^k - 1$ are frozen, no gradients are computed for them, effectively truncating the gradient flow at the first active layer.
- (ii) **Forward pass:** To evaluate the loss, activations must in principle be propagated through all layers $1, \dots, b$. However, since only layers $[X_{s^k}^k, \dots, X_b^k]$ are updated at iteration k , the activations up to layer $s^k - 1$ may be cached and reused in the next step.
- (iii) **Gradient transformation:** Given the gradients $\{\nabla_i f(X^k)\}_{i \in S^k}$, compute the corresponding sharp operators $\{(\nabla_i f(X^k))^\sharp\}_{i \in S^k}$ (or, equivalently, the LMOs; see (4)).
- (iv) **Parameter updates:** Update the parameters of layers $\{X_i^k\}_{i \in S^k}$ using their computed (transformed) gradients, while keeping the frozen layers unchanged.

To model the total computational effort of the optimization procedure, we associate a *cost* with each step (measured, for example, in FLOPs or wall-clock time). Let $c_{\text{ov}} \geq 0$ denote the fixed per-iteration overhead (e.g., data loading). As noted above, backpropagation must be performed from the last layer b down to layer s^k , while forward-pass activations up to layer $s^k - 1$ can be cached and reused in subsequent iterations. Hence, the costs of steps (i) and (ii) can be aggregated into a single per-layer constant $c_i > 0$ for each $i \in [b]$. In the non-Euclidean setting (where $M^\sharp \neq M$), an additional cost arises from computing sharp operators. We denote by $c_i^\sharp \geq 0$ the combined cost of evaluating this operator and performing the corresponding parameter update for layer i (steps (iii) and (iv)). Under this model, the total

compute cost of iteration k is

$$\text{cost}(S^k) := c_{\text{ov}} + \sum_{i=s^k}^b c_i + \sum_{i \in S^k} c_i^{\#}, \quad (6)$$

and, consequently, for a fixed target accuracy $\varepsilon > 0$, the expected cost of the entire optimization procedure can be expressed as

$$\text{cost}_{\varepsilon}(\mathcal{D}) := K \times \mathbb{E} \left[\text{cost}(\hat{S}) \right], \quad (7)$$

where we write $\hat{S} \sim \mathcal{D}$ to denote a random variable with the same distribution as that of the samplings (since $\{S^k\}_{k \geq 0}$ are i.i.d.), and K is the number of iterations to reach convergence (interpreted in the nonconvex case as reaching an ε -stationary point in expectation). In the remainder of this paper, we compute K under two smoothness regimes: layer-wise smoothness (Assumption 4.2) and layer-wise (L^0, L^1) -smoothness (Assumption 4.3). We then evaluate various layer-update strategies and compute their expected costs as in (7). These results converge to a single conclusion:

Full-network updates are *not* optimal unless a very specific relationship between layer smoothness constants holds.

We provide a rigorous statement and justification for this claim in the sections that follow. The main takeaway, however, is simple: as this condition is highly unlikely to be realized in practice, updating only a subset of layers at each iteration should be seen as more efficient than the default strategy of updating all parameters at each iteration.

Our **contributions** can be summarized as follows:

1. **Challenging full network updates.** We provide, to our knowledge, the first systematic investigation of the practice of updating *all layers* of a network at every iteration. We argue—and rigorously demonstrate both in theory and in practice—that this design choice is *generally suboptimal*.
2. **General framework for sub-network optimization.** We introduce Drop-Muon (Algorithm 1 and its deterministic gradient counterpart Algorithm 2), a principled layer-wise optimization framework with randomized layer subsampling. Drop-Muon strictly generalizes LMO-type methods (including Muon [Jordan et al., 2024], Scion [Pethick et al., 2025], and Gluon [Riabini et al., 2025]) by allowing random subsets of layers to be updated per step, with full-network training as a special case. Drop-Muon supports virtually any layer sampling scheme (Section C). In the main part of this paper, we focus on Randomized Progressive Training (RPT) [Szlendak et al., 2024], a natural strategy aligned with backpropagation mechanics that avoids redundant gradient computations and reduces compute cost while maintaining strong convergence guarantees.
3. **Tight iteration complexity guarantees under novel smoothness regimes.** We establish convergence guarantees for Drop-Muon under two regimes: *layer-wise smoothness* (Theorem 4.1) and *layer-wise (L^0, L^1) -smoothness* (Theorems 4.2 and 4.4). Our rates recover the state of the art for SGD- and Muon-type methods, and, to our knowledge, provide the first convergence guarantees for progressive training-style methods in the non-smooth setting.
4. **Theoretical compute-optimality results.** To isolate the key phenomena, we first consider a deterministic gradient variant of Drop-Muon (Algorithm 2). Using a simple yet expressive cost model (Equation (6)) accounting for per-layer forward/backward passes, gradient transformations, and parameter updates, we prove that full-network updates are *not* optimal unless a very specific condition on layer smoothness constants holds (Theorems 4.3 and E.2), which is unlikely in practice. This formally justifies selective layer updates as the compute-optimal default.
5. **Empirical validation.** Controlled CNN experiments on MNIST, Fashion-MNIST, and CIFAR-10 show that Drop-Muon consistently outperforms standard full-network Muon, achieving the same accuracy up to $1.4\times$ faster in wall-clock time.

4 Randomized Progressive Training

The general framework in Algorithm 2 allows virtually any sampling strategy. However, due to the mechanics of backpropagation, it is most natural to update all layers from the last one down to some sampled minimal index. Specifically, if the smallest sampled index at iteration k is s^k , then computing the gradient $\nabla_{s^k} f(X^k)$ requires backpropagating from the last layer b up to layer s^k , which automatically produces all gradient components $[\nabla_{s^k} f(X^k), \dots, \nabla_b f(X^k)]$.

Formally, we can define the sampling distribution $\mathcal{D}$ as follows: at each iteration k , sample $s^k \in [b]$ with probabilities $p_i := \mathbb{P}(s^k = i)$, where $\sum_{i=1}^b p_i = 1$ and $p_1 > 0$, and set $S^k = \{s^k, \dots, b\}$. Algorithms 1 and 2 then update the layers

$[X_{s^k}^k, \dots, X_b^k]$, while $[X_1^k, \dots, X_{s^k-1}^k]$ remain frozen at their previous values (with the convention that $[X_1^k, X_0^k] = \emptyset$ when $s^k = 1$). We refer to this sampling scheme as *Randomized Progressive Training* (RPT), or Drop-training for short (see Section 5).

4.1 Iteration Complexity – Deterministic Gradient Setting

We now analyze the iteration complexity of Algorithm 2 under two smoothness regimes; the proofs are deferred to the Appendix. Throughout, we make the standard assumption that the objective function is lower-bounded.

Assumption 4.1. There exist $f^* \in \mathbb{R}$ such that $f(X) \geq f^*$ for all $X \in \mathcal{X}$.

This ensures the existence of an approximately stationary point for any desired level of accuracy.

Smooth case. We first establish convergence under the *layer-wise smoothness* assumption.

Assumption 4.2 (supp($\mathcal{D}$)-layer-wise smoothness). The function $f : \mathcal{X} \mapsto \mathbb{R}$ is supp($\mathcal{D}$)-layer-wise L^0 -smooth with constants $L^0 := \{(L_{1,S}^0, \dots, L_{b,S}^0)\}_{S \in \text{supp}(\mathcal{D})}$, $(L_{1,S}^0, \dots, L_{b,S}^0) \in \mathbb{R}_+^b$, i.e., for any $S \in \text{supp}(\mathcal{D})$,

$$f(X + \Gamma) - f(X) - \langle \nabla f(X), \Gamma \rangle \leq \sum_{i \in S} \frac{L_{i,S}^0}{2} \|\Gamma_i\|_{(i)}^2.$$

for all $X = [X_1, \dots, X_b] \in \mathcal{X}$ and $\Gamma = [\Gamma_1, \dots, \Gamma_b] \in \mathcal{X}$ such that $\Gamma_i = 0$ for all $i \notin S$.

We take L^0 to be the smallest collection of constants satisfying the above. Throughout, we use $\text{supp}(\mathcal{D})$ to denote the subsets of $[b]$ assigned positive probability mass by $\mathcal{D}$. In the progressive training setting, these supported sets take the form $\text{supp}(\mathcal{D}) = \{\{j, \dots, b\}, j \in [b]\}$. By definition, $L_{i,S}^0 = 0$ whenever $i \notin S$. Moreover, if $S_1, S_2 \in \text{supp}(\mathcal{D})$ are such that $S_1 \subseteq S_2$, then $L_{i,S_1}^0 \leq L_{i,S_2}^0$ (see Lemma B.1).

The assumption is inspired by the coordinate descent (CD) literature [Wright, 2015], reducing to the standard block-wise Lipschitz continuity of the gradient [Richtárik and Takáč, 2014a, Fercoq and Richtárik, 2015, Qu and Richtárik, 2016] in the special case when $\text{supp}(\mathcal{D}) = \{\{1\}, \{2\}, \dots, \{b\}\}$. Assumption 4.2 captures the intuition that each subset of layers can have its own effective smoothness constant, allowing tighter bounds on the local curvature of f and better reflecting the structure of the model. Importantly, it is *not* more restrictive than standard smoothness—rather, it offers a richer parametric description by assigning separate constants to different layer subsets, allowing a more precise analysis without shrinking the function class.

With the assumptions in place, we are now ready to state the first formal convergence result.

Theorem 4.1. Let Assumptions 4.1 and 4.2 hold, and let $\{X^k\}_{k=0}^{K-1}$ be the iterates of Algorithm 2 run with stepsizes $\gamma_i^k = 1/L_{i,S^k}^0$. Then

$$\frac{1}{K} \sum_{k=0}^{K-1} \sum_{i=1}^b \frac{w_i}{\frac{1}{b} \sum_{j=1}^b w_j} \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star}^2 \right] \leq \frac{f(X^0) - f^*}{K \left(\frac{1}{b} \sum_{j=1}^b w_j \right)},$$

where $w_i := \sum_{s=1}^i \frac{p_s}{2L_{i,\{s, \dots, b\}}^0}$.

Theorem 4.1 establishes an $\mathcal{O}(K^{-1})$ convergence rate for a weighted sum of squared gradient component norms, matching the theoretical rates previously established for Muon-type methods under classical smoothness [Li and Hong, 2025, Pethick et al., 2025, Kovalev, 2025]. For the result to be meaningful, every gradient component must contribute to the weighted average. In other words, the sampling distribution must satisfy $w_i > 0$ for all $i \in [b]$. This is a natural requirement, equivalent to ensuring that $p_1 > 0$, i.e., that all layers are updated with nonzero probability. Obviously, if this was not the case, the first layer would be completely ignored, making convergence impossible.

Generalized smooth case. Layer-wise optimizers considered here are designed for deep learning, where the classical smoothness assumption is often violated [Zhang et al., 2020, Riabinin et al., 2025]. Consequently, the layer-wise smoothness model in Assumption 4.2 may not accurately capture the local geometry of the loss. To address this, we adopt a more expressive framework building upon (L^0, L^1) -smoothness [Zhang et al., 2020, Chen et al., 2023b].

Assumption 4.3 below generalizes Assumption 4.2 by letting the local curvature of each layer depend not only on fixed constants $L_{i,S}^0$, but also on the magnitude of the layer's gradient via additional terms $L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}$.

Assumption 4.3 (supp($\mathcal{D}$)-layer-wise (L^0, L^1) -smoothness). The function $f : \mathcal{X} \mapsto \mathbb{R}$ is supp($\mathcal{D}$)-layer-wise (L^0, L^1) -smooth with constants $L^\alpha := \{(L_{1,S}^\alpha, \dots, L_{b,S}^\alpha)\}_{S \in \text{supp}(\mathcal{D})}$, $(L_{1,S}^\alpha, \dots, L_{b,S}^\alpha) \in \mathbb{R}_+^b$, $\alpha \in \{0, 1\}$, i.e., for any $S \in \text{supp}(\mathcal{D})$,

$$f(X + \Gamma) - f(X) - \langle \nabla f(X), \Gamma \rangle \leq \sum_{i \in S} \frac{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}}{2} \|\Gamma_i\|_{(i)}^2,$$

for all $X = [X_1, \dots, X_b] \in \mathcal{X}$ and $\Gamma = [\Gamma_1, \dots, \Gamma_b] \in \mathcal{X}$ such that $\Gamma_i = 0$ for all $i \notin S$.

As with Assumption 4.2, assigning separate constants to each subset of layers S allows for tighter, subset-specific bounds, reflecting the interactions among layers.

Theorem 4.2. Let Assumptions 4.1 and 4.3 hold, fix $\varepsilon > 0$, and let $\{X^k\}_{k=0}^{K-1}$ be the iterates of Algorithm 2 run with stepsizes $\gamma_i^k = (L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star})^{-1}$. Then, to guarantee that

$$\min_{k=0, \dots, K-1} \sum_{i=1}^b \left[\frac{w_i}{\frac{1}{b} \sum_{l=1}^b w_l} \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star} \right] \right] \leq \varepsilon,$$

it suffices to run the algorithm for

$$K = \left\lceil \frac{2\delta^0 \sum_{i=1}^b \frac{(\sum_{s=1}^i p_s)^2 (\sum_{s=1}^i p_s L_{i,\{s, \dots, b\}}^0)}{(\sum_{s=1}^i p_s L_{i,\{s, \dots, b\}}^1)^2}}{\varepsilon^2 \left(\frac{1}{b} \sum_{l=1}^b w_l \right)^2} + \frac{2\delta^0}{\varepsilon \left(\frac{1}{b} \sum_{l=1}^b w_l \right)} \right\rceil$$

iterations, where $\delta^0 := f(X^0) - f^\star$ and $w_i := \frac{(\sum_{s=1}^i p_s)^2}{\sum_{s=1}^i p_s L_{i,\{s, \dots, b\}}^1}$.

Similar to Theorem 4.1, Theorem 4.2 guarantees an $\mathcal{O}(K^{-1/2})$ convergence rate for a weighted sum of gradient component norms.² Again, the sampling distribution must ensure that $w_i > 0$ for all $i \in [b]$, which amounts to requiring that $p_1 > 0$. In the extreme case when $(p_1, p_2, \dots, p_b) = (1, 0, \dots, 0)$, corresponding to full-network training, the weights simplify to $w_i = 1/L_{i,[b]}^1$, exactly recovering the convergence rate of deterministic Gluon [Riabini et al., 2025, Theorem 1], demonstrating the tightness of our guarantees. Importantly, the stepsizes naturally scale inversely with the layer-specific smoothness constants and gradient magnitudes. This automatic adaptation to local geometry prevents overshooting, ensures stable convergence, and allows more aggressive updates when the (L^0, L^1) constants and gradient norms are small.

4.1.1 Cost Optimization

Let us now make clear why performing full-network updates in all iterations is *not* in general an optimal strategy. We first consider the layer-wise smooth case. According to Theorem 4.1, under Assumption 4.2, Algorithm 2 guarantees that $\frac{1}{K} \sum_{k=0}^{K-1} \sum_{i=1}^b \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star}^2 \right] \leq \varepsilon$ after

$$K = \left\lceil \frac{f(X^0) - f^\star}{\varepsilon \times \min_{i \in [b]} \left[\sum_{s=1}^i \frac{p_s}{2L_{i,\{s, \dots, b\}}^0} \right]} \right\rceil$$

²Theorem 4.2 bounds the gradient norms directly, while Theorem 4.1 bounds their squares; this difference naturally arises from the distinct smoothness models used in each analysis.

iterations. Since in this case $\mathbb{E} [\text{cost}(\hat{S})] = c_{\text{ov}} + \sum_{j=1}^b (c_j + c_j^\#) \sum_{i=1}^j p_i$, the expected total cost of the optimization procedure satisfies

$$\text{cost}_\varepsilon(\mathcal{D}) \stackrel{(7)}{=} K \times \mathbb{E} [\text{cost}(\hat{S})] \propto \frac{c_{\text{ov}} + \sum_{i=1}^b (c_i + c_i^\#) \sum_{s=1}^i p_s}{\min_{i \in [b]} \left[\sum_{s=1}^i \frac{p_s}{2L_{i,\{s,\dots,b\}}^0} \right]} \quad (8)$$

(see Section E.1.1 for the details of the derivation). Finding the sampling distribution that minimizes $\text{cost}_\varepsilon(\mathcal{D})$ is equivalent to optimizing over $\{p_i\}_{i \in [b]}$. Letting $(p_1, p_2, \dots, p_b) = (1, 0, \dots, 0)$ recovers full-network training, which serves as a natural baseline. Yet, can we do better? The following theorem shows that this configuration is optimal under very specific conditions only.

Theorem 4.3. The cost (8) is minimized by $(p_1, p_2, \dots, p_b) = (1, 0, \dots, 0)$ **if and only if**

$$L_{1,\{1,\dots,b\}} = \max_{i \in [b]} L_{i,\{1,\dots,b\}}.$$

Note that the condition in Theorem 4.3 is entirely independent of the cost parameters! In fact, one can derive a *recursive construction of the optimal probabilities* that depends solely on the smoothness constants, from which Theorem 4.3 follows as a simple corollary (see Remark E.1).

The layer-wise (L^0, L^1) -smooth case (Theorem E.2) leads to the same conclusion: **full-network updates are optimal only if the first layer is associated with the largest smoothness constant**—a restrictive and rarely observed condition, as confirmed by experiments on NanoGPT [Riabinin et al., 2025]. While broader validation is needed, there is no reason to expect this phenomenon to hold broadly. Overall, from the theoretical standpoint, *the prevalent practice of always updating all layers is fundamentally flawed*.

4.2 Iteration Complexity – Stochastic Gradient Setting

We now turn to the convergence analysis of the practical variant of Drop-Muon with stochastic gradients and momentum (Algorithm 1), within the general layer-wise (L^0, L^1) -smoothness framework.

Assumption 4.4 (supp($\mathcal{D}$)-layer-wise (L^0, L^1) -smoothness II). The function $f : \mathcal{X} \mapsto \mathbb{R}$ is supp($\mathcal{D}$)-layer-wise (L^0, L^1) -smooth with constants $L^\alpha := \{(L_{1,S}^\alpha, \dots, L_{b,S}^\alpha)\}_{S \in \text{supp}(\mathcal{D})}$, $(L_{1,S}^\alpha, \dots, L_{b,S}^\alpha) \in \mathbb{R}_+^b$, $\alpha \in \{0, 1\}$, i.e., for any $S \in \text{supp}(\mathcal{D})$,

$$\|\nabla_i f(X + \Gamma) - \nabla_i f(X)\|_{(i)\star} \leq \left(L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star} \right) \|\Gamma_i\|_{(i)}$$

for all $X = [X_1, \dots, X_b] \in \mathcal{X}$ and $\Gamma = [\Gamma_1, \dots, \Gamma_b] \in \mathcal{X}$ such that $\Gamma_i = 0$ for all $i \notin S$.

Assumption 4.4 is a slightly stronger variant of Assumption 4.3 used in the analysis of Algorithm 2 (see Lemma H.3). Consequently, all results proven under Assumption 4.3 carry over to this setting. This stronger form is necessary to rigorously establish convergence in the stochastic case.

Furthermore, we assume access to a standard stochastic gradient oracle with bounded variance.

Assumption 4.5. The stochastic gradient estimator $\nabla f(\cdot; \xi) : \mathcal{X} \mapsto \mathcal{X}$ is unbiased and has bounded variance, i.e., $\mathbb{E}_{\xi \sim \mathcal{D}} [\nabla f(X; \xi)] = \nabla f(X)$ for all $X \in \mathcal{X}$ and there exist $\sigma_i \geq 0$ such that $\mathbb{E}_{\xi \sim \mathcal{D}} [\|\nabla_i f(X; \xi) - \nabla_i f(X)\|_2^2] \leq \sigma_i^2$ for all $X \in \mathcal{X}$ and $i = 1, \dots, p$.

Note that, to facilitate the proofs, the variance bound in Assumption 4.5 is defined with respect to the Euclidean norm, consistent with prior analyses [Pethick et al., 2025, Kovalev, 2025, Riabinin et al., 2025]. This is not restrictive: since $\mathcal{X}$ is finite-dimensional, there exist $\underline{\rho}_i, \bar{\rho}_i > 0$ such that $\underline{\rho}_i \|X_i\|_{(i)} \leq \|X_i\|_2 \leq \bar{\rho}_i \|X_i\|_{(i)}$ for all $X_i \in \mathcal{X}_i$ (equivalently, $\underline{\rho}_i \|X_i\|_2 \leq \|X_i\|_{(i)\star} \leq \bar{\rho}_i \|X_i\|_2$).

Theorem 4.4. Let Assumptions 4.1, 4.4, and 4.5 hold. Let $\{X^k\}_{k=0}^K$, $K \geq 0$, be the iterates of Algorithm 1 initialized with $M_i^0 = \nabla_i f(X^0; \xi^0)$ and run with $t_i^k \equiv t_i = \eta_i / (K+1)^{3/4}$, where $0 < \eta_i^2 \leq \min \left\{ \frac{1}{4} (K+1)^{1/2} \left(\sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^1 \right)^{-1} \left(\sum_{s=1}^b p_s \max_{i \in [b]} L_{i,\{s,\dots,b\}}^1 \right)^{-1} \right\}$,

$\frac{\rho_i p_1}{16\bar{\rho}_i(1-\beta_i)} \left(\sum_{s=1}^i p_s \right)^{-1} \left(\sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^1 \right)^{-1} \left(\sum_{s=1}^b p_s \max_{i \in [b]} L_{i,\{s,\dots,b\}}^1 \right)^{-1}, 1 \Big\}$, and $\beta_i \equiv \beta = (K + 1)^{-1/2}$. Then

$$\begin{aligned} & \min_{k=0,\dots,K} \sum_{i=1}^b \frac{\left(\sum_{s=1}^i p_s \right) \eta_i}{\frac{1}{b} \sum_{l=1}^b \sum_{s=1}^l p_s \eta_l} \mathbb{E} \left[\left\| \nabla_i f(X^k) \right\|_{(i)\star} \right] \\ & \leq \frac{3\delta^0}{(K+1)^{1/4} \frac{1}{b} \sum_{l=1}^b \sum_{s=1}^l p_s \eta_l} + \frac{6}{(K+1)^{1/2}} \sum_{i=1}^b \frac{\eta_i \bar{\rho}_i \sigma_i}{\frac{1}{b} \sum_{l=1}^b \sum_{s=1}^l p_s \eta_l} + \sum_{i=1}^b \frac{2\bar{\rho}_i \sigma_i \left(\sum_{s=1}^i p_s \right) \eta_i}{(K+1)^{1/4} \frac{1}{b} \sum_{l=1}^b \sum_{s=1}^l p_s \eta_l} \\ & \quad + \sum_{i=1}^b \frac{2\bar{\rho}_i \eta_i^2}{\rho_i (K+1)^{1/4} \frac{1}{b} \sum_{l=1}^b \sum_{s=1}^l p_s \eta_l} \left(\sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^0 + \left(\sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^1 \right) \left(\sum_{s=1}^i p_s \frac{L_{i,\{s,\dots,b\}}^0}{L_{i,\{s,\dots,b\}}^1} \right) \right) \\ & \quad + \sum_{i=1}^b \frac{\eta_i^2}{2(K+1)^{3/4} \frac{1}{b} \sum_{l=1}^b \sum_{s=1}^l p_s \eta_l} \left(\sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^0 + \left(\sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^1 \right) \left(\sum_{s=1}^i p_s \frac{L_{i,\{s,\dots,b\}}^0}{L_{i,\{s,\dots,b\}}^1} \right) \right). \end{aligned}$$

Theorem 4.4 establishes an $\mathcal{O}(K^{-1/4})$ convergence rate for a weighted sum of gradient component norms, in line with the state-of-the-art results for SGD- and Muon-type methods [Cutkosky and Mehta, 2020, Sun et al., 2023, Kovalev, 2025, Riabinin et al., 2025]. As in the deterministic setting, one could attempt a cost analysis; however, the stochastic bound’s complexity necessitates a case-by-case treatment depending on which term dominates. While this prevents deriving as clean analytic results as in Section 4.2, our experiments (Section 6) clearly confirm that the approach remains effective in the stochastic setting, too.

5 Prior Work on Progressive Training

The concept of partial network updates has been explored in several prior works. Progressive Training (PT) [Karras et al., 2017] was first introduced in the context of Generative Adversarial Networks (GANs) [Goodfellow et al., 2014]. The central idea is to begin with a shallow network trained on low-resolution inputs and gradually increase both the input resolution and network depth by adding layers over time. This progressive growing strategy has been shown to improve computational efficiency and stabilize the training process. Despite its empirical success, PT has remained a largely heuristic method without formal convergence guarantees. An early attempt to provide such guarantees was made by Wang et al. [2022] with ProgFed, which also extended progressive training to distributed settings. However, the accompanying theoretical analysis has been criticized as vacuous [Szlendak et al., 2024]. A more rigorous treatment was proposed by Szlendak et al. [2024], who introduced Randomized Progressive Training (RPT). This method can be viewed as a randomized proxy for PT and was the first to establish theoretical convergence rates for progressive training on general smooth objectives, building on the framework of Randomized Coordinate Descent (RCD) [Nesterov, 2012, Richtárik and Takáč, 2014b]. Conceptually, RPT is closely related to our approach: like Drop-Muon, it updates a random subnetwork at each iteration. However, the method of Szlendak et al. [2024] is simply a form of sketched GD with a particular choice of sketch operator, and has several limitations that restrict its practical utility. First, it treats network parameters as a flat vector, ignoring the layer-wise structure. Consequently, it fails to exploit layer-specific, non-Euclidean update rules—failing, to recover methods such as Muon as special cases. Second, RPT requires non-stochastic gradient computation, making it computationally infeasible for large-scale architectures.

6 Experiments

We evaluate Drop-Muon on three benchmarks—MNIST, Fashion-MNIST, and CIFAR-10—using 3-layer convolutional neural networks (CNNs) of varying capacity. For all $i \in [b]$, we set $\|\cdot\|_{(i)} = \|\cdot\|_{2 \rightarrow 2}$, i.e., the spectral norm, consistent with Muon. We study RPT sampling strategy with several smallest-index sampling rules, including uniform and an epoch-shift distribution that gradually shifts probability mass from shallow to deep layers as training progresses (see Section G). Our baseline is standard Muon (i.e., Drop-Muon with spectral norms and full-network training) run under identical hyperparameter configurations. This design allows us to isolate the effect of partial network updates while controlling for all other factors. For each configuration, we record accuracy as a function of both epochs and wall-clock time. Further experiments and details of hyperparameter selection procedure are provided in Section G.

Results on MNIST. Figure 1 illustrates a representative run with uniform layer index sampling: although Drop-Muon converges slightly slower per epoch (left), its lower per-epoch cost (27s vs. 39s for Muon) yields faster overall training

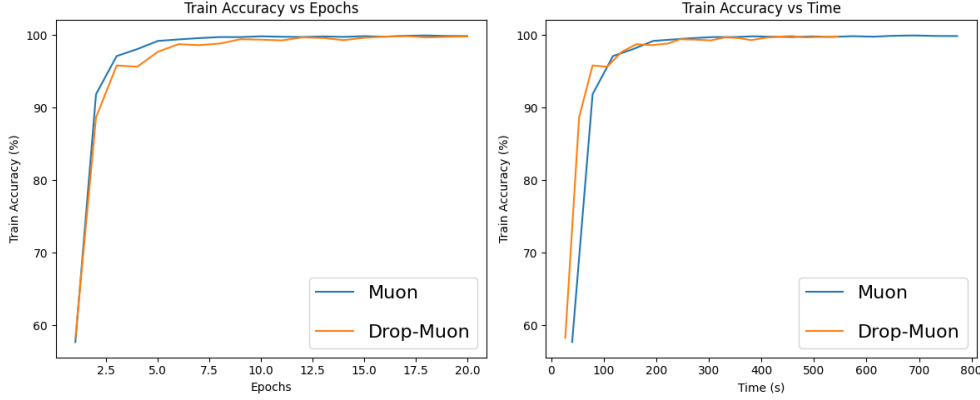


Figure 1: Evolution of the training accuracy for Muon and Drop-Muon with uniform index sampling on MNIST. Batch size = 8192, learning rate = 0.1, channels = [64, 128, 256].

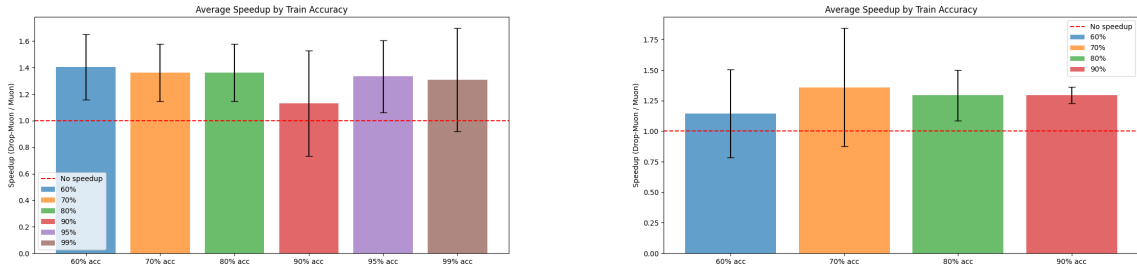


Figure 2: Averaged time-to-target speed-up over multiple runs comparing Muon and Drop-Muon with epoch-shift index sampling. Left: MNIST with batch size = 8192, learning rate = 0.1, and channels = [64, 128, 256]. Right: Fashion-MNIST with batch size = 32768, learning rate = 0.1, and channels = [64, 128, 256].

in wall-clock time (right). This advantage is even clearer with epoch-shift sampling: Figure 2 (left) aggregates time-to-target results over multiple seeds, showing that Drop-Muon consistently outperforms Muon and achieves up to $1.4\times$ speed-up.

Results on Fashion-MNIST. We repeat the experiment on Fashion-MNIST. Drop-Muon again delivers meaningful acceleration: as before, its per-epoch convergence is slightly slower, but it overtakes Muon in wall-clock time. Figure 2 (right) shows aggregated time-to-target results over multiple seeds, with Drop-Muon achieving roughly $1.2\times$ faster convergence on average across accuracy thresholds (we omit the 99% threshold since neither method reaches it).

Results on CIFAR-10. Figure 3 presents a representative run. Although the absolute gain is smaller than on MNIST or Fashion-MNIST, Drop-Muon still reaches 90% training accuracy earlier in wall-clock time (see also Figure 9).

In summary, even though Drop-Muon trains on partial gradients, it reaches high accuracy levels earlier than Muon in wall-clock time. Importantly, Drop-Muon is simple to implement, requiring only a few lines of code. The observed speed-up over Muon could be even greater with dedicated tuning: in our experiments, both methods use the same constant learning rates, whereas theory (Theorems 4.1, 4.2, and 4.4) and the practice of coordinate descent (Section A.3) suggest that Drop-Muon can safely employ larger stepsizes than the full-network Muon baseline. Here, we kept the learning rates identical for both optimizers to isolate the wall-clock benefit of partial layer updates, but additional gains are likely achievable with method-specific tuning. Overall, our results demonstrate that Drop-Muon is an effective, practical, and easily implementable strategy that, with a well-chosen sampling strategy, consistently accelerates training while retaining high accuracy, making it a compelling choice for modern neural network optimization.

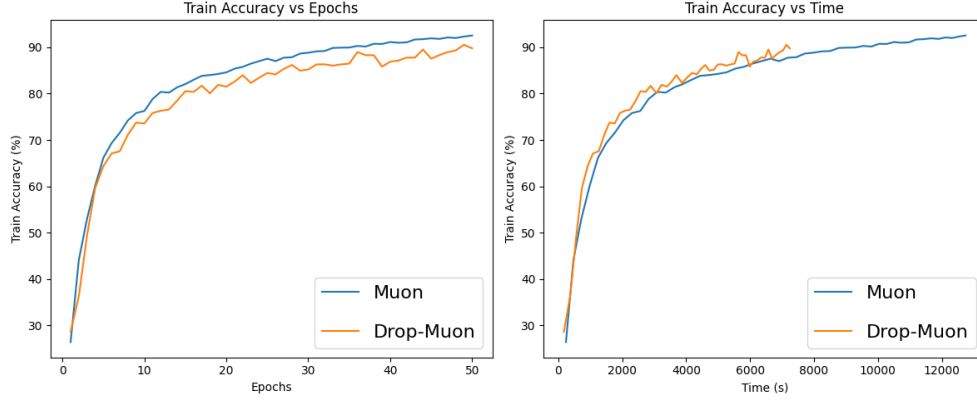


Figure 3: Evolution of the training accuracy for Muon and Drop-Muon with epoch-shift index sampling on CIFAR-10. Batch size = 8192, learning rate = 0.1, channels = [128, 256, 512].

Acknowledgments

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST): i) KAUST Baseline Research Scheme, ii) CRG Grant ORFS-CRG12-2024-6460, and iii) Center of Excellence for Generative AI, under award number 5940.

References

- Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018. URL <https://arxiv.org/abs/1802.04434>. (Cited on page 16)
- Å. Björck and C. Bowie. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971. URL <https://doi.org/10.1137/0708036>. (Cited on page 15)
- Ziang Chen, Yingzhou Li, and Jianfeng Lu. On the global convergence of randomized coordinate gradient descent for nonconvex optimization. *SIAM Journal on Optimization*, 33(2):713–738, 2023a. doi:10.1137/21M1460375. URL <https://doi.org/10.1137/21M1460375>. (Cited on page 17)
- Ziyi Chen, Yi Zhou, Yingbin Liang, and Zhaosong Lu. Generalized-smooth nonconvex optimization is as efficient as smooth nonconvex optimization. In *International Conference on Machine Learning*, pages 5396–5427. PMLR, 2023b. URL <https://arxiv.org/abs/2303.02854>. (Cited on page 5 and 16)
- Michael Crawshaw, Mingrui Liu, Francesco Orabona, Wei Zhang, and Zhenxun Zhuang. Robustness to unbounded smoothness of generalized signSGD. *Advances in neural information processing systems*, 35:9955–9968, 2022. URL <https://arxiv.org/abs/2208.11195>. (Cited on page 15 and 16)
- Ashok Cutkosky and Harsh Mehta. Momentum improves normalized SGD. In *International conference on machine learning*, pages 2260–2268. PMLR, 2020. URL <https://arxiv.org/abs/2002.03305>. (Cited on page 3, 8, and 49)
- Cong D. Dang and Guanghui Lan. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. *SIAM Journal on Optimization*, 25(2):856–881, 2015. doi:10.1137/130936361. URL <https://doi.org/10.1137/130936361>. (Cited on page 16)
- Werner Dinkelbach. On nonlinear fractional programming. *Management Science*, 13:492–498, 1967. URL <https://api.semanticscholar.org/CorpusID:119939254>. (Cited on page 28)
- Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015. URL <https://arxiv.org/abs/1312.5799>. (Cited on page 5)
- Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. URL <https://arxiv.org/abs/1406.2661>. (Cited on page 8)
- Ruichen Jiang, Devyani Maladkar, and Aryan Mokhtari. Convergence analysis of adaptive gradient methods under refined smoothness and noise assumptions. *arXiv preprint arXiv:2406.04592*, 2024. URL <https://arxiv.org/abs/2406.04592>. (Cited on page 16)

- Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL <https://kellerjordan.github.io/posts/muon/>. (Cited on page 1, 4, and 15)
- Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. URL <https://arxiv.org/abs/1710.10196>. (Cited on page 8)
- Jonathan A Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 217–226. SIAM, 2014. URL <https://arxiv.org/abs/1304.2338>. (Cited on page 3)
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL <https://arxiv.org/abs/1412.6980>. (Cited on page 1)
- Dmitry Kovalev. Understanding gradient orthogonalization for deep learning via non-Euclidean trust-region optimization, 2025. URL <https://arxiv.org/abs/2503.12645>. (Cited on page 1, 5, 7, and 8)
- Zdislav Kovarik. Some iterative methods for improving orthonormality. *SIAM Journal on Numerical Analysis*, 7(3): 386–389, 1970. URL <https://doi.org/10.1137/0707031>. (Cited on page 15)
- Haochuan Li, Jian Qian, Yi Tian, Alexander Rakhlin, and Ali Jadbabaie. Convex and non-convex optimization under generalized smoothness. *Advances in Neural Information Processing Systems*, 36:40238–40271, 2023. URL <https://arxiv.org/abs/2306.01264>. (Cited on page 16)
- Jiaxiang Li and Mingyi Hong. A note on the convergence of Muon and further, 2025. URL <https://arxiv.org/abs/2502.02900>. (Cited on page 5)
- Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025. URL <https://arxiv.org/abs/2502.16982>. (Cited on page 1)
- Yuxing Liu, Rui Pan, and Tong Zhang. AdaGrad under anisotropic smoothness, 2024. URL <https://arxiv.org/abs/2406.15244>. (Cited on page 16)
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL <https://arxiv.org/abs/1711.05101>. (Cited on page 1 and 15)
- Moonshot AI. Kimi K2: Open agentic intelligence, 2025. URL <https://moonshotai.github.io/Kimi-K2/>. (Cited on page 1)
- Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. URL <https://epubs.siam.org/doi/10.1137/100802001>. (Cited on page 3, 8, and 16)
- Andrei Patrascu and Ion Necoara. Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. *Journal of Global Optimization*, 61(1):19–46, 2015. doi:10.1007/s10898-014-0151-9. URL <https://doi.org/10.1007/s10898-014-0151-9>. (Cited on page 16)
- Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained LMOs. *arXiv preprint arXiv:2502.07529*, 2025. URL <https://arxiv.org/abs/2502.07529>. (Cited on page 1, 4, 5, 7, and 15)
- Michael J. D. Powell. On search directions for minimization algorithms. *Mathematical Programming*, 4:193–201, 1973. URL <https://link.springer.com/article/10.1007/BF01584660>. (Cited on page 16)
- Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling I: Algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016. URL <https://arxiv.org/abs/1412.8060>. (Cited on page 5 and 16)
- Artem Riabinin, Egor Shulgin, Kaja Grutkowska, and Peter Richtárik. Gluon: Making Muon & Scion great again! (Bridging theory and practice of LMO-based optimizers for LLMs). *arXiv preprint arXiv:2505.13416*, 2025. URL <https://arxiv.org/abs/2505.13416>. (Cited on page 1, 4, 5, 6, 7, 8, 15, 16, 23, and 66)
- Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014a. URL <https://arxiv.org/abs/1107.2848>. (Cited on page 5)
- Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1):1–38, 2014b. URL <https://arxiv.org/abs/1107.2848>. (Cited on page 8 and 16)

- Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016. URL <https://arxiv.org/abs/1212.0873>. (Cited on page 16)
- Ishaan Shah, Anthony M Polloreno, Karl Stratos, Philip Monk, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, Anil Thomas, Ashish Tanwer, Darsh J Shah, et al. Practical efficiency of Muon for pretraining. *arXiv preprint arXiv:2505.02222*, 2025. URL <https://arxiv.org/abs/2505.02222>. (Cited on page 1)
- Richard V. Southwell. *Relaxation Methods in Engineering Science: A Treatise on Approximate Computation*. Read Books, 1940. (Cited on page 16)
- Tao Sun, Qingsong Wang, Dongsheng Li, and Bao Wang. Momentum ensures convergence of SignSGD under weaker assumptions. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 33077–33099. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/sun231.html>. (Cited on page 8)
- Rafał Szlendak, Elnur Gasanov, and Peter Richtarik. Understanding progressive training through the framework of randomized coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pages 2161–2169. PMLR, 2024. URL <https://arxiv.org/abs/2306.03626>. (Cited on page 4 and 8)
- Benjamin Thérien, Xiaolong Huang, Irina Rish, and Eugene Belilovsky. MuLoCo: Muon is a practical inner optimizer for DiLoCo. *arXiv preprint arXiv:2505.23725*, 2025. URL <https://arxiv.org/abs/2505.23725>. (Cited on page 1)
- Hui-Po Wang, Sebastian Stich, Yang He, and Mario Fritz. ProgFed: Effective, communication, and computation efficient federated learning by progressive training. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23034–23054. PMLR, 17–23 Jul 2022. URL <https://proceedings.mlr.press/v162/wang22y.html>. (Cited on page 8)
- Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. *arXiv preprint arXiv:2509.02046*, 2025. URL <https://arxiv.org/abs/2509.02046>. (Cited on page 1)
- Stephen J. Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, 2015. URL <https://arxiv.org/abs/1502.04759>. (Cited on page 5 and 16)
- Shuo Xie, Mohamad Amin Mohamadi, and Zhiyuan Li. Adam exploits ℓ_∞ -geometry of loss landscape via coordinate-wise adaptivity. *arXiv preprint arXiv:2410.08198*, 2024. URL <https://arxiv.org/abs/2410.08198>. (Cited on page 16)
- Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020. URL <https://arxiv.org/abs/1905.11881>. (Cited on page 5 and 15)

Appendix

Contents

1	Introduction	1
1.1	Background	2
2	The Algorithm	2
3	Cost Model	3
4	Randomized Progressive Training	4
4.1	Iteration Complexity – Deterministic Gradient Setting	5
4.1.1	Cost Optimization	6
4.2	Iteration Complexity – Stochastic Gradient Setting	7
5	Prior Work on Progressive Training	8
6	Experiments	8
A	Additional Literature Review	15
A.1	Muon (and Friends)	15
A.2	Generalized Smoothness	15
A.3	Randomized Block Coordinate Descent	16
B	More on Smoothness Assumptions	18
C	Arbitrary Sampling	20
D	Convergence Results – Deterministic Gradient Case	21
D.1	Layer-Wise Smooth Case	21
D.2	Layer-Wise Generalized Smooth Case	22
E	Optimizing the Cost – Deterministic Gradient Case	26
E.1	Smooth Case	27
E.1.1	Randomized Progressive Training	28
E.1.2	τ -nice Sampling	30
E.1.3	τ -Submodel Sampling	33
E.1.4	Arbitrary Submodel Sampling	36
E.2	Generalized Smooth Case	39
E.2.1	Randomized Progressive Training	40
E.2.2	τ -Nice Sampling	43
E.2.3	τ -Submodel Sampling	45
E.2.4	Arbitrary Submodel Sampling	46

F	Convergence Results – Stochastic Gradient Case	48
G	Experiments	61
G.1	Key Training Parameters	61
G.2	Epoch-Shift Distribution	61
G.3	Evaluation Metrics	62
G.4	Results on MNIST	62
G.5	Results on Fashion-MNIST	63
G.6	Results on CIFAR-10	64
G.7	Discussion and Practical Remarks	64
H	Useful Facts	66

A Additional Literature Review

A.1 Muon (and Friends)

Recent work has revisited how neural network parameters are updated, moving beyond simple element-wise gradient steps toward more structured update rules. Notable examples include the Muon optimizer [Jordan et al., 2024], as well as the closely related Scion [Pethick et al., 2025] and Gluon [Riabini et al., 2025]. All of these methods can be interpreted as instances of algorithms driven by a *linear minimization oracle* (LMO) over norm balls.

Muon, introduced by Jordan et al. [2024], is an optimizer for the hidden layers of neural networks (with first and last layers typically trained using AdamW [Loshchilov and Hutter, 2019]). Given a layer X_i and the associated (stochastic) gradient G_i , the update is obtained by solving the constrained optimization problem

$$\min_{\Delta X_i} \langle G_i, \Delta X_i \rangle \quad \text{subject to} \quad \|\Delta X_i\|_{2 \rightarrow 2} \leq t_i, \quad (9)$$

where $t_i > 0$ plays the role of a trust-region radius or stepsize. The solution is characterized by the singular vectors of G_i : if $G_i = U_i \Sigma_i V_i^\top$ is its singular value decomposition, then

$$\Delta X_i = -t_i U_i V_i^\top, \quad X_i^{k+1} = X_i^k + \Delta X_i. \quad (10)$$

In other words, Muon moves in the direction of steepest descent measured in the spectral norm.

Since computing an exact SVD can be expensive, Muon uses the Newton-Schulz method [Kovarik, 1970, Björck and Bowie, 1971] to approximate the orthogonalization. In practice, the algorithm also incorporates momentum, yielding the update

$$\begin{aligned} M_i^k &= (1 - \beta_i) M_i^{k-1} + \beta_i G_i^k, \\ X_i^{k+1} &= X_i^k - t_i^k \text{NewtonSchulz}(M_i^k), \end{aligned}$$

where $\beta_i \in (0, 1]$ is the momentum parameter and M_i^k is the running average of past gradients.

The update rule in (10) can be interpreted as a special case of the more general LMO-based update described in Section 1. By selecting different norms for the LMO ball constraint, one obtains different algorithmic variants. Since our primary focus is on deep learning applications, we are especially interested in matrix norms. Within this setting, a particularly important family is that of *operator norms*, defined for any $A \in \mathbb{R}^{m \times n}$ as

$$\|A\|_{\alpha \rightarrow \beta} := \sup_{\|Z\|_\alpha = 1} \|AZ\|_\beta,$$

where $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ denote norms on $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively.

In particular, Muon’s update in (9) corresponds to an LMO over the spectral norm ball $\mathcal{B}_i^{2 \rightarrow 2}(0, t_i) := \{Z_i \in \mathcal{X}_i : \|Z_i\|_{2 \rightarrow 2} \leq t_i\}$, leading to the equivalent update

$$X_i^{k+1} = X_i^k + \text{LMO}_{\mathcal{B}_i^{2 \rightarrow 2}(0, t_i)}(G_i).$$

Hence, Muon is just one member of a broader family of methods parameterized by the geometry of the update set.

Building on this perspective, Pethick et al. [2025] introduced Scion, which extends LMO-based updates to *all* network layers, rather than being limited to the hidden matrix-shaped layers as in Muon. To better capture layer-specific behavior, Scion employs different norms depending on layer type (scaled spectral norms for the weight matrices of transformer blocks and the $\|\cdot\|_{1 \rightarrow \infty}$ norm for embedding and output layers).

Finally, Gluon [Riabini et al., 2025] generalizes the theory behind both Muon and Scion. It provides a convergence framework for updates over *arbitrary* norm balls. The analysis relies on a novel layer-wise smoothness model (closely related to our Assumptions 4.2, 4.3, and 4.4), capturing heterogeneity through (L^0, L^1) parameters, that more accurately reflects the non-uniform characteristics of deep learning models (see more details in Section A.2). As such, Gluon can be seen as the theoretical foundation unifying the entire class of LMO-based optimizers for training deep networks.

A.2 Generalized Smoothness

Gradient-based methods are traditionally analyzed under the assumption of Lipschitz smoothness of the gradient. Yet, many modern deep learning tasks violate this assumption [Zhang et al., 2020, Crawshaw et al., 2022], prompting the development of alternative smoothness models. One such model is (L^0, L^1) -smoothness, initially introduced by Zhang

Algorithm 3 Randomized Block Coordinate Descent

```

1: Input: Initial iterate  $X^0 = [X_1^0, \dots, X_b^0] \in \mathcal{X}$ ; stepsizes  $\gamma_i^k > 0, i \in [b], k \geq 0$ 
2: for  $k = 0, \dots, K - 1$  do
3:   Choose index  $i_k \in [b]$  randomly ▷ Serial sampling
4:   Set  $X_i^{k+1} = X_i^k$  for  $i \neq i_k$ 
5:   Update  $X_{i_k}^{k+1} = X_{i_k}^k - \gamma_{i_k}^k \nabla_{i_k} f(X^k)$ 
6: end for

```

et al. [2020] for twice continuously differentiable functions and later generalized beyond this setting [Li et al., 2023, Chen et al., 2023b].

This framework has been further tailored to deep learning through the non-Euclidean layer-wise (L^0, L^1) -smoothness assumption [Riabinin et al., 2025], which closely relates to our Assumptions 4.2, 4.3, and 4.4. Concretely, Riabinin et al. [2025] assume that

$$\|\nabla_i f(X) - \nabla_i f(Y)\|_{(i)\star} \leq (L_i^0 + L_i^1 \|\nabla_i f(X)\|_{(i)\star}) \|X_i - Y_i\|_{(i)} \quad (11)$$

for all $i \in [b]$ and all $X = [X_1, \dots, X_p] \in \mathcal{X}, Y = [Y_1, \dots, Y_p] \in \mathcal{X}$. Condition (11) can be interpreted as a global version of our Assumption 4.4. However, it is less precise because it does not allow each subset of layers to have its own effective smoothness constants. By adopting Assumption 4.4 instead of (11), we can derive tighter bounds without restricting the function class under consideration.

Accounting for heterogeneous parameter structures is not novel and has been studied in the context of coordinate descent [Nesterov, 2012, Richtárik and Takáč, 2014b] and in the analysis of algorithms such as signSGD [Bernstein et al., 2018, Crawshaw et al., 2022], AdaGrad [Jiang et al., 2024, Liu et al., 2024], and Adam [Xie et al., 2024].

A.3 Randomized Block Coordinate Descent

Coordinate Descent (CD) algorithms have a long history [Southwell, 1940, Powell, 1973, Nesterov, 2012, Wright, 2015] and are among the most widely studied methods for large-scale optimization. Their randomized variants, known as Randomized Block Coordinate Descent (RBCD), have emerged as a powerful class of first-order methods, particularly in high-dimensional problems. Instead of updating the entire parameter vector, RBCD updates only a subset (“block”) of variables at each iteration, reducing per-iteration computational cost and improving scalability.

In the context of CD, the parameters are viewed as a flat vector in $\mathbb{R}^d$. The general procedure of RBCD is summarized in Algorithm 3. At each iteration, one block is selected at random and updated, while the remaining blocks remain unchanged. Clearly, Algorithm 3 is a special case of Algorithm 2 with the *serial sampling* strategy (i.e., sampling a single coordinate per iteration) and standard Euclidean updates.

We define a family of linear maps $\{\mathbf{U}_i : \mathcal{X}_i \rightarrow \mathcal{X} : i \in [b]\}$ (where $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{X}_i = \mathbb{R}^{d_i}$, with $\sum_{i=1}^b d_i = d$), forming a partition of the identity map, i.e., $X = \sum_{i \in [b]} \mathbf{U}_i(X_i)$. The standard analysis of RBCD relies on the assumption of *blockwise Lipschitz continuity* of the gradient:

$$\|\nabla_i f(X + \mathbf{U}_i(\Gamma_i)) - \nabla_i f(X)\|_{(i)\star} \leq L_i \|\Gamma_i\|_{(i)}, \quad \forall i \in [b], X \in \mathcal{X}, \Gamma_i \in \mathcal{X}_i. \quad (12)$$

From (12), one obtains the standard quadratic upper bound

$$f(X + \mathbf{U}_i(\Gamma_i)) - f(X) - \langle \nabla_i f(X), \Gamma_i \rangle \leq \frac{L_i}{2} \|\Gamma_i\|_{(i)}^2. \quad (13)$$

This inequality plays a pivotal role in establishing the iteration complexity of RBCD. Moreover, it explains why CD-type methods can be advantageous compared to full gradient descent: they allow for larger stepsizes, which in turn leads to faster convergence. Indeed, the standard analysis of GD relies on *global* L -smoothness, enforcing stepsizes on the order of $1/L$. By contrast, the blockwise smoothness assumption (12) bounds the function’s variation when only a single block of coordinates is updated. Consequently, in CD, the safe stepsize for block i scales with $1/L_i$. Since L must capture worst-case variations across all directions (including cross-block interactions), it is often much larger than the individual L_i ’s.

Complexity analyses of RBCD in the convex setting can be found in Nesterov [2012], Wright [2015]. Generalizations to parallel and arbitrary sampling variants were developed by Richtárik and Takáč [2016], Qu and Richtárik [2016]. For the nonconvex case considered in this paper, iteration complexity results of Algorithm 2 are only known for serial sampling [Patrascu and Necoara, 2015, Dang and Lan, 2015]. Otherwise, some work has focused on the global

convergence properties of randomized CD methods [Chen et al., 2023a], rather than the iteration complexity analysis we develop here. It is worth noting that serial sampling is not suitable in the deep neural network context due to the special entanglement structure and the requirements of backpropagation for gradient computation.

B More on Smoothness Assumptions

Recall that a function $f : \mathcal{X} \mapsto \mathbb{R}$ is $\text{supp}(\mathcal{D})$ -layer-wise L^0 -smooth with constants $L^0 := \{(L_{1,S}^0, \dots, L_{b,S}^0)\}_{S \in \text{supp}(\mathcal{D})}$, $(L_{1,S}^0, \dots, L_{b,S}^0) \in \mathbb{R}_+^b$ (Assumption 4.2) if for any $S \in \text{supp}(\mathcal{D})$,

$$f(X + \Gamma) - f(X) - \langle \nabla f(X), \Gamma \rangle \leq \sum_{i \in S} \frac{L_{i,S}^0}{2} \|\Gamma_i\|_{(i)}^2$$

for all $X = [X_1, \dots, X_b] \in \mathcal{X}$ and $\Gamma = [\Gamma_1, \dots, \Gamma_b] \in \mathcal{X}$ such that $\Gamma_i = 0$ for all $i \notin S$.

Lemma B.1. Let $S_1, S_2 \in \text{supp}(\mathcal{D})$ satisfy $S_1 \subseteq S_2$, and suppose Assumption 4.2 holds. Then one can choose the constants $\{L_{i,S_1}^0\}_{i \in S_1}$ so that

$$L_{i,S_1}^0 \leq L_{i,S_2}^0.$$

Proof. Let $X = [X_1, \dots, X_b] \in \mathcal{X}$ be arbitrary. By Assumption 4.2, for any $\Gamma = [\Gamma_1, \dots, \Gamma_b] \in \mathcal{X}$ such that $\Gamma_i = 0$ for all $i \notin S_1$, we have

$$f(X + \Gamma) - f(X) - \langle \nabla f(X), \Gamma \rangle \leq \sum_{i \in S_1} \frac{L_{i,S_1}^0}{2} \|\Gamma_i\|_{(i)}^2.$$

Now, since $S_1 \subseteq S_2$ and $\Gamma_i = 0$ for all $i \in S_2 \setminus S_1$, Assumption 4.2 also gives

$$f(X + \Gamma) - f(X) - \langle \nabla f(X), \Gamma \rangle \leq \sum_{i \in S_2} \frac{L_{i,S_2}^0}{2} \|\Gamma_i\|_{(i)}^2 = \sum_{i \in S_1} \frac{L_{i,S_2}^0}{2} \|\Gamma_i\|_{(i)}^2.$$

Therefore, one can always choose $L_{i,S_1}^0 \leq L_{i,S_2}^0$. $\square$

Remark B.1. We can show that, without further assumptions, there is no finite function $C : [b] \rightarrow (0, \infty)$ such that

$$L_{i,S}^0 \leq C(|S|)L_{i,\{i\}}^0 \quad (14)$$

holds for all f . Consider the case $b = 2$ with scalar blocks and the Euclidean norm, suppose that $\text{supp}(\mathcal{D}) = \{\{1\}, \{2\}, \{1, 2\}\}$, and define

$$f(x, y) = \alpha xy$$

for some $\alpha > 0$. For a single-block perturbation in the x -coordinate only, we have

$$f(x + \gamma_1, y) - f(x, y) - \partial_x f(x, y)\gamma_1 = \alpha(x + \gamma_1)y - \alpha xy - \alpha\gamma_1 y = 0,$$

which shows that $L_{1,\{1\}}^0 = 0$. By symmetry, $L_{2,\{2\}}^0 = 0$ as well. However, for a joint perturbation $\Gamma = (\gamma_1, \gamma_2)$, we obtain

$$\begin{aligned} f(x + \gamma_1, y + \gamma_2) - f(x, y) - \langle \nabla f(x, y), \Gamma \rangle &= \alpha(x + \gamma_1)(y + \gamma_2) - \alpha xy - \alpha\gamma_1 y - \alpha\gamma_2 x \\ &= \alpha\gamma_1\gamma_2. \end{aligned}$$

Now, suppose for a contradiction that (14) holds for some function $C : [b] \rightarrow \mathbb{R}$. Then, since $L_{1,\{1\}}^0 = 0$, it must be that $L_{1,\{1,2\}}^0 = 0$. By Assumption 4.2, this requires

$$\alpha\gamma_1\gamma_2 \leq \frac{L_{1,\{1,2\}}^0}{2}\gamma_1^2 + \frac{L_{2,\{1,2\}}^0}{2}\gamma_2^2 = \frac{L_{2,\{1,2\}}^0}{2}\gamma_2^2.$$

But the right-hand side is independent of γ_1 , whereas the left-hand side grows without bound as $\gamma_1 \rightarrow \infty$ whenever $\gamma_2 \neq 0$. Hence no such constant $L_{2,\{1,2\}}^0$ can exist. This proves that no uniform bound of the form (14) can hold without additional assumptions.

Remark B.2. The above argument extends to any two sets with one contained in the other (not just singletons). Let $S_1, S_2 \in \text{supp}(\mathcal{D})$ satisfy $S_1 \subseteq S_2$. We show that, without further assumptions, there is no finite function $C : [b] \times [b] \rightarrow (0, \infty)$ such that

$$L_{i,S_2}^0 \leq C(|S_1|, |S_2|)L_{i,S_1}^0. \quad (15)$$

Without loss of generality we can restrict attention to the coordinates indexed by S_2 (so that $S_2 = [b]$). Let $S_1 = \{1, \dots, s\}$ and $S_2 \setminus S_1 = \{s+1, \dots, s+u\}$, so $|S_1| = s$, $|S_2| = s+u$, and $|S_2 \setminus S_1| = u$. Again, consider scalar blocks and define a function $f : \mathbb{R}^b \rightarrow \mathbb{R}$ via

$$f(x_1, \dots, x_s, y_1, \dots, y_u) = \alpha \left(\sum_{i=1}^s x_i \right) \left(\sum_{j=1}^u y_j \right)$$

for some $\alpha > 0$. For any increment Γ_{S_1} supported on S_1 only (i.e., $\gamma_j = 0$ for all $j \notin S_1$) the function f is affine in the x -variables with coefficients given by $\alpha \sum_j y_j$. Therefore, for any $X \in \mathbb{R}^b$,

$$\begin{aligned} f(X + \Gamma_{S_1}) - f(X) - \langle \nabla f(X), \Gamma_{S_1} \rangle \\ = \alpha \left(\sum_{i=1}^s (x_i + \gamma_i) \right) \left(\sum_{j=1}^u y_j \right) - \alpha \left(\sum_{i=1}^s x_i \right) \left(\sum_{j=1}^u y_j \right) - \alpha \sum_{i=1}^s \left(\sum_{j=1}^u y_j \right) \gamma_i = 0, \end{aligned}$$

and hence $L_{i,S_1}^0 = 0$ for every $i \in S_1$. For a joint perturbation $\Gamma_{S_2} = (\gamma_1, \dots, \gamma_s, \eta_1, \dots, \eta_u)$ supported on S_2 , we have

$$\begin{aligned} f(X + \Gamma_{S_2}) - f(X) - \langle \nabla f(X), \Gamma_{S_2} \rangle \\ = \alpha \left(\sum_{i=1}^s (x_i + \gamma_i) \right) \left(\sum_{j=1}^u (y_j + \eta_j) \right) - \alpha \left(\sum_{i=1}^s x_i \right) \left(\sum_{j=1}^u y_j \right) \\ - \alpha \sum_{i=1}^s \left(\sum_{j=1}^u y_j \right) \gamma_i - \alpha \sum_{i=1}^s \left(\sum_{j=1}^u x_j \right) \eta_i \\ = \alpha \left(\sum_{i=1}^s \gamma_i \right) \left(\sum_{j=1}^u \eta_j \right). \end{aligned}$$

In particular, fix any nonzero vectors $\tilde{\gamma} = (\tilde{\gamma}_1, \dots, \tilde{\gamma}_s)$ and $\eta = (\eta_1, \dots, \eta_u)$ and let the x -perturbation scale by a factor λ , i.e. $\gamma_i = \lambda \tilde{\gamma}_i$. Then

$$f(X + \Gamma) - f(X) - \langle \nabla f(X), \Gamma \rangle = \alpha \lambda \left(\sum_{i=1}^s \tilde{\gamma}_i \right) \left(\sum_{j=1}^u \eta_j \right),$$

which grows linearly in λ . Now, suppose, for contradiction that (15) holds with some finite $C(|S_1|, |S_2|)$. Since $L_{i,S_1}^0 = 0$ for every $i \in S_1$, the inequality forces $L_{i,S_2}^0 = 0$ for all $i \in S_1$. But then Assumption 4.2 would yield

$$\begin{aligned} \alpha \lambda \left(\sum_{i=1}^s \tilde{\gamma}_i \right) \left(\sum_{j=1}^u \eta_j \right) &= \alpha \left(\sum_{i=1}^t \gamma_i \right) \left(\sum_{j=1}^u \eta_j \right) \\ &\leq \sum_{i \in S_1} \frac{L_{i,S_2}^0}{2} \gamma_i^2 + \sum_{i \in S_2 \setminus S_1} \frac{L_{i,S_2}^0}{2} \eta_i^2 \\ &= \sum_{i \in S_2 \setminus S_1} \frac{L_{i,S_2}^0}{2} \eta_i^2, \end{aligned}$$

where the right-hand side is independent of the scaling factor λ . Taking $\lambda \rightarrow \infty$ makes the left-hand side arbitrarily large, resulting in a contradiction. Hence no such C can exist.

C Arbitrary Sampling

One might wonder why the cost model in Section 3 distinguishes between two sets of constants $\{c_i\}_{i \in [b]}$ and $\{c_i^\sharp\}_{i \in [b]}$, given that in the RPT case, c_i and $c_i^\sharp$ could be combined into a single constant. This is because Algorithms 1 and 2 do not restrict us to the layer sampling scheme studied in Section 4. One could argue that deviating from this progressive training framework is inefficient, as it discards gradients obtained “for free” during backpropagation: by design, at iteration k , the gradients $[\nabla_{s^k} f(X^k), \dots, \nabla_b f(X^k)]$, where $s^k := \min S^k$, are necessarily computed, and thus available for the update (if, for some block $i \in [b]$, the cost $c_i^\sharp$ of applying the sharp operator is large, one can simply bypass this step and instead use the standard gradient update; this corresponds to setting $\|\cdot\|_{(i)} = \|\cdot\|_2$ in the algorithm). Nevertheless, since the general iteration complexity results in Theorems D.1, D.4 and F.1 hold for any distribution $\mathcal{D}$, the framework naturally allows experimentation with alternative sampling schemes, even if only out of theoretical curiosity.

With this in mind, we generalize the results from Section 4 to *arbitrary* layer samplings. Formally, at each iteration k , Algorithms 1 and 2 sample a random subset of layers $S^k \subseteq [b]$ from a distribution

$$\mathcal{D} : \mathfrak{P}([b]) \rightarrow [0, 1], \quad \sum_{S \subseteq [b]} \mathcal{D}(S) = 1,$$

with support $\text{supp}(\mathcal{D}) := \{S \in \mathfrak{P}([b]) : \mathcal{D}(S) > 0\}$, where $\mathfrak{P}([b])$ denotes the power set of $[b]$. Only the layers in S^k are updated, while the rest remain fixed. We write $\hat{S} \sim \mathcal{D}$ for a set-valued random variable with the same distribution as S^k . This formulation defines a broad family of algorithms, parameterized jointly by the sampling distribution $\mathcal{D}$ and the choice of layer-wise norms $\|\cdot\|_{(i)}$.

Mirroring the earlier discussion, we organize the analysis around two smoothness regimes—layer-wise smooth (Assumption 4.2) and generalized layer-wise smooth (Assumptions 4.3 and 4.4).

In what follows, we present convergence results for Algorithms 1 (Theorem F.1) and 2 (Theorems D.1 and D.4). When specialized to the RPT setting, these general results recover the guarantees stated in Theorems 4.1, 4.2, and 4.4, as detailed in Remarks D.3, D.6, and F.3.

D Convergence Results – Deterministic Gradient Case

D.1 Layer-Wise Smooth Case

Theorem D.1. Let Assumptions 4.1 and 4.2 hold, and let $\{X^k\}_{k=0}^{K-1}$ be the iterates of Algorithm 2 run with stepsizes $\gamma_i^k = 1/L_{i,S^k}^0$. Then

$$\frac{1}{K} \sum_{k=0}^{K-1} \sum_{i=1}^b \frac{w_i}{\frac{1}{b} \sum_{j=1}^b w_j} \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star}^2 \right] \leq \frac{f(X^0) - f^*}{K \left(\frac{1}{b} \sum_{j=1}^b w_j \right)},$$

where $w_i := \mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right]$.

Remark D.2. For Theorem D.1 to be meaningful, the sampling must ensure that $w_i > 0$ for every $i \in [b]$, i.e.,

$$\mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right] > 0.$$

Otherwise, some layers would receive zero weight and the bound would provide no control over them. This is a natural condition, requiring that every layer is sampled with positive probability. Indeed, if $\mathbb{P}(i \in \hat{S}) = 0$, then the expectation vanishes, so $w_i > 0$ necessarily implies $\mathbb{P}(i \in \hat{S}) > 0$ for all $i \in [b]$.

Remark D.3. In the case of RPT (see Section 4), the weights from Theorem D.1 become

$$\mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right] = \sum_{s=1}^b \frac{\mathbb{I}(i \in \{s, \dots, b\})}{2L_{i,\{s, \dots, b\}}^0} p_s = \sum_{s=1}^i \frac{p_s}{2L_{i,\{s, \dots, b\}}^0}.$$

Substituting this expression into the rate yields the result in Theorem 4.1.

Proof of Theorem D.1. Using Assumption 4.2, we get

$$\begin{aligned} & f(X^{k+1}) \\ & \leq f(X^k) + \langle \nabla f(X^k), X^{k+1} - X^k \rangle + \sum_{i \in S^k} \frac{L_{i,S^k}^0}{2} \|X_i^k - X_i^{k+1}\|_{(i)}^2 \\ & = f(X^k) + \sum_{i \in S^k} \left(\langle \nabla_i f(X^k), X_i^{k+1} - X_i^k \rangle_{(i)} + \frac{L_{i,S^k}^0}{2} \|X_i^k - X_i^{k+1}\|_{(i)}^2 \right) \\ & = f(X^k) + \sum_{i \in S^k} \left(-\gamma_i^k \langle \nabla_i f(X^k), (\nabla_i f(X^k))^\# \rangle_{(i)} + \frac{L_{i,S^k}^0 (\gamma_i^k)^2}{2} \|(\nabla_i f(X^k))^\#\|_{(i)}^2 \right) \\ & \stackrel{(50),(51)}{=} f(X^k) + \sum_{i \in S^k} \left(-\gamma_i^k \|\nabla_i f(X^k)\|_{(i)\star}^2 + \frac{L_{i,S^k}^0 (\gamma_i^k)^2}{2} \|\nabla_i f(X^k)\|_{(i)\star}^2 \right). \end{aligned}$$

Choosing $\gamma_i^k = 1/L_{i,S^k}^0$ and rearranging, we get

$$\sum_{i \in S^k} \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{2L_{i,S^k}^0} \leq f(X^k) - f(X^{k+1}).$$

Taking expectation conditional on X^k (denoted as $\mathbb{E}_k[\cdot]$) gives

$$f(X^k) - \mathbb{E}_k[f(X^{k+1})] \geq \mathbb{E}_k \left[\sum_{i \in S^k} \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{2L_{i,S^k}^0} \right]$$

$$\begin{aligned}
 &= \mathbb{E}_k \left[\sum_{i=1}^b \mathbb{I}(i \in S^k) \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{2L_{i,S^k}^0} \right] \\
 &= \sum_{i=1}^b \mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right] \|\nabla_i f(X^k)\|_{(i)\star}^2 \\
 &= \sum_{i=1}^b w_i \|\nabla_i f(X^k)\|_{(i)\star}^2,
 \end{aligned}$$

where we denoted $w_i := \mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right]$ and $\mathbb{I}(\cdot)$ is the indicator function (i.e., for any event A , $\mathbb{I}(A) = 1$ if A and $\mathbb{I}(A) = 0$ otherwise). Taking full expectation, summing over the first K iterations and dividing by $\frac{K}{b} \sum_{j=1}^b w_j$, we get

$$\begin{aligned}
 \frac{1}{K} \sum_{k=0}^{K-1} \sum_{i=1}^b \frac{w_i}{\frac{1}{b} \sum_{j=1}^b w_j} \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star}^2 \right] &\leq \frac{1}{\frac{K}{b} \sum_{j=1}^b w_j} \sum_{k=0}^{K-1} (\mathbb{E}[f(X^k)] - \mathbb{E}[f(X^{k+1})]) \\
 &\leq \frac{f(X^0) - f^*}{K \left(\frac{1}{b} \sum_{j=1}^b w_j \right)}.
 \end{aligned}$$

□

D.2 Layer-Wise Generalized Smooth Case

Theorem D.4. Let Assumptions 4.1 and 4.3 hold, fix $\varepsilon > 0$, and let $\{X^k\}_{k=0}^{K-1}$ be the iterates of Algorithm 2 run with stepsizes $\gamma_i^k = \left(L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star} \right)^{-1}$. Then, to guarantee that

$$\min_{k=0, \dots, K-1} \sum_{i=1}^b \left[\frac{w_i}{\frac{1}{b} \sum_{l=1}^b w_l} \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star} \right] \right] \leq \varepsilon,$$

it suffices to run the algorithm for

$$K = \left\lceil \frac{2\delta^0 \sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S}) \mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}]}{\left(\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}] \right)^2}}{\varepsilon^2 \left(\frac{1}{b} \sum_{l=1}^b w_l \right)^2} + \frac{2\delta^0}{\varepsilon \left(\frac{1}{b} \sum_{l=1}^b w_l \right)} \right\rceil$$

iterations, where $\delta^0 := f(X^0) - f^*$ and $w_i := \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]}$.

Remark D.5. The guarantee in Theorem D.4 is meaningful only if $w_i > 0$ for all $i \in [b]$, which is equivalent to requiring that $\mathbb{P}(i \in \hat{S}) > 0$ for all $i \in [b]$.

Remark D.6. For the RPT case (see Section 4), we have

$$\mathbb{P}(i \in \hat{S}) = \mathbb{P}(s \leq i) = \sum_{s=1}^i p_s$$

and

$$\mathbb{E} \left[L_{i,\hat{S}}^\alpha | i \in \hat{S} \right] = \frac{\mathbb{E} \left[L_{i,\hat{S}}^\alpha \mathbb{I}(i \in \hat{S}) \right]}{\mathbb{P}(i \in \hat{S})} = \frac{\sum_{s=1}^i p_s L_{i,\{s, \dots, b\}}^\alpha}{\sum_{s=1}^i p_s}$$

for $\alpha \in \{0, 1\}$. Hence the weights from Theorem D.4 become

$$\frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E} \left[L_{i,\hat{S}}^1 | i \in \hat{S} \right]} = \frac{\left(\sum_{s=1}^i p_s \right)^2}{\sum_{s=1}^i p_s L_{i,\{s, \dots, b\}}^\alpha}.$$

Substituting this expression into the rate yields the result in Theorem 4.2.

Remark D.7. When $\hat{S} = [b]$ with probability 1, the weights become $w_i = 1/L_{i,[b]}^1$, and hence Theorem D.4 guarantees that

$$\min_{k=0,\dots,K-1} \sum_{i=1}^b \left[\frac{1/L_{i,[b]}^1}{\frac{1}{b} \sum_{l=1}^b 1/L_{l,[b]}^1} \|\nabla_i f(X^k)\|_{(i)\star} \right] \leq \varepsilon,$$

after

$$K = \left\lceil \frac{2\delta^0 \sum_{i=1}^b L_{i,[b]}^0 / (L_{i,[b]}^1)^2}{\varepsilon^2 \left(\frac{1}{b} \sum_{l=1}^b 1/L_{l,[b]}^1 \right)^2} + \frac{2\delta^0}{\varepsilon \left(\frac{1}{b} \sum_{l=1}^b 1/L_{l,[b]}^1 \right)} \right\rceil$$

iterations, recovering the rate of Gluon [Riabini et al., 2025].

Proof of Theorem D.4. Starting with Assumption 4.3, we have

$$\begin{aligned} & f(X^{k+1}) \\ & \leq f(X^k) + \sum_{i \in S^k} \left(\langle \nabla_i f(X^k), X_i^{k+1} - X_i^k \rangle_{(i)} \right. \\ & \quad \left. + \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star}}{2} \|X_i^{k+1} - X_i^k\|_{(i)}^2 \right) \\ & = f(X^k) + \sum_{i \in S^k} \left(-\gamma_i^k \langle \nabla_i f(X^k), (\nabla_i f(X^k))^\# \rangle_{(i)} \right. \\ & \quad \left. + \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star}}{2} (\gamma_i^k)^2 \|(\nabla_i f(X^k))^\#\|_{(i)}^2 \right) \\ & \stackrel{(50),(51)}{=} f(X^k) + \sum_{i \in S^k} \left(-\gamma_i^k \|\nabla_i f(X^k)\|_{(i)\star}^2 \right. \\ & \quad \left. + \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star}}{2} (\gamma_i^k)^2 \|\nabla_i f(X^k)\|_{(i)\star}^2 \right). \end{aligned}$$

Taking $\gamma_i^k = \frac{1}{L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star}}$ gives

$$f(X^{k+1}) \leq f(X^k) - \sum_{i \in S^k} \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{2 \left(L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star} \right)},$$

and hence

$$\begin{aligned} \sum_{k=0}^{K-1} \sum_{i \in S^k} \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{2 \left(L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star} \right)} & \leq \sum_{k=0}^{K-1} (f(X^k) - f(X^{k+1})) \\ & \leq f(X^0) - f^\star := \delta^0. \end{aligned}$$

Taking expectation

$$\begin{aligned} \delta^0 & \geq \sum_{k=0}^{K-1} \mathbb{E} \left[\sum_{i \in S^k} \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{2 \left(L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star} \right)} \right] \\ & = \sum_{k=0}^{K-1} \mathbb{E} \left[\mathbb{E} \left[\sum_{i=1}^b \mathbb{I}(i \in S^k) \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{2 \left(L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star} \right)} \middle| X^k \right] \right] \end{aligned}$$

$$\begin{aligned}
 &= \sum_{k=0}^{K-1} \sum_{i=1}^b \mathbb{E} \left[\mathbb{P}(i \in S^k) \mathbb{E} \left[\frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{2(L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star})} \middle| X^k, \{i \in S^k\} \right] \right] \\
 &\stackrel{(i)}{\geq} \sum_{k=0}^{K-1} \sum_{i=1}^b \mathbb{E} \left[\frac{\|\nabla_i f(X^k)\|_{(i)\star}^2 \mathbb{P}(i \in S^k)}{2(\mathbb{E}[L_{i,S^k}^0 | X^k, \{i \in S^k\}] + \mathbb{E}[L_{i,S^k}^1 | X^k, \{i \in S^k\}] \|\nabla_i f(X^k)\|_{(i)\star})} \right] \\
 &\stackrel{(ii)}{=} \frac{1}{2} \sum_{k=0}^{K-1} \sum_{i=1}^b \mathbb{E} \left[\frac{\|\nabla_i f(X^k)\|_{(i)\star}^2 \mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}] + \mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}] \|\nabla_i f(X^k)\|_{(i)\star}} \right] \\
 &\stackrel{(iii)}{\geq} \frac{1}{2} \sum_{k=0}^{K-1} \sum_{i=1}^b \frac{\mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}]^2 \mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}] + \mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}] \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}]}, \tag{16}
 \end{aligned}$$

where in (i) we used Jensen's inequality and convexity of the function $t \mapsto \frac{1}{t}$ for $t > 0$, (ii) follows from independence of $\hat{S}$ and X^k , and (iii) is a consequence of convexity of the function $t \mapsto \frac{t^2 \mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}] + \mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}] t}$ and Jensen's inequality.

Now, using Lemma H.2 with $x_i = \sqrt{\mathbb{P}(i \in \hat{S})} / \mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]$, $y_i = \sqrt{\mathbb{P}(i \in \hat{S})} \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}]$ and $z_i = \mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}] + \mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}] \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}]$, we obtain

$$\begin{aligned}
 &\sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S}) \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}]^2}{\mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}] + \mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}] \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}]} \\
 &\geq \frac{\left(\sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]} \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}] \right)^2}{\sum_{i=1}^b \left(\frac{\mathbb{P}(i \in \hat{S})}{(\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}])^2} \mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}] + \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]} \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}] \right)}.
 \end{aligned}$$

Applying this in (16), we get

$$\begin{aligned}
 \delta^0 &\geq \frac{1}{2} \sum_{k=0}^{K-1} \sum_{i=1}^b \frac{\mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}]^2 \mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}] + \mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}] \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}]} \\
 &\geq \frac{1}{2} \sum_{k=0}^{K-1} \frac{\left(\sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]} \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}] \right)^2}{\sum_{i=1}^b \left(\frac{\mathbb{P}(i \in \hat{S})}{(\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}])^2} \mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}] + \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]} \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}] \right)} \\
 &= \frac{1}{2} \sum_{k=0}^{K-1} \psi \left(\sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]} \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}] \right),
 \end{aligned}$$

where $\psi(t) := \frac{t^2}{\sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})}{(\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}])^2} \mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}] + t}$. Since ψ is increasing for $t > 0$, we have

$$\begin{aligned}
 \delta^0 &\geq \frac{1}{2} \sum_{k=0}^{K-1} \psi \left(\sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]} \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}] \right) \\
 &\geq \frac{K}{2} \psi \left(\min_{k=0, \dots, K-1} \sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]} \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}] \right).
 \end{aligned}$$

Moreover, since ψ is monotonic, it has an inverse ψ^{-1} . Thus

$$\begin{aligned}\psi^{-1}\left(\frac{2\delta^0}{K}\right) &\geq \min_{k=0,\dots,K-1} \sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]} \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}] \\ &= \min_{k=0,\dots,K-1} \sum_{i=1}^b w_i \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}],\end{aligned}$$

where $w_i := \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]}$. This in turn means that to reach the precision

$$\min_{k=0,\dots,K-1} \sum_{i=1}^b \left[\frac{w_i}{\frac{1}{b} \sum_{l=1}^b w_l} \mathbb{E}[\|\nabla_i f(X^k)\|_{(i)\star}] \right] \leq \varepsilon,$$

it suffices to run the algorithm for

$$\begin{aligned}K &= \left\lceil \frac{2\delta^0}{\psi\left(\varepsilon \left(\frac{1}{b} \sum_{l=1}^b w_l\right)\right)} \right\rceil = \left\lceil \frac{2\delta^0 \sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S}) \mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}]}{\left(\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]\right)^2} + 2\delta^0 \left(\varepsilon \left(\frac{1}{b} \sum_{l=1}^b w_l\right)\right)}{\left(\varepsilon \left(\frac{1}{b} \sum_{l=1}^b w_l\right)\right)^2} \right\rceil \\ &= \left\lceil \frac{2\delta^0 \sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S}) \mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}]}{\left(\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]\right)^2}}{\varepsilon^2 \left(\frac{1}{b} \sum_{l=1}^b \frac{\mathbb{P}(l \in \hat{S})}{\mathbb{E}[L_{l,\hat{S}}^1 | l \in \hat{S}]}\right)^2} + \frac{2\delta^0}{\varepsilon \left(\frac{1}{b} \sum_{l=1}^b \frac{\mathbb{P}(l \in \hat{S})}{\mathbb{E}[L_{l,\hat{S}}^1 | l \in \hat{S}]}\right)} \right\rceil\end{aligned}$$

iterations. □

E Optimizing the Cost – Deterministic Gradient Case

Let $S^k \subseteq \{1, \dots, b\}$ be the random subset sampled at iteration k , and $s^k := \min S^k$ its smallest index. Recall the per-iteration cost

$$\text{cost}(S^k) = c_{\text{ov}} + \sum_{i=s^k}^b c_i + \sum_{i \in S^k} c_i^\sharp.$$

Define the two marginal probabilities

$$F_i := \mathbb{P}(\hat{s} \leq i), \quad Q_i := \mathbb{P}(i \in \hat{S}),$$

where $\hat{s}$ is a random variable following the same distribution as s^k (since $S^k \sim \mathcal{D}$, $k \geq 0$, are i.i.d., the same holds for s^k). Since

$$\mathbb{E} \left[\sum_{i=s^k}^b c_i \right] = \sum_{j=1}^b \mathbb{P}(s^k = j) \sum_{i=j}^b c_i = \sum_{i=1}^b c_i \sum_{j=1}^i \mathbb{P}(s^k = j) = \sum_{i=1}^b c_i \mathbb{P}(s^k \leq i)$$

and

$$\mathbb{E} \left[\sum_{i \in S^k} c_i^\sharp \right] = \mathbb{E} \left[\sum_{i=1}^b \mathbb{I}(i \in S^k) c_i^\sharp \right] = \sum_{i=1}^b c_i^\sharp \mathbb{P}(i \in S^k),$$

the expected cost of one iteration is

$$\mathbb{E} [\text{cost}(\hat{S})] = c_{\text{ov}} + \sum_{i=1}^b c_i F_i + \sum_{i=1}^b c_i^\sharp Q_i. \quad (17)$$

Hence, to evaluate the expected cost, it suffices to compute the two marginals F_i and Q_i under the chosen sampling scheme.

In the sequel, we describe a few example sampling strategies considered in this work and analyze their costs under the layer-wise smooth setting (see Section E.1) and the generalized layer-wise smooth setting (see Section E.2).

1. **RPT.** Sample $\hat{s} \in \{1, \dots, b\}$, where $p_i = \mathbb{P}(\hat{s} = i)$, and set $\hat{S} = \{\hat{s}, \dots, b\}$. Then

$$F_i = \mathbb{P}(\hat{s} \leq i) = \sum_{j=1}^i p_j, \quad Q_i = \mathbb{P}(i \in \hat{S}) = \mathbb{P}(\hat{s} \leq i) = F_i.$$

Therefore,

$$\mathbb{E} [\text{cost}(\hat{S})] = c_{\text{ov}} + \sum_{i=1}^b (c_i + c_i^\sharp) F_i = c_{\text{ov}} + \sum_{i=1}^b (c_i + c_i^\sharp) \sum_{j=1}^i p_j.$$

2. **τ -nice sampling.** Choose $\hat{S}$ uniformly from all subsets of $[b]$ of size τ . Then

$$\begin{aligned} F_i &= 1 - \mathbb{P}(\hat{s} > i) = 1 - \frac{\binom{b-i}{\tau}}{\binom{b}{\tau}}, \\ Q_i &= \frac{\binom{b-1}{\tau-1}}{\binom{b}{\tau}} = \frac{\tau}{b}, \end{aligned}$$

where we use the convention $\binom{n}{k} = 0$ for $n < k$ (so the formula for F_i automatically gives $F_i = 1$ when $b - i < \tau$). Hence

$$\mathbb{E} [\text{cost}(\hat{S})] = c_{\text{ov}} + \sum_{i=1}^b c_i \left(1 - \frac{\binom{b-i}{\tau}}{\binom{b}{\tau}} \right) + \frac{\tau}{b} \sum_{i=1}^b c_i^\sharp.$$

3. **τ -submodel sampling.** Sample a starting index $\hat{s} \in \{1, \dots, b - \tau + 1\}$ with probability $p_i = \mathbb{P}(\hat{s} = i)$ (where $\tau \in [b]$ is fixed), and set $\hat{S} = \{\hat{s}, \dots, \hat{s} + \tau - 1\}$ (i.e., a block of τ consecutive layers). Then the marginals are

$$\begin{aligned} F_i &= \mathbb{P}(\hat{s} \leq i) = \sum_{j=1}^{\min\{i, b-\tau+1\}} p_j, \\ Q_i &= \mathbb{P}(i \in \hat{S}) = \sum_{j=\max\{1, i-\tau+1\}}^{\min\{i, b-\tau+1\}} p_j. \end{aligned}$$

The expected per-iteration cost is therefore

$$\mathbb{E}[\text{cost}(\hat{S})] = c_{\text{ov}} + \sum_{i=1}^b c_i \left(\sum_{j=1}^{\min\{i, b-\tau+1\}} p_j \right) + \sum_{i=1}^b c_i^\# \left(\sum_{j=\max\{1, i-\tau+1\}}^{\min\{i, b-\tau+1\}} p_j \right).$$

4. **Arbitrary submodel sampling.** Let $\{B_1, \dots, B_m\}$ be a partition of $[b]$ into disjoint blocks of arbitrary indices, i.e.,

$$B_1 \cup \dots \cup B_m = [b], \quad B_k \cap B_l = \emptyset \quad \text{for } k \neq l.$$

At each iteration, pick block B_i with probability p_i (where $\sum_{i=1}^m p_i = 1$) and set $\hat{S} = B_i$.

For $i \in [b]$, let $k(i)$ denote the unique block with $i \in B_{k(i)}$, and let $b_k := \min B_k$. Then

$$\begin{aligned} F_i &= \mathbb{P}(\hat{s} \leq i) = \sum_{j: b_j \leq i} p_j, \\ Q_i &= \mathbb{P}(i \in \hat{S}) = p_{k(i)}. \end{aligned}$$

The expected cost per iteration is

$$\mathbb{E}[\text{cost}(\hat{S})] = c_{\text{ov}} + \sum_{i=1}^b c_i \left(\sum_{k: b_k \leq i} p_k \right) + \sum_{i=1}^b c_i^\# p_{k(i)}.$$

We now consider the algorithm's performance under the two smoothness regimes.

E.1 Smooth Case

According to Theorem D.1, under Assumption 4.2, Algorithm 2 run with stepsizes $\gamma_i^k = 1/L_{i, S^k}^0$ guarantees that

$$\begin{aligned} \left(\min_{i \in [b]} w_i \right) \frac{1}{K} \sum_{k=0}^{K-1} \sum_{i=1}^b \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)^\star}^2 \right] &\leq \frac{1}{K} \sum_{k=0}^{K-1} \sum_{i=1}^b w_i \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)^\star}^2 \right] \\ &\leq \frac{f(X^0) - f^\star}{K}, \end{aligned}$$

where $w_i := \mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i, \hat{S}}^0} \right]$. Thus, to ensure that $\frac{1}{K} \sum_{k=0}^{K-1} \sum_{i=1}^b \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)^\star}^2 \right] \leq \varepsilon$, it suffices to run it for

$$K = \left\lceil \frac{f(X^0) - f^\star}{\varepsilon (\min_{i \in [b]} w_i)} \right\rceil$$

iterations. Now, recall from (17) that the expected cost of a single iteration is

$$\mathbb{E}[\text{cost}(S^k)] = c_{\text{ov}} + \sum_{i=1}^b c_i \mathbb{P}(\min S^k \leq i) + \sum_{i=1}^b c_i^\# \mathbb{P}(i \in S^k).$$

Hence, the expected cost of the entire optimization procedure can be written as

$$\text{cost}_\varepsilon(\mathcal{D}) = K \times \mathbb{E}[\text{cost}(\hat{S})]$$

$$\begin{aligned}
 &= K \times \left(c_{\text{ov}} + \sum_{i=1}^b c_i \mathbb{P}(\min \hat{S} \leq i) + \sum_{i=1}^b c_i^\# \mathbb{P}(i \in \hat{S}) \right) \\
 &\propto \frac{c_{\text{ov}} + \sum_{i=1}^b c_i \mathbb{P}(\min \hat{S} \leq i) + \sum_{i=1}^b c_i^\# \mathbb{P}(i \in \hat{S})}{\min_{i \in [b]} \mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right]},
 \end{aligned}$$

and the cost minimization problem to be solved is

$$\min_{\mathcal{D}: \mathfrak{P}([b]) \rightarrow [0,1], \sum_{S \subseteq [b]} \mathcal{D}(S) = 1} \frac{c_{\text{ov}} + \sum_{i=1}^b c_i \mathbb{P}(\min \hat{S} \leq i) + \sum_{i=1}^b c_i^\# \mathbb{P}(i \in \hat{S})}{\min_{i \in [b]} \mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right]}. \quad (18)$$

The task above is an optimization over probability distributions on the power set $\mathfrak{P}([b])$, which has dimension 2^b , making a direct solution intractable for large b . Instead of tackling it in full generality, we can restrict $\mathcal{D}$ to some parametric family. For certain such families, the ratio objective simplifies to a linear-fractional program in the cumulative marginals, which has a closed form solution or can be solved efficiently (e.g., via the Dinkelbach algorithm [Dinkelbach, 1967]).

Let us now consider some specific examples, starting with the procedure considered in the main part of this paper.

E.1.1 Randomized Progressive Training

Sample $\hat{s} \in \{1, \dots, b\}$, where $p_i = \mathbb{P}(\hat{s} = i)$, and set $\hat{S} = \{\hat{s}, \dots, b\}$.

We first consider the randomized progressive training setting introduced in Section 4. Under this sampling strategy, we have

$$F_i = \mathbb{P}(\hat{s} \leq i) = \sum_{j=1}^i p_j, \quad Q_i = \mathbb{P}(i \in \hat{S}) = \mathbb{P}(\hat{s} \leq i) = F_i,$$

and hence

$$\mathbb{E}[\text{cost}(\hat{S})] = c_{\text{ov}} + \sum_{i=1}^b (c_i + c_i^\#) F_i = c_{\text{ov}} + \sum_{i=1}^b (c_i + c_i^\#) \sum_{j=1}^i p_j.$$

Combining it with the fact that

$$\mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right] = \sum_{s=1}^b \frac{\mathbb{I}(i \in \{s, \dots, b\})}{2L_{i,\{s, \dots, b\}}^0} p_s = \sum_{s=1}^i \frac{p_s}{2L_{i,\{s, \dots, b\}}^0},$$

we get

$$\text{cost}_\varepsilon(\mathcal{D}) \propto \frac{\mathbb{E}[\text{cost}(\hat{S})]}{\min_{i \in [b]} \mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right]} = \frac{c_{\text{ov}} + \sum_{i=1}^b (c_i + c_i^\#) \sum_{j=1}^i p_j}{\min_{i \in [b]} \left\{ \sum_{s=1}^i \frac{p_s}{2L_{i,\{s, \dots, b\}}^0} \right\}}. \quad (19)$$

Optimal probabilities. First, note that the numerator can be rewritten as

$$c_{\text{ov}} + \sum_{i=1}^b (c_i + c_i^\#) \sum_{j=1}^i p_j = c_{\text{ov}} + \sum_{j=1}^b \left[\sum_{i \geq j} (c_i + c_i^\#) \right] p_j = \sum_{j=1}^b \left[c_{\text{ov}} + \sum_{i \geq j} (c_i + c_i^\#) \right] p_j,$$

where the second equality follows from $\sum_{j=1}^b p_j = 1$. Denote

$$d_j := c_{\text{ov}} + \sum_{i \geq j} (c_i + c_i^\#)$$

for $j \in [b]$, and let

$$\delta_{i,s} := \frac{1}{2L_{i,\{s,\dots,b\}}}$$

for $i \in [b]$ and $s \leq i$. Clearly

$$d_1 > d_2 > \dots > d_b \quad (20)$$

and

$$\delta_{i,1} \leq \delta_{i,2} \leq \dots \leq \delta_{i,i} \quad \forall i \in [b]. \quad (21)$$

Based on (19), the search for optimal probabilities reduces to solving the following linear fractional program:

$$\begin{aligned} \min_{p,t} \quad & \frac{d_1 p_1 + \dots + d_b p_b}{t} \\ \text{s.t.} \quad & p_1, \dots, p_b \geq 0 \\ & p_1 + \dots + p_b = 1 \\ & t \geq 0 \\ & t \leq \delta_{1,1} p_1 \\ & \vdots \\ & t \leq \delta_{i,1} p_1 + \dots + \delta_{i,i} p_i \\ & \vdots \\ & t \leq \delta_{b,1} p_1 + \dots + \delta_{b,b} p_b. \end{aligned} \quad (22)$$

This program can be written equivalently as

$$\begin{aligned} \min_q \quad & d_1 q_1 + \dots + d_b q_b \\ \text{s.t.} \quad & q_1, \dots, q_b \geq 0 \\ & \delta_{1,1} q_1 \geq 1 \\ & \vdots \\ & \delta_{i,1} q_1 + \dots + \delta_{i,i} q_i \geq 1 \\ & \vdots \\ & \delta_{b,1} q_1 + \dots + \delta_{b,b} q_b \geq 1. \end{aligned} \quad (23)$$

Based on that, we can derive a recursive prescription for the optimal probabilities.

First, we show that if $\sum_{s=1}^i \delta_{i,s} q_s > 1$ and $q_i > 0$ for some i , then one can shift a small amount of mass from q_i to q_{i+1} and strictly reduce the objective, without violating any constraints.

Lemma E.1. Let q be an optimal point of (23) and fix $i \in [b-1]$. If $\sum_{s=1}^i \delta_{i,s} q_s > 1$, then $q_i = 0$. Equivalently,

$$q_i > 0 \quad \implies \quad \sum_{s=1}^i \delta_{i,s} q_s = 1. \quad (24)$$

Proof. Suppose that i is such that $\sum_{s=1}^i \delta_{i,s} q_s > 1$, but $q_i > 0$. Then there exists $\varepsilon > 0$ such that $q_i \geq \varepsilon$ and

$$0 < \varepsilon \leq \frac{\sum_{s=1}^i \delta_{i,s} q_s - 1}{\delta_{i,i}}.$$

Define

$$\tilde{q}_i = q_i - \varepsilon, \quad \tilde{q}_{i+1} = q_{i+1} + \varepsilon, \quad \tilde{q}_s = q_s \text{ for } s \notin \{i, i+1\}.$$

Then $\{\tilde{q}_i\}_{i \in [b]}$ satisfy all constraints:

- Constraint $k < i$: $\sum_{s=1}^k \delta_{k,s} \tilde{q}_s = \sum_{s=1}^k \delta_{k,s} q_s \geq 1$,
- Constraint i : $\sum_{s=1}^i \delta_{i,s} \tilde{q}_s = \sum_{s=1}^i \delta_{i,s} q_s - \delta_{i,i} \varepsilon \geq 1$ by the choice of ε ,
- Constraint $k > i$: $\sum_{s=1}^k \delta_{k,s} \tilde{q}_s = \sum_{s=1}^k \delta_{k,s} q_s + \varepsilon(\delta_{k,i+1} - \delta_{k,i}) \geq \sum_{s=1}^k \delta_{k,s} q_s \geq 1$ using the monotonicity $\delta_{k,i} \leq \delta_{k,i+1}$.

At the same time, the value of the objective decreases, as $\sum_j d_j \tilde{q}_j - \sum_j d_j q_j = (d_{i+1} - d_i) \varepsilon < 0$, contradicting optimality of q . Therefore, we must have $q_i = 0$. $\square$

Let us now use Lemma E.1 to derive a recursive construction of the optimal probabilities. Let $q^* = (q_1^*, \dots, q_b^*)$ be a solution of (23). As established in (24), any positive coordinate forces its constraint to be tight. Constraint 1 is $\delta_{1,1} q_1^* \geq 1$, meaning that $q_1^* > 0$, and so $\delta_{1,1} q_1^* = 1$. Thus

$$q_1^* = \frac{1}{\delta_{1,1}} = 2L_{1,\{1,\dots,b\}}.$$

Now, suppose that $q_1^*, \dots, q_{i-1}^*$ have already been determined. Then, constraint i reads

$$\sum_{s=1}^{i-1} \delta_{i,s} q_s^* + \delta_{i,i} q_i^* \geq 1.$$

Define the residual

$$r_i^* := 1 - \sum_{s=1}^{i-1} \delta_{i,s} q_s^* = 1 - \sum_{s=1}^{i-1} \frac{q_s^*}{2L_{i,\{s,\dots,b\}}}.$$

If $r_i^* \leq 0$, then $(q_1^*, \dots, q_{i-1}^*)$ already satisfy the constraint, so by (24), we must have $q_i^* = 0$. If $r_i^* > 0$, then for $(q_1^*, \dots, q_i^*)$ to satisfy the constraint, we need $q_i^* > 0$, and hence by Lemma E.1, we have $\sum_{s=1}^i \delta_{i,s} q_s^* = 1$, meaning that

$$q_i^* = \frac{r_i^*}{\delta_{i,i}} = 2r_i^* L_{i,\{i,\dots,b\}}.$$

Combining the above yields the recursion

$$\begin{aligned} q_1^* &= 2L_{1,\{1,\dots,b\}}, \\ q_i^* &= 2[r_i^*]_+ L_{i,\{i,\dots,b\}}, \quad r_i^* = 1 - \sum_{s=1}^{i-1} \frac{q_s^*}{2L_{i,\{s,\dots,b\}}}, \quad i \in \{2, \dots, b\}, \end{aligned}$$

where $[x]_+ := \max\{x, 0\}$. The optimal probabilities are finally recovered by normalization

$$p_i^* = \frac{q_i^*}{\sum_{j=1}^b q_j^*}.$$

Remark E.1. Note that $(p_1^*, p_2^*, \dots, p_b^*) = (1, 0, \dots, 0)$ if and only if $q_1^* = \sum_{j=1}^b q_j^*$, i.e., $q_2^* = \dots = q_b^* = 0$. But for that to be the case, we need $r_i^* \leq 0$ for all $i \in \{2, \dots, b\}$, so

$$1 \leq \sum_{s=1}^{i-1} \delta_{i,s} q_s^* = \delta_{i,1} q_1^* = \frac{\delta_{i,1}}{\delta_{1,1}} = \frac{L_{1,\{1,\dots,b\}}}{L_{i,\{1,\dots,b\}}} \quad \forall i \in \{2, \dots, b\}.$$

Therefore, choosing $p_1 = 1$ is optimal if and only if

$$L_{1,\{1,\dots,b\}} = \max_{i \in [b]} L_{i,\{1,\dots,b\}},$$

which proves Theorem 4.3.

E.1.2 τ -nice Sampling

Choose $\hat{S}$ uniformly from all subsets of $[b]$ of size τ .

Every τ -subset has probability $\binom{b}{\tau}^{-1}$. Thus

$$\mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right] = \frac{1}{\binom{b}{\tau}} \sum_{\substack{S \subseteq [b] \\ |S|=\tau}} \frac{\mathbb{I}(i \in S)}{2L_{i,S}^0},$$

and hence

$$\begin{aligned} \text{cost}_\varepsilon(\mathcal{D}) &\propto \frac{\mathbb{E} [\text{cost}(\hat{S})]}{\min_{i \in [b]} \mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right]} = \frac{c_{\text{ov}} + \sum_{j=1}^b c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}} \right) + \frac{\tau}{b} \sum_{j=1}^b c_j^\#}{\min_{i \in [b]} \frac{1}{\binom{b}{\tau}} \sum_{\substack{S \subseteq [b] \\ |S|=\tau}} \frac{\mathbb{I}(i \in S)}{2L_{i,S}^0}} \\ &= \frac{c_{\text{ov}} \binom{b}{\tau} + \sum_{j=1}^b c_j \left(\binom{b}{\tau} - \binom{b-j}{\tau} \right) + \binom{b}{\tau} \frac{\tau}{b} \sum_{j=1}^b c_j^\#}{\min_{i \in [b]} \left\{ \sum_{\substack{S \subseteq [b] \\ |S|=\tau}} \frac{\mathbb{I}(i \in S)}{2L_{i,S}^0} \right\}}. \end{aligned}$$

In general there is no closed-form expression for τ minimizing this cost because the denominator depends on $\{L_{i,S}^0\}_{S \subseteq [b]}$ in a highly problem-specific way. That said, it can be shown that $\tau = b$ need not be optimal (indeed, it can be very sub-optimal). For $\tau = b$, we always have $S = [b]$, and the cost becomes

$$\begin{aligned} \text{cost}_\varepsilon(\mathcal{D}) &\propto \frac{c_{\text{ov}} + \sum_{j=1}^b c_j \left(1 - \frac{\binom{b-j}{b}}{\binom{b}{b}} \right) + \sum_{j=1}^b c_j^\#}{\min_{i \in [b]} \frac{1}{2L_{i,[b]}^0}} \\ &= 2 \max_{i \in [b]} L_{i,[b]}^0 \left(c_{\text{ov}} + \sum_{j=1}^b (c_j + c_j^\#) \right) \end{aligned}$$

(recall that we use the convention $\binom{n}{k} = 0$ for $n < k$).

Now, let us make a simplifying assumption that $L_{i,S}^0$ depends only on i and $|S|$ and define $L_{i,\tau}^0 := L_{i,S}^0$ for $|S| = \tau$. Then

$$\begin{aligned} \text{cost}_\varepsilon(\mathcal{D}) &\propto \frac{c_{\text{ov}} \binom{b}{\tau} + \sum_{j=1}^b c_j \left(\binom{b}{\tau} - \binom{b-j}{\tau} \right) + \binom{b}{\tau} \frac{\tau}{b} \sum_{j=1}^b c_j^\#}{\min_{i \in [b]} \left\{ \sum_{\substack{S \subseteq [b] \\ |S|=\tau}} \frac{\mathbb{I}(i \in S)}{2L_{i,\tau}^0} \right\}} \\ &= \frac{c_{\text{ov}} \binom{b}{\tau} + \sum_{j=1}^b c_j \left(\binom{b}{\tau} - \binom{b-j}{\tau} \right) + \binom{b}{\tau} \frac{\tau}{b} \sum_{j=1}^b c_j^\#}{\min_{i \in [b]} \left\{ \frac{1}{2L_{i,\tau}^0} \binom{b-1}{\tau-1} \right\}} \\ &= 2 \max_{i \in [b]} L_{i,\tau}^0 \left(\frac{b}{\tau} c_{\text{ov}} + \sum_{j=1}^b c_j \left(\frac{b}{\tau} - \frac{\binom{b-j}{\tau}}{\binom{b-1}{\tau-1}} \right) + \sum_{j=1}^b c_j^\# \right). \end{aligned}$$

Define

$$A(\tau) := \max_{i \in [b]} L_{i,\tau}^0, \quad B(\tau) := \frac{b}{\tau} c_{\text{ov}} + \sum_{j=1}^b c_j \left(\frac{b}{\tau} - \frac{\binom{b-j}{\tau}}{\binom{b-1}{\tau-1}} \right) + \sum_{j=1}^b c_j^\#.$$

Then, the objective function to be minimized is

$$f(\tau) := A(\tau)B(\tau).$$

By definition, $\tau = b$ is optimal if and only if $f(b) \leq f(\tau)$ for all $\tau \in \{1, \dots, b-1\}$.

First, we show that B is decreasing in τ . To this end, let $1 \leq \tau_1 < \tau_2 \leq b$ and consider the difference

$$\begin{aligned} B(\tau_1) - B(\tau_2) &= \left(\frac{b}{\tau_1} - \frac{b}{\tau_2} \right) c_{\text{ov}} + \sum_{j=1}^b c_j \left(\frac{b}{\tau_1} - \frac{\binom{b-j}{\tau_1}}{\binom{b-1}{\tau_1-1}} - \frac{b}{\tau_2} + \frac{\binom{b-j}{\tau_2}}{\binom{b-1}{\tau_2-1}} \right) \\ &= \left(\frac{b}{\tau_1} - \frac{b}{\tau_2} \right) c_{\text{ov}} + \sum_{j=1}^b c_j (h_j(\tau_1) - h_j(\tau_2)), \end{aligned}$$

where $h_j(\tau) := \frac{b}{\tau} - \frac{\binom{b-j}{\tau}}{\binom{b-1}{\tau-1}}$. Clearly, the first term is positive. Let us focus on the second term. Using Pascal's identity, we have

$$h_{j+1}(\tau) - h_j(\tau) = \frac{\binom{b-j}{\tau} - \binom{b-j-1}{\tau}}{\binom{b-1}{\tau-1}} = \frac{\binom{b-j-1}{\tau-1}}{\binom{b-1}{\tau-1}}.$$

Therefore,

$$h_j(\tau) = h_j(\tau) - h_0(\tau) = \sum_{m=0}^{j-1} (h_{m+1}(\tau) - h_m(\tau)) = \sum_{m=0}^{j-1} \frac{\binom{b-m-1}{\tau-1}}{\binom{b-1}{\tau-1}},$$

where

$$\frac{\binom{b-m-1}{\tau-1}}{\binom{b-1}{\tau-1}} = \frac{b-\tau}{b-m-\tau} \frac{\binom{b-m-1}{\tau}}{\binom{b-1}{\tau}} \geq \frac{\binom{b-m-1}{\tau}}{\binom{b-1}{\tau}},$$

and the inequality is strict for $m > 0$. It follows that for any $j \in [b]$

$$h_j(\tau) = \sum_{m=0}^{j-1} \frac{\binom{b-m-1}{\tau-1}}{\binom{b-1}{\tau-1}} \geq \sum_{m=0}^{j-1} \frac{\binom{b-m-1}{\tau}}{\binom{b-1}{\tau}} = h_j(\tau+1).$$

Thus, except for the trivial case when $j = 1$ (where $h_1(\tau) \equiv 1$), the function $h_j(\tau)$ is strictly decreasing in τ , implying that $h_j(\tau_1) - h_j(\tau_2) > 0$, which proves that $B(\tau_1) > B(\tau_2)$. Thus

$$B(\tau) \geq B(b) \quad \forall \tau \leq b,$$

with strict inequality if $c_{\text{ov}} > 0$ and $\tau < b$.

Now, note that in general larger τ leads to a higher Lipschitz constant, and hence one may expect A to be an increasing function of τ . If $A(b)$ is much larger than $A(\tau)$, it may compensate for the decrease in $B(\tau)$, resulting in $f(\tau) = A(\tau)B(\tau) < A(b)B(b) = f(b)$, and so $\tau = b$ may not be optimal.

As an example, suppose that $L_{i,\tau}^0$ scales linearly with τ , i.e., $A(\tau) = \max_{i \in [b]} \tau L_i^0$ for some $L_i^0 \geq 0$. Then

$$\begin{aligned} \text{cost}_\varepsilon(\mathcal{D}) &\propto \max_{i \in [b]} L_i^0 \left(bc_{\text{ov}} + \sum_{j=1}^b c_j \left(b - \tau \frac{\binom{b-j}{\tau}}{\binom{b-1}{\tau-1}} \right) + \tau \sum_{j=1}^b c_j^\# \right) \\ &= \max_{i \in [b]} L_i^0 \left(bc_{\text{ov}} + b \sum_{j=1}^b c_j + \tau \sum_{j=1}^b \left(c_j^\# - c_j \frac{\binom{b-j}{\tau}}{\binom{b-1}{\tau-1}} \right) \right). \end{aligned}$$

Now, consider the function

$$\Phi(\tau) := \tau \sum_{j=1}^b \left(c_j^\# - c_j \frac{\binom{b-j}{\tau}}{\binom{b-1}{\tau-1}} \right).$$

Note that

$$\Phi(\tau+1) - \Phi(\tau) = (\tau+1) \sum_{j=1}^b \left(c_j^\# - c_j \frac{\binom{b-j}{\tau+1}}{\binom{b-1}{\tau}} \right) - \tau \sum_{j=1}^b \left(c_j^\# - c_j \frac{\binom{b-j}{\tau}}{\binom{b-1}{\tau-1}} \right)$$

$$\begin{aligned}
 &= \sum_{j=1}^b c_j \left(\tau \frac{\binom{b-j}{\tau}}{\binom{b-1}{\tau-1}} - (\tau+1) \frac{\binom{b-j}{\tau+1}}{\binom{b-1}{\tau}} \right) + \sum_{j=1}^b c_j^\# \\
 &= \sum_{j=1}^b c_j \frac{j(b-j)!(b-\tau-1)!}{(b-1)!(b-j-\tau)!} + \sum_{j=1}^b c_j^\# \geq 0
 \end{aligned}$$

(with the convention that the right-hand side is 0 when $b-j-\tau < 0$). Moreover, the increment is strictly positive if either $\sum_{j=1}^b c_j^\# > 0$ or there exists j with $c_j > 0$ and $b-j-\tau \geq 0$. Hence $\Phi(\tau)$ is non-decreasing in τ , and strictly increasing whenever one of these conditions holds, and the optimal choice is $\tau^* = 1$.

E.1.3 τ -Submodel Sampling

Sample a starting index $\hat{s} \in \{1, \dots, b-\tau+1\}$ with probability $p_i = \mathbb{P}(\hat{s} = i)$ and set $\hat{S} = \{\hat{s}, \dots, \hat{s} + \tau - 1\}$.

The denominator is

$$\mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right] = \sum_{j=1}^{b-\tau+1} p_j \frac{\mathbb{I}(i \in \{j, \dots, j+\tau-1\})}{2L_{i,\{j,\dots,j+\tau-1\}}^0} = \sum_{j=\max\{1, i-\tau+1\}}^{\min\{i, b-\tau+1\}} \frac{p_j}{2L_{i,\{j,\dots,j+\tau-1\}}^0}.$$

Hence the total expected cost is proportional to

$$\begin{aligned}
 \text{cost}_\varepsilon(\mathcal{D}) &\propto \frac{\mathbb{E}[\text{cost}(\hat{S})]}{\min_{i \in [b]} \mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right]} \\
 &= \frac{c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{i=1}^{\min\{j, b-\tau+1\}} p_i \right) + \sum_{j=1}^b c_j^\# \left(\sum_{i=\max\{1, j-\tau+1\}}^{\min\{j, b-\tau+1\}} p_i \right)}{\min_{i \in [b]} \left\{ \sum_{j=\max\{1, i-\tau+1\}}^{\min\{i, b-\tau+1\}} \frac{p_j}{2L_{i,\{j,\dots,j+\tau-1\}}^0} \right\}}.
 \end{aligned}$$

Partitioned τ -submodel sampling. Let us consider a special case of the above sampling scheme where the submodels assigned non-zero probability *partition* $[b]$. For simplicity, suppose that b is divisible by τ , let $m = b/\tau$, and define the block start indices via

$$s_k = (k-1)\tau + 1, \quad k = 1, \dots, m.$$

The algorithm then picks block $B_k := \{s_k, \dots, s_k + \tau - 1\}$ with probability $p_{s_k} > 0$ (where $\sum_{k=1}^m p_{s_k} = 1$). This is equivalent to the submodel sampling with starting-index distribution satisfying

$$p_{s_k} = p_{(k-1)\tau+1} > 0 \quad (k = 1, \dots, m), \quad p_j = 0 \text{ otherwise.}$$

Plugging this choice into the general submodel expressions immediately gives

$$\begin{aligned}
 F_i &= \sum_{j=1}^{\min\{i, b-\tau+1\}} p_j = \sum_{k=1}^{\min\{m, \lfloor (i-1)/\tau \rfloor + 1\}} p_{s_k}, \\
 Q_i &= \sum_{j=\max\{1, i-\tau+1\}}^{\min\{i, b-\tau+1\}} p_j = p_{s_{\lceil i/\tau \rceil}}, \\
 \mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0} \right] &= \sum_{j=\max\{1, i-\tau+1\}}^{\min\{i, b-\tau+1\}} \frac{p_j}{2L_{i,\{j,\dots,j+\tau-1\}}^0} = \frac{p_{s_{\lceil i/\tau \rceil}}}{2L_{i, B_{\lceil i/\tau \rceil}}^0}.
 \end{aligned}$$

Therefore the full cost reduces to

$$\text{cost}_\varepsilon(\mathcal{D}) \propto \frac{c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{k=1}^{\min\{m, \lfloor (j-1)/\tau \rfloor + 1\}} p_{s_k} \right) + \sum_{j=1}^b c_j^\# p_{s_{\lceil j/\tau \rceil}}}{\min_{i \in [b]} \left\{ \frac{p_{s_{\lceil i/\tau \rceil}}}{2L_{i, B_{\lceil i/\tau \rceil}}^0} \right\}}$$

$$= \frac{c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{k=1}^{\lceil j/\tau \rceil} p_{s_k} \right) + \sum_{j=1}^b c_j^{\#} p_{s_{\lceil j/\tau \rceil}}}{\min_{i \in [b]} \left\{ \frac{p_{s_{\lceil i/\tau \rceil}}}{2L_{i,B}^0} \right\}}. \quad (25)$$

Now, note that

$$\begin{aligned} \sum_{j=1}^b c_j \left(\sum_{k=1}^{\lceil j/\tau \rceil} p_{s_k} \right) &= \sum_{j=1}^{\tau} c_j p_{s_1} + \sum_{j=\tau+1}^{2\tau} c_j (p_{s_1} + p_{s_2}) + \dots + \sum_{j=(m-1)\tau+1}^{m\tau} c_j (p_{s_1} + \dots + p_{s_m}) \\ &= p_{s_1} \sum_{j=1}^{\tau} c_j + (p_{s_1} + p_{s_2}) \sum_{j=\tau+1}^{2\tau} c_j + \dots + (p_{s_1} + \dots + p_{s_m}) \sum_{j=(m-1)\tau+1}^{m\tau} c_j \\ &= p_{s_1} \sum_{j=1}^{m\tau} c_j + p_{s_2} \sum_{j=\tau+1}^{m\tau} c_j + \dots + p_{s_m} \sum_{j=(m-1)\tau+1}^{m\tau} c_j \end{aligned}$$

and

$$\sum_{j=1}^b c_j^{\#} p_{s_{\lceil j/\tau \rceil}} = \sum_{j=1}^{\tau} c_j^{\#} p_{s_1} + \sum_{j=\tau+1}^{2\tau} c_j^{\#} p_{s_2} + \dots + \sum_{j=(m-1)\tau+1}^{m\tau} c_j^{\#} p_{s_m}.$$

Hence

$$\begin{aligned} \text{cost}_{\varepsilon}(\mathcal{D}) &\propto \frac{1}{\min_{i \in [b]} \left\{ \frac{p_{s_{\lceil i/\tau \rceil}}}{2L_{i,B}^0} \right\}} \left[c_{\text{ov}} + p_{s_1} \left(\sum_{j=1}^{m\tau} c_j + \sum_{j=1}^{\tau} c_j^{\#} \right) + p_{s_2} \left(\sum_{j=\tau+1}^{m\tau} c_j + \sum_{j=\tau+1}^{2\tau} c_j^{\#} \right) + \dots \right. \\ &\quad \left. + p_{s_m} \left(\sum_{j=(m-1)\tau+1}^{m\tau} c_j + \sum_{j=(m-1)\tau+1}^{m\tau} c_j^{\#} \right) \right] \\ &= \frac{p_{s_1} d_1 + p_{s_2} d_2 + \dots + p_{s_m} d_m}{\min_{i \in [b]} \left\{ \frac{p_{s_{\lceil i/\tau \rceil}}}{2L_{i,B}^0} \right\}}, \end{aligned}$$

where we used the fact that $\sum_{j=1}^m p_{s_j} = 1$ and denoted $d_i := c_{\text{ov}} + \sum_{j=(i-1)\tau+1}^{m\tau} c_j + \sum_{j=(i-1)\tau+1}^{i\tau} c_j^{\#}$.

Optimal probabilities for fixed τ . We follow an approach similar to that in Section E.1.1 to find the optimal probabilities for a fixed choice of τ . For $i \in B_k$, define

$$\delta_{i,k} := \frac{1}{2L_{i,B_k}^0}$$

and set

$$\delta_k^{\min} := \min_{i \in B_k} \delta_{i,k} = \frac{1}{2 \max_{i \in B_k} L_{i,B_k}^0}.$$

Then the expected cost can be represented as

$$\text{cost}_{\varepsilon}(\mathcal{D}) \propto \frac{d_1 p_{s_1} + \dots + d_m p_{s_m}}{\min_{i \in [b]} \left\{ \frac{p_{s_{\lceil i/\tau \rceil}}}{2L_{i,B}^0} \right\}} = \frac{d_1 p_{s_1} + \dots + d_m p_{s_m}}{\min_{k \in [m]} \min_{i \in B_k} \{ \delta_{i,k} p_{s_k} \}}.$$

This corresponds to the linear fractional program

$$\begin{aligned} \min_{p,t} \quad & \frac{d_1 p_{s_1} + \dots + d_m p_{s_m}}{t} \\ \text{s.t.} \quad & p_{s_1}, \dots, p_{s_m} \geq 0, \\ & p_{s_1} + \dots + p_{s_m} = 1, \\ & t \geq 0, \\ & t \leq \delta_{i,k} p_{s_k}, \quad k \in [m], i \in B_k. \end{aligned} \quad (26)$$

The standard change of variables $q_k = p_{s_k}/t$ yields the equivalent linear program

$$\begin{aligned} \min_q \quad & d_1 q_1 + \dots + d_m q_m \\ \text{s.t.} \quad & q_1, \dots, q_m \geq 0, \\ & \delta_{i,k} q_k \geq 1 \quad k \in [m], i \in B_k. \end{aligned} \tag{27}$$

Because each block k only appears in constraints of the form $\delta_{i,k} q_k \geq 1$ (for $i \in B_k$), the constraints for block k reduce to the single constraint

$$\delta_k^{\min} q_k \geq 1 \iff q_k \geq \frac{1}{\delta_k^{\min}}.$$

Hence (27) is separable and its optimal solution is

$$q_k^* = \frac{1}{\delta_k^{\min}}, \quad k \in [m],$$

with optimal objective value

$$\sum_{k=1}^m d_k q_k^* = \sum_{k=1}^m \frac{d_k}{\delta_k^{\min}}.$$

Now, let us introduce dual variables $\lambda_i \geq 0$ for the constraints $\delta_{i,k} q_k \geq 1$, $i \in B_k$. The dual of (27) is

$$\begin{aligned} \max_{\lambda} \quad & \lambda_1 + \dots + \lambda_b \\ \text{s.t.} \quad & \lambda_1, \dots, \lambda_b \geq 0, \\ & d_k \geq \sum_{i \in B_k} \lambda_i \delta_{i,k}, \quad k \in [m]. \end{aligned}$$

To certify optimality of q^* , for each block k , choose an index $i_k \in B_k$ attaining the minimum $\delta_k^{\min}$ and set

$$\lambda_{i_k} = \frac{d_k}{\delta_{i_k,k}}, \quad \lambda_i = 0 \text{ for } i \notin \{i_1, \dots, i_m\}.$$

Then for each block k ,

$$\sum_{i \in B_k} \delta_{i,k} \lambda_i = \delta_{i_k,k} \lambda_{i_k} = d_k,$$

so the dual constraints hold with equality and the dual objective equals

$$\sum_{i=1}^b \lambda_i = \sum_{k=1}^m \frac{d_k}{\delta_k^{\min}},$$

matching the primal objective. By strong duality q^* is optimal.

From $q_k^* = 1/\delta_k^{\min}$ and $p_{s_k} = t q_k = q_k / \sum_{l=1}^m q_l$ we obtain the optimal block probabilities

$$p_{s_k}^* = \frac{1/\delta_k^{\min}}{\sum_{l=1}^m 1/\delta_l^{\min}} = \frac{\max_{i \in B_k} L_{i,B_k}^0}{\sum_{l=1}^m \max_{i \in B_l} L_{i,B_l}^0}.$$

so that each block's probability is proportional to the worst-case local smoothness constant inside that block. For this choice, the minimal expected cost is

$$\text{cost}_{\varepsilon}(\mathcal{D}) \propto \sum_{k=1}^m \frac{d_k}{\delta_k^{\min}} = 2 \sum_{k=1}^m d_k \max_{i \in B_k} L_{i,B_k}^0.$$

Choosing τ . We now show that the cost is not necessarily minimized by choosing $\tau = b$. To this end, we want to minimize the function

$$\Phi(\tau) := 2 \sum_{k=1}^m d_k(\tau) \max_{i \in B_k} L_{i, B_k}^0$$

(where we explicitly emphasize the dependence of d_k on τ).

To gain intuition about which extreme ($\tau = 1$ or $\tau = b$) may be preferable, assume that the costs are constant, i.e.,

$$c_j \equiv c, \quad c_j^\# \equiv c^\#,$$

in which case

$$\begin{aligned} \sum_{k=1}^m d_k(\tau) &= \sum_{k=1}^m \left(c_{\text{ov}} + \sum_{j=(k-1)\tau+1}^{m\tau} c + \sum_{j=(k-1)\tau+1}^{k\tau} c^\# \right) \\ &= \sum_{k=1}^m (c_{\text{ov}} + c(m\tau - (k-1)\tau) + c^\#(k\tau - (k-1)\tau)) \\ &= mc_{\text{ov}} + c\tau \frac{m(m+1)}{2} + m\tau c^\#. \end{aligned}$$

Suppose in addition that the worst-case local smoothness per block does not depend on the block index, i.e.,

$$\max_{i \in B_k} L_{i, B_k}^0 \equiv L^0(\tau) \quad \forall k \in [m]$$

for some non-decreasing function L . Then

$$\Phi(\tau) = 2L^0(\tau) \sum_{k=1}^m d_k(\tau) = 2L^0(\tau) \left(\frac{b}{\tau} \left(c_{\text{ov}} + \frac{bc}{2} \right) + \frac{bc}{2} + bc^\# \right).$$

Thus, the τ -dependence of Φ is the product of a non-decreasing factor $L^0(\tau)$ and a factor that decreases like $1/\tau$ plus additive constants. Consequently:

- If $L^0(\tau)$ is constant in τ (no worsening with larger blocks), then $\Phi(\tau)$ is strictly decreasing in τ and the minimizer is $\tau^* = b$.
- If $L^0(\tau)$ grows sublinearly, then $L^0(\tau) \frac{b}{\tau} (c_{\text{ov}} + \frac{bc}{2})$ is decreasing in τ , while $L^0(\tau) (\frac{bc}{2} + bc^\#)$ is increasing. Hence, the optimal τ may lie strictly between 1 and b , depending on the relative magnitudes of the costs.
- If $L^0(\tau)$ increases at least linearly in τ , then $\Phi(\tau)$ is increasing in τ , and hence $\tau^* = 1$.

E.1.4 Arbitrary Submodel Sampling

Let $\{B_1, \dots, B_m\}$ be a partition of $[b]$. Set $\hat{S} = B_i$ with probability p_i (where $\sum_{i=1}^m p_i = 1$). For $j \in [b]$, $k(j)$ denotes the unique block with $j \in B_{k(j)}$ and $b_k := \min B_k$.

First, note that the cost can be expressed block-wise as

$$\begin{aligned} \mathbb{E} [\text{cost}(\hat{S})] &= c_{\text{ov}} + \sum_{i=1}^b c_i \left(\sum_{k: b_k \leq i} p_k \right) + \sum_{i=1}^b c_i^\# p_{k(i)} \\ &= c_{\text{ov}} + \sum_{k=1}^m \sum_{i \geq b_k} c_i p_k + \sum_{k=1}^m \sum_{i \in B_k} c_i^\# p_k \\ &= \sum_{k=1}^m d_k p_k, \end{aligned}$$

where $d_k := c_{\text{ov}} + \sum_{j \geq b_k} c_j + \sum_{j \in B_k} c_j^\#$. We also have

$$\mathbb{E} \left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i, \hat{S}}^0} \right] = \frac{p_{k(i)}}{2L_{i, B_{k(i)}}^0},$$

and hence

$$\text{cost}_\varepsilon(\mathcal{D}) \propto \frac{\mathbb{E}[\text{cost}(\hat{S})]}{\min_{i \in [b]} \mathbb{E}\left[\frac{\mathbb{I}(i \in \hat{S})}{2L_{i,\hat{S}}^0}\right]} = \frac{\sum_{k=1}^m d_k p_k}{\min_{k \in [m]} \min_{i \in B_k} \left\{ \frac{p_k}{2L_{i,B_k}^0} \right\}}.$$

Now, define

$$\delta_{i,k} := \frac{1}{2L_{i,B_k}^0}, \quad \delta_k^{\min} := \min_{i \in B_k} \delta_{i,k}.$$

By the same linear fractional reduction as in (26), the unique optimal solution is

$$p_k^* = \frac{1/\delta_k^{\min}}{\sum_{l=1}^m 1/\delta_l^{\min}} = \frac{\max_{i \in B_k} L_{i,B_k}^0}{\sum_{l=1}^m \max_{i \in B_l} L_{i,B_l}^0}.$$

With this choice, the objective is

$$\text{cost}_\varepsilon(\mathcal{D}) \propto 2 \sum_{k=1}^m d_k \max_{i \in B_k} L_{i,B_k}^0 = 2 \sum_{k=1}^m \left(c_{\text{ov}} + \sum_{j \geq b_k} c_j + \sum_{j \in B_k} c_j^\# \right) \max_{i \in B_k} L_{i,B_k}^0. \quad (28)$$

Let us compare it with the cost for full model training

$$\mathbb{E}[\text{cost}_{\text{full}}(K)] \propto 2 \max_{i \in [b]} L_{i,[b]}^0 \left(c_{\text{ov}} + \sum_{j=1}^b (c_j + c_j^\#) \right).$$

Example 1 (Grouping layers by cost similarity). Suppose that the layers are partitioned into groups according to their similarity, so that within each block, all per-layer costs are (approximately) the same. Concretely, for every $k \in [m]$ and every $j \in B_k$ assume that

$$c_j \equiv \underline{c}_k, \quad c_j^\# \equiv \underline{c}_k^\#.$$

Then

$$\begin{aligned} \text{cost}_\varepsilon(\mathcal{D}) &\propto 2 \sum_{k=1}^m \left(c_{\text{ov}} + \sum_{j \geq b_k} c_j + \sum_{j \in B_k} c_j^\# \right) \max_{i \in B_k} L_{i,B_k}^0 \\ &= 2 \sum_{k=1}^m \left(c_{\text{ov}} + \sum_{j \geq b_k} c_j + |B_k| \underline{c}_k^\# \right) \max_{i \in B_k} L_{i,B_k}^0 \end{aligned}$$

and

$$\mathbb{E}[\text{cost}_{\text{full}}(K)] \propto 2 \max_{i \in [b]} L_{i,[b]}^0 \left(c_{\text{ov}} + \sum_{l=1}^m |B_l| (\underline{c}_l + \underline{c}_l^\#) \right).$$

Therefore, partitioned arbitrary submodel sampling with the optimal probabilities is better than full model training if and only if

$$\sum_{k=1}^m \max_{i \in B_k} L_{i,B_k}^0 \left(c_{\text{ov}} + \sum_{j \geq b_k} c_j + |B_k| \underline{c}_k^\# \right) \leq \max_{i \in [b]} L_{i,[b]}^0 \left(c_{\text{ov}} + \sum_{k=1}^m |B_k| (\underline{c}_k + \underline{c}_k^\#) \right). \quad (29)$$

If each block is much better conditioned than the full model, i.e. $\max_{i \in B_k} L_{i,B_k}^0 \ll \max_{i \in [b]} L_{i,[b]}^0$ for all k , then the left side of (29) can be much smaller than the right side and partitioned submodel sampling will be advantageous even if the block-wise tail sums are moderately large. Indeed, if $\max_{i \in B_k} L_{i,B_k}^0 \approx \frac{1}{m} \max_{i \in [b]} L_{i,[b]}^0$, then

$$\sum_{k=1}^m \max_{i \in B_k} L_{i,B_k}^0 \left(c_{\text{ov}} + \sum_{j \geq b_k} c_j + |B_k| \underline{c}_k^\# \right)$$

$$\begin{aligned}
 &\approx \frac{1}{m} \max_{i \in [b]} L_{i,[b]}^0 \sum_{k=1}^m \left(c_{\text{ov}} + \sum_{j \geq \underline{b}_k} c_j + |B_k| \underline{c}_k^\# \right) \\
 &= \max_{i \in [b]} L_{i,[b]}^0 \left(c_{\text{ov}} + \frac{1}{m} \sum_{k=1}^m \sum_{j \geq \underline{b}_k} c_j + \frac{1}{m} \sum_{k=1}^m |B_k| \underline{c}_k^\# \right),
 \end{aligned}$$

which improves upon full model training if $\frac{1}{m} \sum_{k=1}^m \sum_{j \geq \underline{b}_k} c_j + \frac{1}{m} \sum_{k=1}^m |B_k| \underline{c}_k^\# \leq \sum_{k=1}^m |B_k| (\underline{c}_k + \underline{c}_k^\#)$.

Example 2 (Grouping layers across transformer blocks). In their general forms, (28) and (29) are hard to interpret. Let us consider a simplified language model motivating example behind considering this sampling strategy. Suppose the network is composed of T identical transformer blocks, each containing L layers, so that $b = TL$, with the natural block structure: layer index $j \in [b]$ corresponds to position $l \in \{1, \dots, L\}$ inside transformer block $t \in [T]$ via $j = l + (t-1)L$.

Consider the partition that groups together *same-position* layers across transformer blocks:

$$B_l = \{l, l+L, l+2L, \dots, l+(T-1)L\}, \quad \forall l \in [L],$$

so $m = L$ and $|B_l| = T$ for every l . For each block B_l we have $\underline{b}_l = \min B_l = l$.

In this setting, for block B_l the tail sum that appears in d_l simplifies to

$$\sum_{j \geq \underline{b}_l} c_j = \sum_{j=l}^b c_j.$$

Note that we grouped the layers in such a way that the layers within each block are of the same type, and hence we may expect the costs to be roughly the same within each block. That is, we assume that

$$c_{l+tL} \equiv \underline{c}_l, \quad c_{l+tL}^\# \equiv \underline{c}_l^\#, \quad \forall l \in [L], t \in \{0, \dots, T-1\},$$

in which case the tail sum further simplifies to

$$\sum_{j=l}^b c_j = \sum_{j=l}^L \underline{c}_j + (T-1) \sum_{j=1}^L \underline{c}_j.$$

Hence, under the periodic costs assumption the block-wise cost constants become

$$d_l = c_{\text{ov}} + \sum_{j \geq \underline{b}_l} c_j + \sum_{j \in B_l} c_j^\# = c_{\text{ov}} + \sum_{j=l}^L \underline{c}_j + (T-1) \sum_{j=1}^L \underline{c}_j + T \underline{c}_l^\#.$$

If the local smoothness within a block are also approximately homogeneous, i.e.,

$$L_{i,B_l}^0 \equiv L_l^0 \quad \forall i \in B_l,$$

then

$$\max_{i \in B_l} L_{i,B_l}^0 = L_l^0,$$

and the previously derived expected cost (for the optimal block probabilities) reduces to

$$\text{cost}_\varepsilon(\mathcal{D}) \propto 2 \sum_{l=1}^L L_l^0 \left(c_{\text{ov}} + \sum_{j=l}^L \underline{c}_j + (T-1) \sum_{j=1}^L \underline{c}_j + T \underline{c}_l^\# \right).$$

Under the same periodicity assumptions, the full-model expected cost is

$$\mathbb{E}[\text{cost}_{\text{full}}(K)] \propto 2 \max_{i \in [b]} L_{i,[b]}^0 \left(c_{\text{ov}} + T \sum_{j=1}^L \underline{c}_j + T \sum_{j=1}^L \underline{c}_j^\# \right).$$

Comparing the two, partitioned sampling is better if

$$\sum_{l=1}^L L_l^0 \left(c_{\text{ov}} + \sum_{j=l}^L \underline{c}_j + (T-1) \sum_{j=1}^L \underline{c}_j + T \underline{c}_l^\# \right) \leq \max_{i \in [b]} L_{i,[b]}^0 \left(c_{\text{ov}} + T \sum_{j=1}^L \underline{c}_j + T \sum_{j=1}^L \underline{c}_j^\# \right). \quad (30)$$

Now, let us denote

$$\underline{C} := \sum_{j=1}^L \underline{c}_j, \quad \underline{C}^\# := \sum_{j=1}^L \underline{c}_j^\#.$$

Then (30) is equivalent to

$$\sum_{l=1}^L L_l^0 \left(c_{\text{ov}} + \sum_{j=l}^L \underline{c}_j \right) + T \sum_{l=1}^L L_l^0 \left(\frac{T-1}{T} \underline{C} + \underline{c}_l^\# \right) \leq T (\underline{C} + \underline{C}^\#) \max_{i \in [b]} L_{i,[b]}^0 + c_{\text{ov}} \max_{i \in [b]} L_{i,[b]}^0. \quad (31)$$

For large T , the dominant terms are those multiplied by T . Then, the leading-order criterion becomes

$$\sum_{l=1}^L L_l^0 (\underline{C} + \underline{c}_l^\#) \leq \max_{i \in [b]} L_{i,[b]}^0 (\underline{C} + \underline{C}^\#), \quad (32)$$

which can hold when $L_l^0 \ll \max_{i \in [b]} L_{i,[b]}^0$.

E.2 Generalized Smooth Case

According to Theorem D.4, under Assumption 4.3, Algorithm 2 run with stepsizes $\gamma_i^k = (L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)^\star})^{-1}$ guarantees that

$$\min_{k=0,\dots,K-1} \sum_{i=1}^b \left[\frac{w_i}{\frac{1}{b} \sum_{l=1}^b w_l} \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)^\star} \right] \right] \leq \varepsilon, \quad (33)$$

after

$$K = \left\lceil \frac{2\delta^0 \sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S}) \mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}]}{(\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}])^2}} + \frac{2\delta^0}{\varepsilon^2 \left(\frac{1}{b} \sum_{l=1}^b \frac{\mathbb{P}(l \in \hat{S})}{\mathbb{E}[L_{l,\hat{S}}^1 | l \in \hat{S}]} \right)^2} + \frac{2\delta^0}{\varepsilon \left(\frac{1}{b} \sum_{l=1}^b \frac{\mathbb{P}(l \in \hat{S})}{\mathbb{E}[L_{l,\hat{S}}^1 | l \in \hat{S}]} \right)} \right\rceil \quad (34)$$

iterations, where $\delta^0 := f(X^0) - f^\star$ and $w_i := \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]}$. To obtain a guarantee on an unweighted gradient sum, note that

$$\begin{aligned} \varepsilon &\geq \min_{k=0,\dots,K-1} \sum_{i=1}^b \left[\frac{w_i}{\frac{1}{b} \sum_{l=1}^b w_l} \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)^\star} \right] \right] \\ &\geq \frac{\min_{i \in [b]} w_i}{\frac{1}{b} \sum_{l=1}^b w_l} \min_{k=0,\dots,K-1} \sum_{i=1}^b \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)^\star} \right], \end{aligned}$$

and hence, substituting in (33) and (34), we have

$$\min_{k=0,\dots,K-1} \sum_{i=1}^b \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)^\star} \right] \leq \frac{\varepsilon \frac{1}{b} \sum_{l=1}^b w_l}{\min_{i \in [b]} w_i} := \varepsilon$$

after

$$K = \left\lceil \frac{2\delta^0 \sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S}) \mathbb{E}[L_{i,\hat{S}}^0 | i \in \hat{S}]}{(\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}])^2}} + \frac{2\delta^0}{\varepsilon^2 \min_{i \in [b]} \left(\frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]} \right)^2} + \frac{2\delta^0}{\varepsilon \min_{i \in [b]} \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E}[L_{i,\hat{S}}^1 | i \in \hat{S}]}} \right\rceil$$

iterations. Moreover, recall from (17) that the expected cost of a single iteration is

$$\mathbb{E} [\text{cost}(\hat{S})] = c_{\text{ov}} + \sum_{i=1}^b c_i F_i + \sum_{i=1}^b c_i^\# Q_i.$$

Hence, using the fact that

$$\mathbb{E} [L_{i,\hat{S}}^\alpha | i \in \hat{S}] = \frac{\mathbb{E} [L_{i,\hat{S}}^\alpha \mathbb{I}(i \in \hat{S})]}{\mathbb{P}(i \in \hat{S})}$$

for $\alpha \in \{0, 1\}$, the expected cost of the entire optimization procedure is

$$\begin{aligned} \text{cost}_\varepsilon(\mathcal{D}) &= K \times \mathbb{E} [\text{cost}(\hat{S})] \\ &= \left[\frac{2\delta^0 \sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S}) \mathbb{E} [L_{i,\hat{S}}^0 | i \in \hat{S}]}{(\mathbb{E} [L_{i,\hat{S}}^1 | i \in \hat{S}])^2} + \frac{2\delta^0}{\varepsilon \min_{i \in [b]} \frac{\mathbb{P}(i \in \hat{S})}{\mathbb{E} [L_{i,\hat{S}}^1 | i \in \hat{S}]}} \right] \left(c_{\text{ov}} + \sum_{i=1}^b c_i F_i + \sum_{i=1}^b c_i^\# Q_i \right) \\ &= \left[\frac{2\delta^0 \sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})^2 \mathbb{E} [L_{i,\hat{S}}^0 \mathbb{I}(i \in \hat{S})]}{(\mathbb{E} [L_{i,\hat{S}}^1 \mathbb{I}(i \in \hat{S})])^2} + \frac{2\delta^0}{\varepsilon \min_{i \in [b]} \frac{\mathbb{P}(i \in \hat{S})^2}{\mathbb{E} [L_{i,\hat{S}}^1 \mathbb{I}(i \in \hat{S})]}} \right] \left(c_{\text{ov}} + \sum_{i=1}^b c_i F_i + \sum_{i=1}^b c_i^\# Q_i \right). \end{aligned}$$

We will consider two regimes.

(1) The $\mathcal{O}(1/\varepsilon^2)$ term dominates. Then the problem to be solved is

$$\min_{\mathcal{D}: \mathfrak{P}([b]) \rightarrow [0,1], \sum_{S \subseteq [b]} \mathcal{D}(S)=1} \underbrace{\frac{\sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})^2 \mathbb{E} [L_{i,\hat{S}}^0 \mathbb{I}(i \in \hat{S})]}{(\mathbb{E} [L_{i,\hat{S}}^1 \mathbb{I}(i \in \hat{S})])^2}}{\min_{i \in [b]} \left(\frac{\mathbb{P}(i \in \hat{S})^2}{\mathbb{E} [L_{i,\hat{S}}^1 \mathbb{I}(i \in \hat{S})]} \right)^2} \left(c_{\text{ov}} + \sum_{i=1}^b c_i F_i + \sum_{i=1}^b c_i^\# Q_i \right)}_{\propto \mathbb{E}[\text{cost}_{\varepsilon^2}(K)]}. \quad (35)$$

(2) The $\mathcal{O}(1/\varepsilon)$ term dominates. Then the problem to be solved is

$$\min_{\mathcal{D}: \mathfrak{P}([b]) \rightarrow [0,1], \sum_{S \subseteq [b]} \mathcal{D}(S)=1} \underbrace{\frac{1}{\min_{i \in [b]} \frac{\mathbb{P}(i \in \hat{S})^2}{\mathbb{E} [L_{i,\hat{S}}^1 \mathbb{I}(i \in \hat{S})]}} \left(c_{\text{ov}} + \sum_{i=1}^b c_i F_i + \sum_{i=1}^b c_i^\# Q_i \right)}_{\propto \mathbb{E}[\text{cost}_\varepsilon(K)]}. \quad (36)$$

As in Section E.1, both tasks above involve optimization over probability distributions, which is intractable in general. Again, for certain parametric families, the objective simplifies to a linear-fractional program, which can be solved efficiently. Let us consider some specific examples.

E.2.1 Randomized Progressive Training

Sample $\hat{s} \in \{1, \dots, b\}$, where $p_i = \mathbb{P}(\hat{s} = i)$, and set $\hat{S} = \{\hat{s}, \dots, b\}$.

In this case, $\hat{S} = \{s, \dots, b\}$ with probability p_s for all $s \in [b]$, and hence

$$\mathbb{P}(i \in \hat{S}) = \mathbb{P}(s \leq i) = \sum_{s=1}^i p_s,$$

and

$$\mathbb{E} [L_{i,\hat{S}}^\alpha \mathbb{I}(i \in \hat{S})] = \sum_{s=1}^i p_s L_{i,\{s, \dots, b\}}^\alpha$$

for $\alpha \in \{0, 1\}$. Moreover, following the same steps as in Section E.1.1,

$$\mathbb{E}[\text{cost}(\hat{S})] = c_{\text{ov}} + \sum_{j=1}^b (c_j + c_j^\#) \sum_{i=1}^j p_i = \sum_{i=1}^b \left[c_{\text{ov}} + \sum_{j \geq i} (c_j + c_j^\#) \right] p_i.$$

Hence, substituting into (35) and (36), the respective optimization problems reduce to minimizing

$$\begin{aligned} \mathbb{E}[\text{cost}_{\varepsilon^2}(K)] &\propto \frac{\sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})^2 \mathbb{E}[L_{i,\hat{S}}^0(i \in \hat{S})]}{(\mathbb{E}[L_{i,\hat{S}}^1(i \in \hat{S})])^2}}{\min_{i \in [b]} \left(\frac{\mathbb{P}(i \in \hat{S})^2}{\mathbb{E}[L_{i,\hat{S}}^1(i \in \hat{S})]} \right)} \left(\sum_{i=1}^b \left[c_{\text{ov}} + \sum_{j \geq i} (c_j + c_j^\#) \right] p_i \right) \\ &= \frac{\sum_{i=1}^b \frac{(\sum_{s=1}^i p_s)^2 \sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^0}{(\sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^1)^2}}{\min_{i \in [b]} \left(\frac{(\sum_{s=1}^i p_s)^2}{\sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^1} \right)} \left(\sum_{i=1}^b \left[c_{\text{ov}} + \sum_{j \geq i} (c_j + c_j^\#) \right] p_i \right) \end{aligned}$$

and

$$\begin{aligned} \mathbb{E}[\text{cost}_{\varepsilon}(K)] &\propto \frac{1}{\min_{i \in [b]} \frac{\mathbb{P}(i \in \hat{S})^2}{\mathbb{E}[L_{i,\hat{S}}^1(i \in \hat{S})]}} \left(\sum_{i=1}^b \left[c_{\text{ov}} + \sum_{j \geq i} (c_j + c_j^\#) \right] p_i \right) \\ &= \frac{1}{\min_{i \in [b]} \frac{(\sum_{s=1}^i p_s)^2}{\sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^1}} \left(\sum_{i=1}^b \left[c_{\text{ov}} + \sum_{j \geq i} (c_j + c_j^\#) \right] p_i \right). \end{aligned}$$

Let us first focus on $\mathbb{E}[\text{cost}_{\varepsilon}(K)]$, following a similar reasoning to that in Section E.1.1. Denote $d_i := c_{\text{ov}} + \sum_{j \geq i} (c_j + c_j^\#)$ for $i \in [b]$ and consider the linear fractional program

$$\begin{aligned} \min_{p,t} \quad & \frac{d_1 p_1 + \dots + d_b p_b}{t} \\ \text{s.t.} \quad & p_1, \dots, p_b \geq 0 \\ & p_1 + \dots + p_b = 1 \\ & t \geq 0 \\ & t \leq \frac{\left(\sum_{s=1}^i p_s \right)^2}{\sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^1}, \quad i \in [b]. \end{aligned} \tag{37}$$

This program can be written equivalently as

$$\begin{aligned} \min_q \quad & d_1 q_1 + \dots + d_b q_b \\ \text{s.t.} \quad & q_1, \dots, q_b \geq 0 \\ & g_i(q) := \left(\sum_{s=1}^i q_s \right)^2 - \sum_{s=1}^i q_s L_{i,\{s,\dots,b\}}^1 \geq 0, \quad i \in [b]. \end{aligned} \tag{38}$$

We now write the KKT conditions for (38). Introduce multipliers $\lambda_i \geq 0$ for the constraints $g_i(q) \geq 0$ and multipliers $\eta_k \geq 0$ for the non-negativity constraints $q_k \geq 0$. The Lagrangian is

$$\mathcal{L}(q, \lambda, s) = \sum_{i=1}^b d_i q_i - \sum_{i=1}^b \lambda_i g_i(q) - \sum_{k=1}^b \eta_k q_k.$$

Fix $k \in [b]$. Differentiating g_i with respect to q_k yields

$$\frac{\partial g_i(q)}{\partial q_k} = \begin{cases} 2 \sum_{s=1}^i q_s - L_{i,\{k,\dots,b\}}^1, & k \leq i, \\ 0, & k > i. \end{cases} \tag{39}$$

Thus, stationarity $\nabla_q \mathcal{L}(q, \lambda, s) = 0$ yields, component-wise for $k \in [b]$,

$$0 = d_k - \sum_{i=k}^b \lambda_i \left(2 \sum_{s=1}^i q_s - L_{i, \{k, \dots, b\}}^1 \right) - \eta_k, \quad (40)$$

and complementarity and sign conditions are

$$\lambda_i g_i(q) = 0, \quad \eta_i q_i = 0, \quad \lambda_i, \eta_i \geq 0 \quad i \in [b].$$

Lemma E.2. Let $q^* = (q_1^*, \dots, q_p^*)$ be a local minimizer of (38) and let $\text{supp}(q^*) := \{k : q_k^* > 0\}$. Then, for every index $k \in \text{supp}(q^*)$, there exists at least one index i with $i \geq k$ such that $\lambda_i > 0$. In particular, not all λ_i can be zero.

Proof. Fix $k \in \text{supp}(q^*)$. If $\lambda_i = 0$ for all $i \geq k$, then (40) at k reduces to

$$0 = d_k - \eta_k.$$

By complementary slackness $\eta_k q_k^* = 0$, and since $q_k^* > 0$, we must have $\eta_k = 0$, so $d_k = 0$ as well. But by definition $d_k = c_{\text{ov}} + \sum_{j \geq k} (c_j + c_j^\#)$, which is (under the model assumptions on the costs) strictly positive. Hence there must exist $i \geq k$ with $\lambda_i > 0$. In particular not all λ_i are zero. $\square$

Lemma E.3. If $\text{supp}(q) = \{k\}$, then

$$q_k \geq \max_{i \in \{k, \dots, b\}} L_{i, \{k, \dots, b\}}^1.$$

Proof. When $q_j = 0$ for $j \neq k$, we have

$$\sum_{s=1}^i q_s = \begin{cases} 0 & i < k, \\ q_k & i \geq k. \end{cases}$$

For $i < k$, trivially $g_i(q) = 0$. For $i \geq k$, we have

$$g_i(q) = q_k^2 - q_k L_{i, \{k, \dots, b\}}^1 = q_k \left(q_k - L_{i, \{k, \dots, b\}}^1 \right).$$

Since $q_k > 0$, the constraint $g_i(q) \geq 0$ is equivalent to $q_k \geq L_{i, \{k, \dots, b\}}^1$. This must hold for every $i \geq k$, and hence q_k is at least the stated maximum. $\square$

Lemma E.4. Let q^* be a KKT point of (38) such that $\text{supp}(q^*) = \{k\}$. Then

$$q_k^* = \max_{i \geq k} L_{i, \{k, \dots, b\}}^1.$$

Proof. By Lemma E.3 we already have $q_k^* \geq \max_{i \geq k} L_{i, \{k, \dots, b\}}^1$. Assume that the inequality is strict, that is,

$$q_k^* > \max_{i \geq k} L_{i, \{k, \dots, b\}}^1.$$

Then for every $i \geq k$ we have

$$g_i(q^*) = q_k^* \left(q_k^* - L_{i, \{k, \dots, b\}}^1 \right) > 0.$$

Since by complementary slackness $\lambda_i g_i(q^*) = 0$, we get $\lambda_i = 0$ for all $i \geq k$. Plugging this into the stationarity condition (40) yields

$$0 = d_k - \eta_k.$$

But $q_k^* > 0$ forces $\eta_k = 0$, and thus $d_k = 0$, yielding a contradiction. Therefore, the strict inequality cannot hold, and we must have

$$q_k^* = \max_{i \geq k} L_{i, \{k, \dots, b\}}^1.$$

$\square$

Theorem E.2. Unless

$$L_{1,[b]}^1 = \max_{i \in [b]} L_{i,[b]}^1,$$

$(p_1, p_2, \dots, p_b) = (1, 0, \dots, 0)$ is *not* an optimal solution of (37).

Proof. According to Remark D.5, the sampling must be such that

$$\mathbb{P}(i \in \hat{S}) = \sum_{s=1}^i p_s > 0 \quad \forall i \in [b],$$

and hence we must have $p_1 > 0$. Thus, the support of the solution p^* of (37) (which coincides with the support of the solution q^* of (38)) can only be a singleton if $\text{supp}(p^*) = \text{supp}(q^*) = \{1\}$. But then, by Lemma E.4, we must have

$$q_1^* = \max_{i \in [b]} L_{i,[b]}^1, \quad q_2^* = \dots = q_b^* = 0,$$

which in turn implies that

$$0 \leq g_i(q^*) = \left(\sum_{s=1}^i q_s^* \right)^2 - \sum_{s=1}^i q_s^* L_{i,\{s,\dots,b\}}^1 = q_1^* (q_1^* - L_{i,[b]}^1)$$

for all $i \in [b]$, so in particular $q_1^* (q_1^* - L_{1,[b]}^1) \geq 0$. Now, suppose $L_{1,[b]}^1 \neq \max_{i \in [b]} L_{i,[b]}^1$. Then the inequality is strict and, by complementary slackness, $\lambda_1 = \eta_1 = 0$, so

$$d_1 = \sum_{i=2}^b \lambda_i (2q_1^* - L_{i,\{1,\dots,b\}}^1) \leq \sum_{i=2}^b \lambda_i (2q_1^* - L_{i,\{2,\dots,b\}}^1) = d_2 - \eta_2 \leq d_2,$$

which is a contradiction. Thus, similar to the result in Remark E.1, we must have $L_{1,[b]}^1 = \max_{i \in [b]} L_{i,[b]}^1$. $\square$

E.2.2 τ -Nice Sampling

Choose S uniformly from all subsets of $[b]$ of size τ .

For every $i \in [b]$ we have

$$\mathbb{P}(i \in \hat{S}) = \frac{\binom{b-1}{\tau-1}}{\binom{b}{\tau}} = \frac{\tau}{b}$$

and

$$\mathbb{E} \left[L_{i,\hat{S}}^\alpha \mathbb{I}(i \in \hat{S}) \right] = \frac{1}{\binom{b}{\tau}} \sum_{\substack{S \subseteq [b] \\ |S|=\tau}} L_{i,S}^\alpha \mathbb{I}(i \in S).$$

for $\alpha \in \{0, 1\}$. Recall that

$$\mathbb{E}[\text{cost}(S)] = c_{\text{ov}} + \sum_{j=1}^b c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}} \right) + \frac{\tau}{b} \sum_{j=1}^b c_j^\#.$$

Hence, substituting into (35) and (36), the objective functions to minimize are

$$\begin{aligned} & \mathbb{E}[\text{cost}_{\varepsilon^2}(K)] \\ & \propto \frac{\sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})^2 \mathbb{E}[L_{i,\hat{S}}^0 \mathbb{I}(i \in \hat{S})]}{(\mathbb{E}[L_{i,\hat{S}}^1 \mathbb{I}(i \in \hat{S})])^2}}{\min_{i \in [b]} \frac{\mathbb{P}(i \in \hat{S})^2}{(\mathbb{E}[L_{i,\hat{S}}^1 \mathbb{I}(i \in \hat{S})])^2}} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}} \right) + \frac{\tau}{b} \sum_{j=1}^b c_j^\# \right) \end{aligned}$$

$$\begin{aligned}
 &= \frac{\max_{i \in [b]} \left(\sum_{\substack{S \subseteq [b] \\ |S|=\tau}} L_{i,S}^1 \mathbb{I}(i \in S) \right)^2}{\frac{\tau}{b} \binom{b-1}{\tau-1}} \\
 &\quad \times \sum_{i=1}^b \frac{\sum_{\substack{S \subseteq [b] \\ |S|=\tau}} L_{i,S}^0 \mathbb{I}(i \in S)}{\left(\sum_{\substack{S \subseteq [b] \\ |S|=\tau}} L_{i,S}^1 \mathbb{I}(i \in S) \right)^2} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}} \right) + \frac{\tau}{b} \sum_{j=1}^b c_j^\# \right)
 \end{aligned}$$

and

$$\begin{aligned}
 \mathbb{E}[\text{cost}_\varepsilon(K)] &\propto \frac{1}{\min_{i \in [b]} \frac{\mathbb{P}(i \in \hat{S})^2}{\mathbb{E}[L_{i,S}^1 \mathbb{I}(i \in \hat{S})]}} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}} \right) + \frac{\tau}{b} \sum_{j=1}^b c_j^\# \right) \\
 &= \frac{\max_{i \in [b]} \sum_{\substack{S \subseteq [b] \\ |S|=\tau}} L_{i,S}^1 \mathbb{I}(i \in S)}{\left(\frac{\tau}{b} \right)^2 \binom{b}{\tau}} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}} \right) + \frac{\tau}{b} \sum_{j=1}^b c_j^\# \right).
 \end{aligned}$$

As in Section E.1.2, in general there is no closed-form expression for τ minimizing these costs, but it can be shown that $\tau = b$ need not be optimal. Let us make a simplifying assumption that $L_{i,S}^0$ depends only on i and $|S|$ and define $L_{i,\tau}^0 := L_{i,S}^0$ for $|S| = \tau$. Then

$$\begin{aligned}
 &\mathbb{E}[\text{cost}_{\varepsilon^2}(K)] \\
 &\propto \frac{\max_{i \in [b]} \left(\sum_{\substack{S \subseteq [b] \\ |S|=\tau}} L_{i,\tau}^1 \mathbb{I}(i \in S) \right)^2}{\frac{\tau}{b} \binom{b-1}{\tau-1}} \\
 &\quad \times \sum_{i=1}^b \frac{\sum_{\substack{S \subseteq [b] \\ |S|=\tau}} L_{i,\tau}^0 \mathbb{I}(i \in S)}{\left(\sum_{\substack{S \subseteq [b] \\ |S|=\tau}} L_{i,\tau}^1 \mathbb{I}(i \in S) \right)^2} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}} \right) + \frac{\tau}{b} \sum_{j=1}^b c_j^\# \right) \\
 &= \frac{\max_{i \in [b]} \left(L_{i,\tau}^1 \binom{b-1}{\tau-1} \right)^2}{\frac{\tau}{b} \binom{b-1}{\tau-1}} \sum_{i=1}^b \frac{L_{i,\tau}^0 \binom{b-1}{\tau-1}}{\left(L_{i,\tau}^1 \binom{b-1}{\tau-1} \right)^2} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}} \right) + \frac{\tau}{b} \sum_{j=1}^b c_j^\# \right) \\
 &= \underbrace{\max_{i \in [b]} (L_{i,\tau}^1)^2 \sum_{i=1}^b \frac{L_{i,\tau}^0}{(L_{i,\tau}^1)^2}}_{:=A_{\varepsilon^2}(\tau)} \underbrace{\left(\frac{b}{\tau} c_{\text{ov}} + \sum_{j=1}^b c_j \left(\frac{b}{\tau} - \frac{\binom{b-j}{\tau}}{\binom{b-1}{\tau-1}} \right) + \sum_{j=1}^b c_j^\# \right)}_{:=B(\tau)}
 \end{aligned}$$

and

$$\begin{aligned}
 \mathbb{E}[\text{cost}_\varepsilon(K)] &\propto \frac{\max_{i \in [b]} \sum_{\substack{S \subseteq [b] \\ |S|=\tau}} L_{i,\tau}^1 \mathbb{I}(i \in S)}{\left(\frac{\tau}{b} \right)^2 \binom{b}{\tau}} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}} \right) + \frac{\tau}{b} \sum_{j=1}^b c_j^\# \right) \\
 &= \frac{\max_{i \in [b]} L_{i,\tau}^1 \binom{b-1}{\tau-1}}{\left(\frac{\tau}{b} \right)^2 \binom{b}{\tau}} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}} \right) + \frac{\tau}{b} \sum_{j=1}^b c_j^\# \right) \\
 &= \underbrace{\max_{i \in [b]} L_{i,\tau}^1}_{:=A_\varepsilon(\tau)} \underbrace{\left(\frac{b}{\tau} c_{\text{ov}} + \sum_{j=1}^b c_j \left(\frac{b}{\tau} - \frac{\binom{b-j}{\tau}}{\binom{b-1}{\tau-1}} \right) + \sum_{j=1}^b c_j^\# \right)}_{:=B(\tau)}.
 \end{aligned}$$

We have already established in Section E.1.2 that B is decreasing in τ . In fact, the expression to be minimized there is structurally very similar to these obtained above. Specifically, in the smooth case we had $\text{cost}_\varepsilon(\mathcal{D}) \propto A(\tau)B(\tau)$, where $A(\tau) := \max_{i \in [b]} L_{i,\tau}^0$. Since $L_{i,\tau}^1$, like $L_{i,\tau}^0$, is non-decreasing in τ , the same reasoning as in Section E.1.2 applies to $\mathbb{E}[\text{cost}_\varepsilon(K)]$. On the other hand, the dependence of $A_{\varepsilon^2}(\tau)$ on τ can be arbitrary depending on the scaling between $L_{i,\tau}^0$ and $L_{i,\tau}^1$.

Consequently, both objectives $\mathbb{E}[\text{cost}_\varepsilon(K)]$ and $\mathbb{E}[\text{cost}_{\varepsilon^2}(K)]$ present a trade-off between a decreasing factor $B(\tau)$ and a (typically) increasing factor $A_1(\tau)$. If $A_\varepsilon(\tau)$ or $A_{\varepsilon^2}(\tau)$ grows sufficiently fast in τ , it can compensate for the decrease of $B(\tau)$ and make a smaller τ (or even $\tau = 1$) optimal.

E.2.3 τ -Submodel Sampling

Sample a starting index $\hat{s} \in \{1, \dots, b - \tau + 1\}$ with probability $p_i = \mathbb{P}(\hat{s} = i)$ and set $\hat{S} = \{\hat{s}, \dots, \hat{s} + \tau - 1\}$.

For any fixed $i \in [b]$, we have

$$\mathbb{P}(i \in \hat{S}) = \sum_{j=\max\{1, i-\tau+1\}}^{\min\{i, b-\tau+1\}} p_j$$

and

$$\mathbb{E}\left[L_{i,\hat{S}}^1 \mathbb{I}(i \in \hat{S})\right] = \sum_{j=\max\{1, i-\tau+1\}}^{\min\{i, b-\tau+1\}} p_j L_{i,\{j, \dots, j+\tau-1\}}^1.$$

We have also already shown that

$$\mathbb{E}[\text{cost}(\hat{S})] = c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{i=1}^{\min\{j, b-\tau+1\}} p_i \right) + \sum_{j=1}^b c_j^\# \left(\sum_{i=\max\{1, j-\tau+1\}}^{\min\{j, b-\tau+1\}} p_i \right).$$

Partitioned τ -submodel sampling. Mimicking Section E.1.3, let us consider a special case of the above sampling scheme where the submodels partition $[b]$. For simplicity, we assume that b is divisible by τ and let $m = b/\tau$. The algorithm is then equivalent to τ -submodel sampling with starting-index distribution satisfying

$$p_{s_k} > 0 \quad (k = 1, \dots, m), \quad p_j = 0 \text{ otherwise,}$$

where $s_k = (k-1)\tau + 1$, $k \in [m]$. For any fixed $i \in [b]$, the derivations above simplify to

$$\mathbb{P}(i \in \hat{S}) = \sum_{j=\max\{1, i-\tau+1\}}^{\min\{i, b-\tau+1\}} p_j = p_{s_{\lceil i/\tau \rceil}}$$

and

$$\mathbb{E}\left[L_{i,\hat{S}}^\alpha \mathbb{I}(i \in \hat{S})\right] = \sum_{j=\max\{1, i-\tau+1\}}^{\min\{i, b-\tau+1\}} p_j L_{i,\{j, \dots, j+\tau-1\}}^\alpha = p_{s_{\lceil i/\tau \rceil}} L_{i, B_{\lceil i/\tau \rceil}}^\alpha$$

for $\alpha \in \{0, 1\}$. Plugging this choice into (35) and (36) gives

$$\begin{aligned} \mathbb{E}[\text{cost}_{\varepsilon^2}(K)] &\propto \frac{\sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})^2 \mathbb{E}[L_{i,\hat{S}}^0 \mathbb{I}(i \in \hat{S})]}{(\mathbb{E}[L_{i,\hat{S}}^1 \mathbb{I}(i \in \hat{S})])^2}}{\min_{i \in [b]} \left(\frac{\mathbb{P}(i \in \hat{S})^2}{\mathbb{E}[L_{i,\hat{S}}^1 \mathbb{I}(i \in \hat{S})]} \right)^2} \\ &\quad \times \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{i=1}^{\min\{j, b-\tau+1\}} p_i \right) + \sum_{j=1}^b c_j^\# \left(\sum_{i=\max\{1, j-\tau+1\}}^{\min\{j, b-\tau+1\}} p_i \right) \right) \\ &= \frac{\sum_{i=1}^b \frac{p_{s_{\lceil i/\tau \rceil}}^2 L_{i, B_{\lceil i/\tau \rceil}}^0}{\left(L_{i, B_{\lceil i/\tau \rceil}}^1 \right)^2}}{\min_{i \in [b]} \left(\frac{p_{s_{\lceil i/\tau \rceil}}}{L_{i, B_{\lceil i/\tau \rceil}}^1} \right)^2} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{k=1}^{\lceil j/\tau \rceil} p_{s_k} \right) + \sum_{j=1}^b c_j^\# p_{s_{\lceil j/\tau \rceil}} \right) \end{aligned}$$

and

$$\begin{aligned} \mathbb{E}[\text{cost}_\varepsilon(K)] &\propto \frac{1}{\min_{i \in [b]} \frac{\mathbb{P}(i \in \hat{S})^2}{\mathbb{E}[L_{i,\hat{S}}^1 \mathbb{I}(i \in \hat{S})]}} \\ &\quad \times \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{i=1}^{\min\{j, b-\tau+1\}} p_i \right) + \sum_{j=1}^b c_j^\# \left(\sum_{i=\max\{1, j-\tau+1\}}^{\min\{j, b-\tau+1\}} p_i \right) \right) \\ &= \frac{1}{\min_{i \in [b]} \frac{p_{s_{\lceil i/\tau \rceil}}}{L_{i, B_{\lceil i/\tau \rceil}}^1}} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{k=1}^{\lceil j/\tau \rceil} p_{s_k} \right) + \sum_{j=1}^b c_j^\# p_{s_{\lceil j/\tau \rceil}} \right). \end{aligned}$$

The expression for $\mathbb{E}[\text{cost}_\varepsilon(K)]$ is entirely analogous to (25) derived in the smooth case, with $L_{i, B_{\lceil i/\tau \rceil}}^0$ replaced by $L_{i, B_{\lceil i/\tau \rceil}}^1$. Consequently, for a fixed τ , the optimal block probabilities are

$$p_{s_k}^* = \frac{\max_{i \in B_k} L_{i, B_k}^1}{\sum_{l=1}^m \max_{i \in B_l} L_{i, B_l}^1},$$

Again, the optimal number of blocks m depends on the growth rate of the constants $L_{i, \tau}^1$ with respect to τ and on their interaction with the costs c_j and $c_j^\#$. In general, the best choice of m is not necessarily b .

The dependence of $\mathbb{E}[\text{cost}_{\varepsilon^2}(K)]$ on τ is significantly more complex and governed by the relative scaling between $L_{i, \tau}^0$ and $L_{i, \tau}^1$.

E.2.4 Arbitrary Submodel Sampling

Let $\{B_1, \dots, B_m\}$ be a partition of $[b]$. Set $\hat{S} = B_i$ with probability p_i (where $\sum_{i=1}^m p_i = 1$). For $i \in [b]$, $k(i)$ denotes the unique block with $i \in B_{k(i)}$ and $\underline{b}_k := \min B_k$.

For any $i \in [b]$, we have

$$\mathbb{P}(i \in \hat{S}) = p_{k(i)}$$

and

$$\mathbb{E}[L_{i, \hat{S}}^\alpha \mathbb{I}(i \in \hat{S})] = p_{k(i)} L_{i, B_{k(i)}}^\alpha$$

for $\alpha \in \{0, 1\}$. Since

$$\mathbb{E}[\text{cost}(\hat{S})] = c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{k: \underline{b}_k \leq j} p_k \right) + \sum_{j=1}^b c_j^\# p_{k(j)},$$

the total expected costs become

$$\begin{aligned} \mathbb{E}[\text{cost}_{\varepsilon^2}(K)] &\propto \frac{\sum_{i=1}^b \frac{\mathbb{P}(i \in \hat{S})^2 \mathbb{E}[L_{i, \hat{S}}^0 \mathbb{I}(i \in \hat{S})]}{\left(\mathbb{E}[L_{i, \hat{S}}^1 \mathbb{I}(i \in \hat{S})] \right)^2}}{\min_{i \in [b]} \left(\frac{\mathbb{P}(i \in \hat{S})^2}{\mathbb{E}[L_{i, \hat{S}}^1 \mathbb{I}(i \in \hat{S})]} \right)^2} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{k: \underline{b}_k \leq j} p_k \right) + \sum_{j=1}^b c_j^\# p_{k(j)} \right) \\ &= \frac{\sum_{i=1}^b \frac{p_{k(i)} L_{i, B_{k(i)}}^0}{\left(L_{i, B_{k(i)}}^1 \right)^2}}{\min_{i \in [b]} \left(\frac{p_{k(i)}}{L_{i, B_{k(i)}}^1} \right)^2} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{k: \underline{b}_k \leq j} p_k \right) + \sum_{j=1}^b c_j^\# p_{k(j)} \right) \end{aligned}$$

and

$$\mathbb{E}[\text{cost}_\varepsilon(K)] \propto \frac{1}{\min_{i \in [b]} \frac{\mathbb{P}(i \in \hat{S})^2}{\mathbb{E}[L_{i, \hat{S}}^1 \mathbb{I}(i \in \hat{S})]}} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{k: \underline{b}_k \leq j} p_k \right) + \sum_{j=1}^b c_j^\# p_{k(j)} \right)$$

$$= \frac{1}{\min_{i \in [b]} \frac{p_{k(i)}}{L_{i, B_{k(i)}}^1}} \left(c_{\text{ov}} + \sum_{j=1}^b c_j \left(\sum_{k: b_k \leq j} p_k \right) + \sum_{j=1}^b c_j^\# p_{k(j)} \right).$$

Following the reasoning in Section E.1.4, we find that for a fixed partition, the choice of probabilities minimizing $\mathbb{E}[\text{cost}_\varepsilon(K)]$ is

$$p_k^\star = \frac{1/\delta_k^{\min}}{\sum_{l=1}^m 1/\delta_l^{\min}} = \frac{\max_{i \in B_k} L_{i, B_k}^0}{\sum_{l=1}^m \max_{i \in B_l} L_{i, B_l}^0},$$

and that submodel training can improve upon full model training in certain regimes.

F Convergence Results – Stochastic Gradient Case

The proof of Theorem F.1 relies on two preliminary lemmas, which we state and prove first.

Lemma F.1 (Descent Lemma). Let Assumption 4.4 hold and consider the update rule $X_i^{k+1} = \text{LMO}_{\mathcal{B}(X_i^k, t_i^k)}(M_i^k)$, $i = 1, \dots, b$, where $X^{k+1} = [X_1^{k+1}, \dots, X_b^{k+1}] \in \mathcal{X}$, $X^k = [X_1^k, \dots, X_b^k] \in \mathcal{X}$, $M^k = [M_1^k, \dots, M_b^k] \in \mathcal{X}$ and $t_i^k > 0$. Then

$$\begin{aligned} f(X^{k+1}) &\leq f(X^k) + \sum_{i \in S^k} 2t_i^k \|\nabla_i f(X^k) - M_i^k\|_{(i)\star} - \sum_{i \in S^k} t_i^k \|\nabla_i f(X^k)\|_{(i)\star} \\ &\quad + \sum_{i \in S^k} \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star}}{2} (t_i^k)^2. \end{aligned}$$

Proof. By Lemma H.3

$$\begin{aligned} &f(X^{k+1}) \\ &\leq f(X^k) + \langle \nabla f(X^k), X^{k+1} - X^k \rangle + \sum_{i \in S^k} \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star}}{2} \|X_i^k - X_i^{k+1}\|_{(i)}^2 \\ &\stackrel{(48)}{=} f(X^k) + \sum_{i \in S^k} \left(\langle \nabla_i f(X^k) - M_i^k, X_i^{k+1} - X_i^k \rangle_{(i)} + \langle M_i^k, X_i^{k+1} - X_i^k \rangle_{(i)} \right) \\ &\quad + \sum_{i \in S^k} \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star}}{2} (t_i^k)^2 \\ &\stackrel{(49)}{=} f(X^k) + \sum_{i \in S^k} \left(\langle \nabla_i f(X^k) - M_i^k, X_i^{k+1} - X_i^k \rangle_{(i)} - t_i^k \|M_i^k\|_{(i)\star} \right) \\ &\quad + \sum_{i \in S^k} \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star}}{2} (t_i^k)^2 \\ &\leq f(X^k) + \sum_{i \in S^k} \left(t_i^k \|\nabla_i f(X^k) - M_i^k\|_{(i)\star} - t_i^k \|M_i^k\|_{(i)\star} \right. \\ &\quad \left. + \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star}}{2} (t_i^k)^2 \right), \end{aligned}$$

where in the last line we used the Cauchy-Schwarz inequality. Therefore, using triangle inequality, we get

$$\begin{aligned} &f(X^{k+1}) \\ &\leq f(X^k) + \sum_{i \in S^k} \left(t_i^k \|\nabla_i f(X^k) - M_i^k\|_{(i)\star} + t_i^k \|\nabla_i f(X^k) - M_i^k\|_{(i)\star} - t_i^k \|\nabla_i f(X^k)\|_{(i)\star} \right) \\ &\quad + \sum_{i \in S^k} \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star}}{2} (t_i^k)^2 \\ &= f(X^k) + \sum_{i \in S^k} \left(2t_i^k \|\nabla_i f(X^k) - M_i^k\|_{(i)\star} - t_i^k \|\nabla_i f(X^k)\|_{(i)\star} \right) \\ &\quad + \sum_{i \in S^k} \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \|\nabla_i f(X^k)\|_{(i)\star}}{2} (t_i^k)^2 \end{aligned}$$

as required. $\square$

Lemma F.2. Let Assumptions 4.4 and 4.5 hold. Then, the iterates of Algorithm 1 run with $t_i^k \equiv t_i$ satisfy

$$\begin{aligned} & \mathbb{E} [\|M_i^{k+1} - \nabla_i f(X^{k+1})\|_2] \\ & \leq \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i\right)^{k+1} \mathbb{E} [\|M_i^0 - \nabla_i f(X^0)\|_2] + \frac{t_i \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i\right) L_{i,\hat{S}}^0 \right]}{\rho_i \beta_i \mathbb{E} [\mathbb{I}(i \in \hat{S})]} \\ & \quad + \frac{t_i}{\rho_i} \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i\right) L_{i,\hat{S}}^1 \right] \sum_{l=0}^k \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i\right)^{k-l} \mathbb{E} [\|\nabla_i f(X^l)\|_{(i)\star}] \\ & \quad + \sigma_i \sqrt{\beta_i}. \end{aligned}$$

Proof. The proof is inspired by Cutkosky and Mehta [2020, Theorem 1]. First, using the momentum update rule, we have

$$\begin{aligned} M_i^{k+1} &= (1 - \mathbb{I}(i \in S^k) \beta_i) M_i^k + \mathbb{I}(i \in S^k) \beta_i \nabla_i f(X^{k+1}; \xi^{k+1}) \\ &= (1 - \beta_i^k) (M_i^k - \nabla_i f(X^k)) + (1 - \beta_i^k) (\nabla_i f(X^k) - \nabla_i f(X^{k+1})) \\ & \quad + \beta_i^k (\nabla_i f(X^{k+1}; \xi^{k+1}) - \nabla_i f(X^{k+1})) + \nabla_i f(X^{k+1}). \end{aligned}$$

where $\beta_i^k := \mathbb{I}(i \in S^k) \beta_i$. To simplify the notation, define $U_{1,i}^k := M_i^k - \nabla_i f(X^k)$, $U_{2,i}^k := \nabla_i f(X^k) - \nabla_i f(X^{k+1})$ and $U_{3,i}^k := \nabla_i f(X^k; \xi^k) - \nabla_i f(X^k)$. Then the above can be written as

$$U_{1,i}^{k+1} = (1 - \beta_i^k) U_{1,i}^k + (1 - \beta_i^k) U_{2,i}^k + \beta_i^k U_{3,i}^{k+1}. \quad (41)$$

Unrolling the recursion in (41), we get

$$\begin{aligned} U_{1,i}^{k+1} &= (1 - \beta_i^k) U_{1,i}^k + (1 - \beta_i^k) U_{2,i}^k + \beta_i^k U_{3,i}^{k+1} \\ &= \left(\prod_{m=0}^k (1 - \beta_i^m) \right) U_{1,i}^0 + \sum_{l=0}^k \left(\prod_{m=l}^k (1 - \beta_i^m) \right) U_{2,i}^l + \sum_{l=0}^k \left(\beta_i^l \prod_{m=l+1}^k (1 - \beta_i^m) \right) U_{3,i}^{l+1}, \end{aligned}$$

where by convention $\prod_{m=a}^b (\cdot) = 1$ if $a > b$. Hence, taking norms and using the triangle inequality,

$$\begin{aligned} \mathbb{E} [\|U_{1,i}^{k+1}\|_2] &\leq \mathbb{E} \left[\left\| \left(\prod_{m=0}^k (1 - \beta_i^m) \right) U_{1,i}^0 \right\|_2 \right] + \mathbb{E} \left[\left\| \sum_{l=0}^k \left(\prod_{m=l}^k (1 - \beta_i^m) \right) U_{2,i}^l \right\|_2 \right] \\ &\quad + \mathbb{E} \left[\left\| \sum_{l=0}^k \left(\beta_i^l \prod_{m=l+1}^k (1 - \beta_i^m) \right) U_{3,i}^{l+1} \right\|_2 \right]. \end{aligned} \quad (42)$$

Let us consider each of the terms separately. First, using the independence of $\{S^k\}_{k \geq 0}$, we have

$$\begin{aligned} \mathbb{E} \left[\left\| \left(\prod_{m=0}^k (1 - \beta_i^m) \right) U_{1,i}^0 \right\|_2 \right] &\leq \mathbb{E} \left[\left\| \left(\prod_{m=0}^k (1 - \beta_i^m) \right) \|U_{1,i}^0\|_2 \right\| \right] \\ &= \mathbb{E} \left[\mathbb{E} \left[\left\| \left(\prod_{m=0}^k (1 - \beta_i^m) \right) \|U_{1,i}^0\|_2 \right\| \middle| X^k, M_i^k \right] \right] \\ &= \prod_{m=0}^k \mathbb{E} [1 - \beta_i^m] \mathbb{E} [\|U_{1,i}^0\|_2]. \end{aligned} \quad (43)$$

Next, by Assumption 4.4

$$\begin{aligned} & \mathbb{E} \left[\left\| \sum_{l=0}^k \left(\prod_{m=l}^k (1 - \beta_i^m) \right) U_{2,i}^l \right\|_2 \right] \\ & \leq \mathbb{E} \left[\sum_{l=0}^k \left(\prod_{m=l}^k (1 - \beta_i^m) \right) \|U_{2,i}^l\|_2 \right] \end{aligned}$$

$$\begin{aligned}
 &\leq \frac{1}{\underline{\rho}_i} \sum_{l=0}^k \mathbb{E} \left[\left(\prod_{m=l}^k (1 - \beta_i^m) \right) \|\nabla_i f(X^l) - \nabla_i f(X^{l+1})\|_{(i)\star} \right] \\
 &\stackrel{(4.3)}{\leq} \frac{1}{\underline{\rho}_i} \sum_{l=0}^k \mathbb{E} \left[\left(\prod_{m=l}^k (1 - \beta_i^m) \right) \left(L_{i,S^l}^0 + L_{i,S^l}^1 \|\nabla_i f(X^l)\|_{(i)\star} \right) \|X_i^l - X_i^{l+1}\|_{(i)} \right].
 \end{aligned}$$

Since the product $\prod_{m=l+1}^k (1 - \beta_i^m)$ depends only on samplings at iterations $> l$, these factors are independent of the σ -algebra generated by (X^l, S^l) . Therefore,

$$\begin{aligned}
 &\mathbb{E} \left[\left\| \sum_{l=0}^k \left(\prod_{m=l}^k (1 - \beta_i^m) \right) U_{2,i}^l \right\|_2 \right] \\
 &\leq \frac{1}{\underline{\rho}_i} \sum_{l=0}^k \left(\left(\prod_{m=l+1}^k \mathbb{E} [1 - \beta_i^m] \right) \mathbb{E} \left[(1 - \beta_i^l) \left(L_{i,S^l}^0 + L_{i,S^l}^1 \|\nabla_i f(X^l)\|_{(i)\star} \right) t_i \right] \right),
 \end{aligned}$$

where

$$\begin{aligned}
 &\mathbb{E} \left[(1 - \beta_i^l) \left(L_{i,S^l}^0 + L_{i,S^l}^1 \|\nabla_i f(X^l)\|_{(i)\star} \right) \right] \\
 &= \mathbb{E} \left[(1 - \beta_i^l) L_{i,S^l}^0 \right] + \mathbb{E} \left[\mathbb{E} \left[(1 - \beta_i^l) L_{i,S^l}^1 \|\nabla_i f(X^l)\|_{(i)\star} \mid X^l \right] \right] \\
 &= \mathbb{E} \left[(1 - \beta_i^l) L_{i,S^l}^0 \right] + \mathbb{E} \left[(1 - \beta_i^l) L_{i,S^l}^1 \right] \mathbb{E} \left[\|\nabla_i f(X^l)\|_{(i)\star} \right].
 \end{aligned}$$

Thus

$$\begin{aligned}
 &\mathbb{E} \left[\left\| \sum_{l=0}^k \left(\prod_{m=l}^k (1 - \beta_i^m) \right) U_{2,i}^l \right\|_2 \right] \\
 &\leq \frac{t_i}{\underline{\rho}_i} \sum_{l=0}^k \left(\prod_{m=l+1}^k \mathbb{E} [1 - \beta_i^m] \right) \left(\mathbb{E} \left[(1 - \beta_i^l) L_{i,S^l}^0 \right] + \mathbb{E} \left[(1 - \beta_i^l) L_{i,S^l}^1 \right] \mathbb{E} \left[\|\nabla_i f(X^l)\|_{(i)\star} \right] \right).
 \end{aligned} \tag{44}$$

Lastly, by Jensen's inequality, the last term can be bounded via

$$\begin{aligned}
 &\mathbb{E} \left[\left\| \sum_{l=0}^k \left(\beta_i^l \prod_{m=l+1}^k (1 - \beta_i^m) \right) U_{3,i}^{l+1} \right\|_2 \right] \\
 &\leq \sqrt{\mathbb{E} \left[\left\| \sum_{l=0}^k \underbrace{\left(\beta_i^l \prod_{m=l+1}^k (1 - \beta_i^m) \right)}_{:=a_l} U_{3,i}^{l+1} \right\|_2^2 \right]} = \sqrt{\mathbb{E} \left[\sum_{l,r=0}^k a_l a_r \langle U_{3,i}^{l+1}, U_{3,i}^{r+1} \rangle \right]} \\
 &\stackrel{(4.5)}{=} \sqrt{\mathbb{E} \left[\sum_{l=0}^k a_l^2 \|U_{3,i}^{l+1}\|_2^2 \right]} = \sqrt{\sum_{l=0}^k \mathbb{E} \left[\mathbb{E} \left[a_l^2 \|U_{3,i}^{l+1}\|_2^2 \mid \{S^r\}_{r=l}^k, X^{l+1} \right] \right]} \\
 &= \sqrt{\sum_{l=0}^k \mathbb{E} \left[a_l^2 \mathbb{E} \left[\|U_{3,i}^{l+1}\|_2^2 \mid \{S^r\}_{r=l}^k, X^{l+1} \right] \right]} \stackrel{(4.5)}{\leq} \sigma_i \sqrt{\sum_{l=0}^k \mathbb{E} [a_l^2]} \\
 &= \sigma_i \sqrt{\sum_{l=0}^k \left(\mathbb{E} [(\beta_i^l)^2] \prod_{m=l+1}^k \mathbb{E} [(1 - \beta_i^m)^2] \right)} \\
 &= \sigma_i \sqrt{\sum_{l=0}^k \left(\mathbb{E} [\mathbb{I}(i \in S^l) \beta_i^2] \prod_{m=l+1}^k \mathbb{E} [(1 - \mathbb{I}(i \in S^m) \beta_i)^2] \right)}
 \end{aligned}$$

$$\begin{aligned}
 &= \sigma_i \sqrt{\sum_{l=0}^k \left(\beta_i^2 \mathbb{E} [\mathbb{I}(i \in \hat{S})] \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i \right)^2 \right]^{k-l} \right)} \\
 &\leq \sigma_i \sqrt{\frac{\beta_i^2 \mathbb{E} [\mathbb{I}(i \in \hat{S})]}{1 - \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i \right)^2 \right]}} \\
 &= \sigma_i \beta_i \sqrt{\frac{\mathbb{E} [\mathbb{I}(i \in \hat{S})]}{\mathbb{E} \left[(2 - \beta_i) \mathbb{I}(i \in \hat{S}) \beta_i \right]}} = \frac{\sigma_i \beta_i}{\sqrt{(2 - \beta_i) \beta_i}} \leq \sigma_i \sqrt{\beta_i}.
 \end{aligned} \tag{45}$$

Applying (43), (44) and (45) in (42) and noting that $\mathbb{E} [\beta_i^k] = \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i$ yields

$$\begin{aligned}
 &\mathbb{E} [\|U_{1,i}^{k+1}\|_2] \\
 &\leq \prod_{m=0}^k \mathbb{E} [1 - \beta_i^m] \mathbb{E} [\|U_{1,i}^0\|_2] + \frac{t_i}{\rho_i} \sum_{l=0}^k \left(\prod_{m=l+1}^k \mathbb{E} [1 - \beta_i^m] \right) \mathbb{E} [(1 - \beta_i^l) L_{i,S^l}^0] \\
 &\quad + \frac{t_i}{\rho_i} \sum_{l=0}^k \left(\prod_{m=l+1}^k \mathbb{E} [1 - \beta_i^m] \right) \mathbb{E} [(1 - \beta_i^l) L_{i,S^l}^1] \mathbb{E} [\|\nabla_i f(X^l)\|_{(i)\star}] + \sigma_i \sqrt{\beta_i} \\
 &= \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i \right)^{k+1} \mathbb{E} [\|U_{1,i}^0\|_2] \\
 &\quad + \frac{t_i}{\rho_i} \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i \right) L_{i,\hat{S}}^0 \right] \sum_{l=0}^k \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i \right)^{k-l} \\
 &\quad + \frac{t_i}{\rho_i} \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i \right) L_{i,\hat{S}}^1 \right] \sum_{l=0}^k \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i \right)^{k-l} \mathbb{E} [\|\nabla_i f(X^l)\|_{(i)\star}] \\
 &\quad + \sigma_i \sqrt{\beta_i} \\
 &\leq \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i \right)^{k+1} \mathbb{E} [\|U_{1,i}^0\|_2] + \frac{t_i}{\rho_i \beta_i \mathbb{E} [\mathbb{I}(i \in \hat{S})]} \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i \right) L_{i,\hat{S}}^0 \right] \\
 &\quad + \frac{t_i}{\rho_i} \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i \right) L_{i,\hat{S}}^1 \right] \sum_{l=0}^k \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i \right)^{k-l} \mathbb{E} [\|\nabla_i f(X^l)\|_{(i)\star}] \\
 &\quad + \sigma_i \sqrt{\beta_i}.
 \end{aligned}$$

□

Theorem F.1. Let Assumptions 4.1, 4.4, and 4.5 hold. Let $\{X^k\}_{k=0}^{K-1}$, $K \geq 1$, be the iterates of Algorithm 1 initialized with $M_i^0 = \nabla_i f(X^0; \xi^0)$ and run with $\beta_i \equiv \beta = 1/(K+1)^{1/2}$ and

$$0 < t_i^k \equiv t_i = \frac{\eta_i}{(K+1)^{3/4}}, \quad i = 1, \dots, b,$$

where $\eta_i^2 \leq \min \left\{ \frac{(K+1)^{1/2}}{4 \mathbb{E} [\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1] \mathbb{E} [\max_{i \in [b]} L_{i,\hat{S}}^1]}, \frac{\rho_i \min_{i \in [b]} (\mathbb{E} [\mathbb{I}(i \in \hat{S})])}{16 \bar{\rho}_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] \mathbb{E} [(1 - \mathbb{I}(i \in \hat{S}) \beta_i) L_{i,\hat{S}}^1] \mathbb{E} [\max_{i \in [b]} L_{i,\hat{S}}^1]}, 1 \right\}$. Then

$$\min_{k=0, \dots, K} \sum_{i=1}^b \frac{\mathbb{E} [\mathbb{I}(i \in \hat{S})] \eta_i}{\frac{1}{b} \sum_{l=1}^b \mathbb{E} [\mathbb{I}(l \in \hat{S})] \eta_l} \mathbb{E} [\|\nabla_i f(X^k)\|_{(i)\star}]$$

$$\begin{aligned}
 &\leq \frac{3\delta^0}{(K+1)^{1/4} \frac{1}{b} \sum_{l=1}^b \mathbb{E} [\mathbb{I}(l \in \hat{S})] \eta_l} + \frac{6}{(K+1)^{1/2} \sum_{i=1}^b \frac{1}{b} \sum_{l=1}^b \mathbb{E} [\mathbb{I}(l \in \hat{S})] \eta_l} \frac{\eta_i \bar{\rho}_i \sigma_i}{\sum_{l=1}^b \mathbb{E} [\mathbb{I}(l \in \hat{S})] \eta_l} \\
 &\quad + \sum_{i=1}^b \frac{\eta_i^2}{(K+1)^{1/4} \frac{1}{b} \sum_{l=1}^b \mathbb{E} [\mathbb{I}(l \in \hat{S})] \eta_l} \frac{2\bar{\rho}_i}{\rho_i} \left(\mathbb{E} [L_{i,\hat{S}}^0] + \mathbb{E} [L_{i,\hat{S}}^1] \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \right) \\
 &\quad + \sum_{i=1}^b \frac{\eta_i^2}{2(K+1)^{3/4} \frac{1}{b} \sum_{l=1}^b \mathbb{E} [\mathbb{I}(l \in \hat{S})] \eta_l} \\
 &\quad \quad \times \left(\mathbb{E} [\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^0] + \mathbb{E} [\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1] \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \right) \\
 &\quad + \sum_{i=1}^b \frac{2\mathbb{E} [\mathbb{I}(i \in \hat{S})] \eta_i}{(K+1)^{1/4} \frac{1}{b} \sum_{l=1}^b \mathbb{E} [\mathbb{I}(l \in \hat{S})] \eta_l} \bar{\rho}_i \sigma_i.
 \end{aligned}$$

Remark F.2. For Theorem F.1 to be meaningful, the sampling must be such that $\mathbb{E} [\mathbb{I}(i \in \hat{S})] \eta_i > 0$ for every $i \in [b]$. Thus, every layer has to be sampled with positive probability.

Remark F.3. Note that under Assumption 4.4, without loss of generality we can set $L_{i,S}^0 = L_{i,S}^1 = 0$ whenever $i \notin S$. Hence, in the case of RPT (see Section 4), we have

$$\begin{aligned}
 \mathbb{E} [\mathbb{I}(i \in \hat{S})] &= \sum_{s=1}^i p_s, \\
 \mathbb{E} \left[\max_{i \in [b]} L_{i,\hat{S}}^1 \right] &= \sum_{s=1}^b p_s \max_{i \in [b]} L_{i,\{s,\dots,b\}}^1, \\
 \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] &= \sum_{s=1}^b p_s \frac{L_{i,\{s,\dots,b\}}^0}{L_{i,\{s,\dots,b\}}^1} = \sum_{s=1}^i p_s \frac{L_{i,\{s,\dots,b\}}^0}{L_{i,\{s,\dots,b\}}^1}, \\
 \mathbb{E} \left[(1 - \mathbb{I}(i \in \hat{S})) \beta_i L_{i,\hat{S}}^1 \right] &= \sum_{s=1}^b p_s (1 - \mathbb{I}(i \in \{s,\dots,b\})) \beta_i L_{i,\{s,\dots,b\}}^1 \\
 &= (1 - \beta_i) \sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^1, \\
 \mathbb{E} [L_{i,\hat{S}}^\alpha] &= \sum_{s=1}^b p_s L_{i,\{s,\dots,b\}}^\alpha = \sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^\alpha, \\
 \mathbb{E} [L_{i,\hat{S}}^\alpha \mathbb{I}(i \in \hat{S})] &= \sum_{s=1}^i p_s L_{i,\{s,\dots,b\}}^\alpha
 \end{aligned}$$

for $\alpha \in \{0, 1\}$. Substituting these expressions into the rate yields the result in Theorem 4.4.

Proof of Theorem F.1. Taking expectation conditional on $[X^k, \{M_i^k\}_{i \in [b]}]$ in Lemma F.1 gives

$$\begin{aligned}
 &\mathbb{E} [f(X^{k+1}) | X^k, \{M_i^k\}_{i \in [b]}] \\
 &\leq f(X^k) + \mathbb{E} \left[\sum_{i \in S^k} 2t_i \|\nabla_i f(X^k) - M_i^k\|_{(i)\star} - \sum_{i \in S^k} t_i \|\nabla_i f(X^k)\|_{(i)\star} \middle| X^k, \{M_i^k\}_{i \in [b]} \right] \\
 &\quad + \mathbb{E} \left[\sum_{i \in S^k} \frac{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X^k)\|_{(i)\star}^2}{2} t_i^2 \middle| X^k, \{M_i^k\}_{i \in [b]} \right]
 \end{aligned}$$

$$\begin{aligned}
 &= f(X^k) + \sum_{i=1}^b \mathbb{E} \left[\mathbb{I}(i \in S^k) \left(2t_i \|\nabla_i f(X^k) - M_i^k\|_{(i)\star} - t_i \|\nabla_i f(X^k)\|_{(i)\star} \right) \middle| X^k, \{M_i^k\}_{i \in [b]} \right] \\
 &\quad + \sum_{i=1}^b \mathbb{E} \left[\mathbb{I}(i \in S^k) \frac{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X^k)\|_{(i)\star}}{2} t_i^2 \middle| X^k, \{M_i^k\}_{i \in [b]} \right] \\
 &\leq f(X^k) + \sum_{i=1}^b \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \left(2t_i \bar{\rho}_i \|\nabla_i f(X^k) - M_i^k\|_2 - t_i \|\nabla_i f(X^k)\|_{(i)\star} \right) \\
 &\quad + \frac{1}{2} \sum_{i=1}^b \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^0 \right] t_i^2 + \frac{1}{2} \sum_{i=1}^b \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] t_i^2 \|\nabla_i f(X^k)\|_{(i)\star}.
 \end{aligned}$$

Hence, taking full expectation

$$\begin{aligned}
 &\mathbb{E} [f(X^{k+1})] \\
 &\leq \mathbb{E} [f(X^k)] + \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] \mathbb{E} [\|\nabla_i f(X^k) - M_i^k\|_2] \\
 &\quad - \sum_{i=1}^b \mathbb{E} [\mathbb{I}(i \in \hat{S})] t_i \mathbb{E} [\|\nabla_i f(X^k)\|_{(i)\star}] \\
 &\quad + \frac{1}{2} \sum_{i=1}^b \mathbb{E} [\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^0] t_i^2 + \frac{1}{2} \sum_{i=1}^b \mathbb{E} [\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1] t_i^2 \mathbb{E} [\|\nabla_i f(X^k)\|_{(i)\star}].
 \end{aligned}$$

To simplify the notation, let $\delta^k := \mathbb{E} [f(X^k) - f^*]$ and $P_i^k := \mathbb{E} [\|\nabla_i f(X^k) - M_i^k\|_2]$. Then, according to the descent inequality above and Lemma F.2

$$\begin{aligned}
 \delta^{k+1} &\leq \delta^k - \sum_{i=1}^b t_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] \mathbb{E} [\|\nabla_i f(X^k)\|_{(i)\star}] + \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] P_i^k \\
 &\quad + \frac{1}{2} \sum_{i=1}^b t_i^2 \mathbb{E} [\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^0] + \frac{1}{2} \sum_{i=1}^b t_i^2 \mathbb{E} [\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1] \mathbb{E} [\|\nabla_i f(X^k)\|_{(i)\star}], \tag{46}
 \end{aligned}$$

$$\begin{aligned}
 P_i^k &\leq \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i \right)^k P_i^0 + \frac{t_i \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i \right) L_{i,\hat{S}}^0 \right]}{\rho_i \beta_i \mathbb{E} [\mathbb{I}(i \in \hat{S})]} \\
 &\quad + \frac{t_i}{\rho_i} \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i \right) L_{i,\hat{S}}^1 \right] \sum_{l=0}^{k-1} \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i \right)^{k-1-l} \mathbb{E} [\|\nabla_i f(X^l)\|_{(i)\star}] \\
 &\quad + \sigma_i \sqrt{\beta_i}. \tag{47}
 \end{aligned}$$

Applying (47) in (46),

$$\begin{aligned}
 &\delta^{k+1} \\
 &\leq \delta^k - \sum_{i=1}^b t_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] \mathbb{E} [\|\nabla_i f(X^k)\|_{(i)\star}] \\
 &\quad + \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] \left(\left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i \right)^k P_i^0 + \frac{t_i \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i \right) L_{i,\hat{S}}^0 \right]}{\rho_i \beta_i \mathbb{E} [\mathbb{I}(i \in \hat{S})]} \right) \\
 &\quad + \sum_{i=1}^b \left(\frac{2t_i^2 \bar{\rho}_i}{\rho_i} \mathbb{E} [\mathbb{I}(i \in \hat{S})] \mathbb{E} \left[\left(1 - \mathbb{I}(i \in \hat{S}) \beta_i \right) L_{i,\hat{S}}^1 \right] \right. \\
 &\quad \quad \left. \times \sum_{l=0}^{k-1} \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i \right)^{k-1-l} \mathbb{E} [\|\nabla_i f(X^l)\|_{(i)\star}] \right)
 \end{aligned}$$

$$\begin{aligned}
 & + \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \sigma_i \sqrt{\beta_i} \\
 & + \frac{1}{2} \sum_{i=1}^b t_i^2 \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^0 \right] + \frac{1}{2} \sum_{i=1}^b t_i^2 \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^1 \right] \mathbb{E} \left[\left\| \nabla_i f(X^k) \right\|_{(i)\star} \right] \\
 = & \delta^k - \sum_{i=1}^b t_i \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \mathbb{E} \left[\left\| \nabla_i f(X^k) \right\|_{(i)\star} \right] \\
 & + \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \left(1 - \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right)^k P_i^0 \\
 & + \sum_{i=1}^b \left(\frac{2t_i^2 \bar{\rho}_i}{\rho_i} \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \mathbb{E} \left[\left(1 - \mathbb{I} \left(i \in \hat{S} \right) \beta_i \right) L_{i,\hat{S}}^1 \right] \right. \\
 & \quad \left. \times \sum_{l=0}^{k-1} \left(1 - \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right)^{k-1-l} \mathbb{E} \left[\left\| \nabla_i f(X^l) \right\|_{(i)\star} \right] \right) \\
 & + \frac{1}{2} \sum_{i=1}^b t_i^2 \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^1 \right] \mathbb{E} \left[\left\| \nabla_i f(X^k) \right\|_{(i)\star} \right] \\
 & + \frac{1}{2} \sum_{i=1}^b t_i^2 \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^0 \right] + \sum_{i=1}^b \frac{2t_i^2 \bar{\rho}_i \mathbb{E} \left[\left(1 - \mathbb{I} \left(i \in \hat{S} \right) \beta_i \right) L_{i,\hat{S}}^0 \right]}{\beta_i \rho_i} \\
 & + \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \sigma_i \sqrt{\beta_i}.
 \end{aligned}$$

Let us now look at the terms involving the gradient norms. Using Lemma H.5, we get

$$\begin{aligned}
 & \sum_{i=1}^b \left(t_i^2 \frac{2\bar{\rho}_i}{\rho_i} \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \underbrace{\mathbb{E} \left[\left(1 - \mathbb{I} \left(i \in \hat{S} \right) \beta_i \right) L_{i,\hat{S}}^1 \right]}_{:=b_i} \right. \\
 & \quad \left. \times \sum_{l=0}^{k-1} \left(1 - \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right)^{k-1-l} \mathbb{E} \left[\left\| \nabla_i f(X^l) \right\|_{(i)\star} \right] \right) \\
 = & \sum_{l=0}^{k-1} \sum_{i=1}^b t_i^2 b_i \left(1 - \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right)^{k-1-l} \mathbb{E} \left[\left\| \nabla_i f(X^l) \right\|_{(i)\star} \right] \\
 \leq & \sum_{l=0}^{k-1} \mathbb{E} \left[4 \max_{i \in [b]} \left(t_i^2 b_i \left(1 - \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right)^{k-1-l} L_{i,S^l}^1 \right) (f(X^l) - f^\star) \right] \\
 & + \mathbb{E} \left[\sum_{i=1}^b \frac{t_i^2 b_i \left(1 - \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right)^{k-1-l} L_{i,S^l}^0}{L_{i,S^l}^1} \right] \\
 \leq & 4 \sum_{l=0}^{k-1} \mathbb{E} \left[\mathbb{E} \left[\max_{i \in [b]} \left(t_i^2 b_i \left(1 - \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right)^{k-1-l} \right) \max_{i \in [b]} \left(L_{i,S^l}^1 \right) (f(X^l) - f^\star) \middle| X^l \right] \right] \\
 & + \sum_{l=0}^{k-1} \sum_{i=1}^b t_i^2 b_i \left(1 - \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right)^{k-1-l} \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 = & 4 \sum_{l=0}^{k-1} \max_{i \in [b]} \left(t_i^2 b_i \left(1 - \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right)^{k-1-l} \right) \mathbb{E} \left[\max_{i \in [b]} L_{i,\hat{S}}^1 \right] \delta^l
 \end{aligned}$$

$$\begin{aligned}
 & + \sum_{i=1}^b t_i^2 b_i \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \sum_{l=0}^{k-1} \left(1 - \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i \right)^{k-1-l} \\
 \leq & 4 \max_{i \in [b]} (t_i^2 b_i) \mathbb{E} \left[\max_{i \in [b]} L_{i,\hat{S}}^1 \right] \sum_{l=0}^{k-1} \max_{i \in [b]} \left(\left(1 - \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i \right)^{k-1-l} \right) \delta^l \\
 & + \sum_{i=1}^b \frac{t_i^2 b_i}{\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i} \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 = & 4 \max_{i \in [b]} (t_i^2 b_i) \mathbb{E} \left[\max_{i \in [b]} L_{i,\hat{S}}^1 \right] \sum_{l=0}^{k-1} \left(1 - \min_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i \right) \right)^{k-1-l} \delta^l \\
 & + \sum_{i=1}^b \frac{t_i^2 b_i}{\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i} \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right],
 \end{aligned}$$

and

$$\begin{aligned}
 & \sum_{i=1}^b t_i^2 \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \mathbb{E} \left[\left\| \nabla_i f(X^k) \right\|_{(i)\star} \right] \\
 \leq & \mathbb{E} \left[4 \max_{i \in [b]} \left(t_i^2 \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] L_{i,S^k}^1 \right) (f(X^k) - f^\star) + \sum_{i=1}^b \frac{t_i^2 \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] L_{i,S^k}^0}{L_{i,S^k}^1} \right] \\
 \leq & 4 \max_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \right) \mathbb{E} \left[\max_{i \in [b]} \left(t_i^2 L_{i,S^k}^1 \right) (f(X^k) - f^\star) \right] \\
 & + \sum_{i=1}^b t_i^2 \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \mathbb{E} \left[\frac{L_{i,S^k}^0}{L_{i,S^k}^1} \right] \\
 = & 4 \max_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \right) \mathbb{E} \left[\mathbb{E} \left[\max_{i \in [b]} \left(t_i^2 L_{i,S^k}^1 \right) (f(X^k) - f^\star) \middle| X^k \right] \right] \\
 & + \sum_{i=1}^b t_i^2 \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 = & 4 \max_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \right) \mathbb{E} \left[\max_{i \in [b]} \left(t_i^2 L_{i,\hat{S}}^1 \right) \right] \delta^k \\
 & + \sum_{i=1}^b t_i^2 \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right].
 \end{aligned}$$

Thus

$$\begin{aligned}
 & \delta^{k+1} \\
 \leq & \delta^k - \sum_{i=1}^b t_i \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \mathbb{E} \left[\left\| \nabla_i f(X^k) \right\|_{(i)\star} \right] \\
 & + \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \left(1 - \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i \right)^k P_i^0 \\
 & + 4 \max_{i \in [b]} (t_i^2 b_i) \mathbb{E} \left[\max_{i \in [b]} L_{i,\hat{S}}^1 \right] \sum_{l=0}^{k-1} \left(1 - \min_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i \right) \right)^{k-1-l} \delta^l \\
 & + \sum_{i=1}^b \frac{t_i^2 b_i}{\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i} \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] + 2 \max_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \right) \mathbb{E} \left[\max_{i \in [b]} \left(t_i^2 L_{i,\hat{S}}^1 \right) \right] \delta^k
 \end{aligned}$$

$$\begin{aligned}
 & + \frac{1}{2} \sum_{i=1}^b t_i^2 \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^1 \right] \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 & + \frac{1}{2} \sum_{i=1}^b t_i^2 \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^0 \right] + \sum_{i=1}^b \frac{2t_i^2 \bar{\rho}_i \mathbb{E} \left[\left(1 - \mathbb{I} \left(i \in \hat{S} \right) \beta_i \right) L_{i,\hat{S}}^0 \right]}{\beta_i \rho_i} \\
 & + \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \sigma_i \sqrt{\beta_i} \\
 \leq & \left(1 + 2 \max_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^1 \right] \right) \mathbb{E} \left[\max_{i \in [b]} \left(t_i^2 L_{i,\hat{S}}^1 \right) \right] \right) \delta^k \\
 & - \sum_{i=1}^b t_i \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \mathbb{E} \left[\left\| \nabla_i f(X^k) \right\|_{(i)\star} \right] \\
 & + 4 \max_{i \in [b]} \left(t_i^2 b_i \right) \mathbb{E} \left[\max_{i \in [b]} L_{i,\hat{S}}^1 \right] \sum_{l=0}^{k-1} \left(1 - \min_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right) \right)^{k-1-l} \delta^l \\
 & + \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \left(1 - \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right)^k P_i^0 \\
 & + \sum_{i=1}^b t_i^2 \left(\frac{2\bar{\rho}_i \mathbb{E} \left[\left(1 - \mathbb{I} \left(i \in \hat{S} \right) \beta_i \right) L_{i,\hat{S}}^1 \right]}{\beta_i \rho_i} + \frac{\mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^1 \right]}{2} \right) \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 & + \sum_{i=1}^b t_i^2 \left(\frac{2\bar{\rho}_i \mathbb{E} \left[\left(1 - \mathbb{I} \left(i \in \hat{S} \right) \beta_i \right) L_{i,\hat{S}}^0 \right]}{\beta_i \rho_i} + \frac{\mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^0 \right]}{2} \right) \\
 & + 2 \sum_{i=1}^b t_i \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \bar{\rho}_i \sigma_i \sqrt{\beta_i} \\
 = & (1 + c_1) \delta^k - \sum_{i=1}^b t_i \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \mathbb{E} \left[\left\| \nabla_i f(X^k) \right\|_{(i)\star} \right] \\
 & + c_2 \sum_{l=0}^{k-1} \left(1 - \min_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right) \right)^{k-1-l} \delta^l \\
 & + \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \left(1 - \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \beta_i \right)^k P_i^0 + \sum_{i=1}^b t_i^2 c_{3,i} + \sum_{i=1}^b t_i c_{4,i},
 \end{aligned}$$

where

$$\begin{aligned}
 c_1 &:= 2 \max_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^1 \right] \right) \mathbb{E} \left[\max_{i \in [b]} \left(t_i^2 L_{i,\hat{S}}^1 \right) \right], \\
 c_2 &:= 4 \max_{i \in [b]} \left(t_i^2 b_i \right) \mathbb{E} \left[\max_{i \in [b]} L_{i,\hat{S}}^1 \right], \\
 c_{3,i} &:= \left(\frac{2\bar{\rho}_i \mathbb{E} \left[\left(1 - \mathbb{I} \left(i \in \hat{S} \right) \beta_i \right) L_{i,\hat{S}}^1 \right]}{\beta_i \rho_i} + \frac{\mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^1 \right]}{2} \right) \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 & \quad + \left(\frac{2\bar{\rho}_i \mathbb{E} \left[\left(1 - \mathbb{I} \left(i \in \hat{S} \right) \beta_i \right) L_{i,\hat{S}}^0 \right]}{\beta_i \rho_i} + \frac{\mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) L_{i,\hat{S}}^0 \right]}{2} \right), \\
 c_{4,i} &:= 2 \mathbb{E} \left[\mathbb{I} \left(i \in \hat{S} \right) \right] \bar{\rho}_i \sigma_i \sqrt{\beta_i}.
 \end{aligned}$$

Now, let us introduce a weighting sequence $w^k := w^{k-1} \left(1 + c_1 + \frac{c_2}{\min_{i \in [b]} (\mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i)} \right)^{-1}$, where $w^{-1} = 1$ and $W^K := \sum_{k=0}^K w^k$. Then, multiplying the inequality above by w^k and summing over the first $K+1$ iterations gives

$$\begin{aligned}
 \sum_{k=0}^K w^k \delta^{k+1} &\leq \sum_{k=0}^K w^k (1 + c_1) \delta^k - \sum_{k=0}^K w^k \sum_{i=1}^b t_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] \mathbb{E} [\|\nabla_i f(X^k)\|_{(i)\star}] \\
 &\quad + \sum_{k=0}^K w^k c_2 \sum_{l=0}^{k-1} \left(1 - \min_{i \in [b]} (\mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i) \right)^{k-1-l} \delta^l \\
 &\quad + \sum_{k=0}^K w^k \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i \right)^k P_i^0 \\
 &\quad + \sum_{k=0}^K w^k \sum_{i=1}^b t_i^2 c_{3,i} + \sum_{k=0}^K w^k \sum_{i=1}^b t_i c_{4,i} \\
 &\leq (1 + c_1) \sum_{k=0}^K w^k \delta^k - \sum_{k=0}^K w^k \sum_{i=1}^b t_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] \mathbb{E} [\|\nabla_i f(X^k)\|_{(i)\star}] \\
 &\quad + c_2 \sum_{l=0}^{K-1} \sum_{k=l+1}^K w^k \left(1 - \min_{i \in [b]} (\mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i) \right)^{k-1-l} \delta^l \\
 &\quad + \sum_{i=1}^b 2t_i \bar{\rho}_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] P_i^0 \sum_{k=0}^K \left(1 - \mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i \right)^k \\
 &\quad + W^K \sum_{i=1}^b t_i^2 c_{3,i} + W^K \sum_{i=1}^b t_i c_{4,i},
 \end{aligned}$$

where in the last line we used the fact that $w^k \leq w^{k-1} \leq w^{-1} = 1$. Therefore,

$$\begin{aligned}
 \sum_{k=0}^K w^k \delta^{k+1} &\leq (1 + c_1) \sum_{k=0}^K w^k \delta^k - \sum_{k=0}^K w^k \sum_{i=1}^b t_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] \mathbb{E} [\|\nabla_i f(X^k)\|_{(i)\star}] \\
 &\quad + c_2 \sum_{l=0}^{K-1} \sum_{k=l+1}^K w^l \left(1 - \min_{i \in [b]} (\mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i) \right)^{k-1-l} \delta^l \\
 &\quad + \sum_{i=1}^b \frac{2t_i \bar{\rho}_i \mathbb{E} [\mathbb{I}(i \in \hat{S})]}{\mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i} P_i^0 + W^K \sum_{i=1}^b t_i^2 c_{3,i} + W^K \sum_{i=1}^b t_i c_{4,i} \\
 &\leq (1 + c_1) \sum_{k=0}^K w^k \delta^k - \sum_{k=0}^K w^k \sum_{i=1}^b t_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] \mathbb{E} [\|\nabla_i f(X^k)\|_{(i)\star}] \\
 &\quad + c_2 \sum_{l=0}^{K-1} w^l \delta^l \sum_{k=l+1}^K \left(1 - \min_{i \in [b]} (\mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i) \right)^{k-1-l} \\
 &\quad + \sum_{i=1}^b \frac{2t_i \bar{\rho}_i}{\beta_i} P_i^0 + W^K \sum_{i=1}^b t_i^2 c_{3,i} + W^K \sum_{i=1}^b t_i c_{4,i} \\
 &\leq (1 + c_1) \sum_{k=0}^K w^k \delta^k - \sum_{k=0}^K w^k \sum_{i=1}^b t_i \mathbb{E} [\mathbb{I}(i \in \hat{S})] \mathbb{E} [\|\nabla_i f(X^k)\|_{(i)\star}] \\
 &\quad + c_2 \sum_{l=0}^{K-1} w^l \delta^l \sum_{k=0}^{\infty} \left(1 - \min_{i \in [b]} (\mathbb{E} [\mathbb{I}(i \in \hat{S})] \beta_i) \right)^k
 \end{aligned}$$

$$\begin{aligned}
 & + \sum_{i=1}^b \frac{2t_i \bar{\rho}_i}{\beta_i} P_i^0 + W^K \sum_{i=1}^b t_i^2 c_{3,i} + W^K \sum_{i=1}^b t_i c_{4,i} \\
 & \leq \left(1 + c_1 + \frac{c_2}{\min_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i \right)} \right) \sum_{k=0}^K w^k \delta^k \\
 & \quad - \sum_{k=0}^K w^k \sum_{i=1}^b t_i \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star} \right] \\
 & \quad + \sum_{i=1}^b \frac{2t_i \bar{\rho}_i}{\beta_i} P_i^0 + W^K \sum_{i=1}^b t_i^2 c_{3,i} + W^K \sum_{i=1}^b t_i c_{4,i} \\
 & = \sum_{k=0}^K w^{k-1} \delta^k - \sum_{k=0}^K w^k \sum_{i=1}^b t_i \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star} \right] \\
 & \quad + \sum_{i=1}^b \frac{2t_i \bar{\rho}_i}{\beta_i} P_i^0 + W^K \sum_{i=1}^b t_i^2 c_{3,i} + W^K \sum_{i=1}^b t_i c_{4,i}.
 \end{aligned}$$

Rearranging the terms and dividing by W^K , we get

$$\begin{aligned}
 & \min_{k=0, \dots, K} \sum_{i=1}^b t_i \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star} \right] \\
 & \leq \sum_{k=0}^K \sum_{i=1}^b \frac{w^k}{W^K} t_i \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star} \right] \\
 & \leq \frac{1}{W^K} \sum_{k=0}^K (w^{k-1} \delta^k - w^k \delta^{k+1}) + \frac{1}{W^K} \sum_{i=1}^b \frac{2t_i \bar{\rho}_i}{\beta_i} P_i^0 + \sum_{i=1}^b t_i^2 c_{3,i} + \sum_{i=1}^b t_i c_{4,i} \\
 & \leq \frac{\delta^0}{W^K} + \frac{1}{W^K} \sum_{i=1}^b \frac{2t_i \bar{\rho}_i}{\beta_i} P_i^0 + \sum_{i=1}^b t_i^2 c_{3,i} + \sum_{i=1}^b t_i c_{4,i}.
 \end{aligned}$$

Now, note that taking $t_i = \frac{\eta_i}{(K+1)^{3/4}}$, where

$$\eta_i^2 \leq \min \left\{ \frac{(K+1)^{1/2}}{4 \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \mathbb{E} \left[\max_{i \in [b]} L_{i,\hat{S}}^1 \right]}, \frac{(K+1)^{1/2} \rho_i \min_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i \right)}{16 \bar{\rho}_i \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \mathbb{E} \left[(1 - \mathbb{I}(i \in \hat{S})) \beta_i \right] L_{i,\hat{S}}^1 \mathbb{E} \left[\max_{i \in [b]} L_{i,\hat{S}}^1 \right]}, 1 \right\},$$

we have

$$\begin{aligned}
 & 2(K+1) \max_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \right) \mathbb{E} \left[\max_{i \in [b]} \left(t_i^2 L_{i,\hat{S}}^1 \right) \right] \leq \frac{1}{2}, \\
 & (K+1) \frac{4 \max_{i \in [b]} (t_i^2 b_i) \mathbb{E} \left[\max_{i \in [b]} L_{i,\hat{S}}^1 \right]}{\min_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i \right)} \leq \frac{1}{2},
 \end{aligned}$$

and hence $(K+1) \left(c_1 + \frac{c_2}{\min_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i \right)} \right) \leq 1$. Thus, the weights satisfy

$$\begin{aligned}
 W^K & = \sum_{k=0}^K w^k \geq (K+1)w^K = \frac{(K+1)w^{-1}}{\left(1 + c_1 + \frac{c_2}{\min_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i \right)} \right)^{K+1}} \\
 & \geq \frac{K+1}{\exp \left((K+1) \left(c_1 + \frac{c_2}{\min_{i \in [b]} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \beta_i \right)} \right) \right)} \geq \frac{K+1}{\exp(1)} \geq \frac{K+1}{3},
 \end{aligned}$$

meaning that

$$\begin{aligned}
 & \min_{k=0,\dots,K} \sum_{i=1}^b t_i \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star} \right] \\
 & \leq \frac{3\delta^0}{K+1} + \frac{6}{K+1} \sum_{i=1}^b \frac{t_i \bar{\rho}_i}{\beta_i} P_i^0 + \sum_{i=1}^b t_i^2 c_{3,i} + \sum_{i=1}^b t_i c_{4,i} \\
 & = \frac{3\delta^0}{K+1} + \frac{6}{K+1} \sum_{i=1}^b \frac{\eta_i \bar{\rho}_i}{\beta_i (K+1)^{3/4}} P_i^0 \\
 & \quad + \sum_{i=1}^b \frac{\eta_i^2}{(K+1)^{3/2}} \frac{2\bar{\rho}_i}{\beta_i \underline{\rho}_i} \left(\mathbb{E} \left[(1 - \mathbb{I}(i \in \hat{S})) \beta_i \right] L_{i,\hat{S}}^0 + \mathbb{E} \left[(1 - \mathbb{I}(i \in \hat{S})) \beta_i \right] L_{i,\hat{S}}^1 \right) \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 & \quad + \sum_{i=1}^b \frac{\eta_i^2}{2(K+1)^{3/2}} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^0 \right] + \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \right) \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 & \quad + \sum_{i=1}^b \frac{2\eta_i}{(K+1)^{3/4}} \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \bar{\rho}_i \sigma_i \sqrt{\beta_i}.
 \end{aligned}$$

Dividing by $\frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] t_l = \frac{1}{(K+1)^{3/4}} \frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l$ gives

$$\begin{aligned}
 & \min_{k=0,\dots,K} \sum_{i=1}^b \frac{\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \eta_i}{\frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l} \mathbb{E} \left[\|\nabla_i f(X^k)\|_{(i)\star} \right] \\
 & \leq \frac{3\delta^0}{(K+1)^{1/4} \frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l} + \frac{6}{K+1} \sum_{i=1}^b \frac{\eta_i \bar{\rho}_i}{\beta_i \frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l} P_i^0 \\
 & \quad + \sum_{i=1}^b \frac{\eta_i^2}{(K+1)^{3/4} \frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l} \frac{2\bar{\rho}_i}{\beta_i \underline{\rho}_i} \\
 & \quad + \times \left(\mathbb{E} \left[(1 - \mathbb{I}(i \in \hat{S})) \beta_i \right] L_{i,\hat{S}}^0 + \mathbb{E} \left[(1 - \mathbb{I}(i \in \hat{S})) \beta_i \right] L_{i,\hat{S}}^1 \right) \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 & \quad + \sum_{i=1}^b \frac{\eta_i^2}{2(K+1)^{3/4} \frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^0 \right] + \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \right) \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 & \quad + \sum_{i=1}^b \frac{2\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \eta_i}{\frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l} \bar{\rho}_i \sigma_i \sqrt{\beta_i} \\
 & \leq \frac{3\delta^0}{(K+1)^{1/4} \frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l} + \frac{6}{(K+1)^{1/2}} \sum_{i=1}^b \frac{\eta_i \bar{\rho}_i}{\frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l} P_i^0 \\
 & \quad + \sum_{i=1}^b \frac{\eta_i^2}{(K+1)^{1/4} \frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l} \frac{2\bar{\rho}_i}{\underline{\rho}_i} \left(\mathbb{E} \left[L_{i,\hat{S}}^0 \right] + \mathbb{E} \left[L_{i,\hat{S}}^1 \right] \right) \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 & \quad + \sum_{i=1}^b \frac{\eta_i^2}{2(K+1)^{3/4} \frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l} \left(\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^0 \right] + \mathbb{E} \left[\mathbb{I}(i \in \hat{S}) L_{i,\hat{S}}^1 \right] \right) \mathbb{E} \left[\frac{L_{i,\hat{S}}^0}{L_{i,\hat{S}}^1} \right] \\
 & \quad + \sum_{i=1}^b \frac{2\mathbb{E} \left[\mathbb{I}(i \in \hat{S}) \right] \eta_i}{(K+1)^{1/4} \frac{1}{b} \sum_{l=1}^b \mathbb{E} \left[\mathbb{I}(l \in \hat{S}) \right] \eta_l} \bar{\rho}_i \sigma_i,
 \end{aligned}$$

where in the last equality we set $\beta_i = \frac{1}{(K+1)^{1/2}}$. Finally, the initialization $M_i^0 = \nabla_i f(X^0; \xi^0)$ guarantees that

$$P_i^0 := \mathbb{E} [\|\nabla_i f(X^0) - M_i^0\|_2] \leq \sqrt{\mathbb{E} [\|\nabla_i f(X^0) - \nabla_i f(X^0; \xi^0)\|_2^2]} \stackrel{(4.5)}{\leq} \sigma_i.$$

□

G Experiments

We evaluate Drop-Muon on three standard benchmarks—MNIST, Fashion-MNIST, and CIFAR-10—using 3-layer convolutional neural networks (CNNs) of varying capacity. Our goal is to study how partial layer updates of Drop-Muon accelerate training relative to standard Muon (a special case of Drop-Muon with full-network updates).

Experiments were run on various NVIDIA GPUs based on availability. MNIST and Fashion-MNIST experiments were run on Tesla V100-SXM2-32GB. CIFAR-10 experiments were performed on a mix of A100-SXM4-80GB, Tesla P100-PCIE-16GB, and GeForce GTX 1080 Ti GPUs.

G.1 Key Training Parameters

We identify several key training parameters and conduct a grid search across these parameters to systematically explore their effects.

Layer sampling distribution. The choice of layer distribution determines which layers are updated at each iteration and is central to the behavior of Drop-Muon. We considered the following strategies:

- **Uniform Distribution:** Each layer is sampled with equal probability ($p_i = 1/b$ for all $i \in [b]$ in the notation from Section 4).
- **Linear Distribution:** Sampling probability increases linearly with the layer index, favoring deeper layers. Empirically, this distribution performed poorly.
- **Quadratic Distribution:** Probability increases quadratically with depth, strongly biasing toward deeper layers. Similarly, this performed poorly.
- **Exponential Distribution:** Layer probability grows exponentially with depth, emphasizing deeper layers. This also proved suboptimal.
- **Epoch-Shift Distribution:** The distribution biases the sampling towards shallow layers at the beginning of training, gradually shifting towards deeper layers as the training progresses (see more details in Section G.2). This dynamic adjustment might help in balancing early feature extraction and later complex feature learning during the optimization process—see example in Figure 4.

Batch size and learning rate. We test batch sizes $\{64, 512, 8192, 16384, 32768\}$ and learning rates $\{0.1, 0.01, 0.001\}$.

Model depth and capacity. To assess the impact of network complexity, we train multiple 3-layer CNNs with varying channel configurations: $[8, 16, 32]$, $[16, 32, 64]$, $[64, 128, 256]$, $[128, 256, 512]$, $[64, 16, 8]$, and $[256, 128, 64]$.

Fixed parameters. We fix the number of epochs to 20 or 50, momentum to 0.5, and the number of Newton-Schulz iterations to 5 (see Section A.1).

This grid search allows us to systematically evaluate Drop-Muon. While many combinations performed poorly (and are therefore omitted in the results that follow), the experiments highlighted configurations—particularly with uniform or epoch-shift distributions—that consistently reduce training time while maintaining final accuracy.

G.2 Epoch-Shift Distribution

Drop-Muon can dynamically adjust which layers are updated at each iteration. The strategy we consider is the *epoch-shift* distribution, which biases training towards shallow layers in the early epochs, gradually shifting focus to deeper layers as training progresses (see Figure 4). Such a schedule balances early feature extraction with later complex feature learning, improving convergence in practice.

Formally, let b denote the total number of layers, α be a constant controlling the sharpness of the bias, and define the training progress as

$$\text{progress} = \frac{\text{epoch}}{\text{max_epochs}} \in [0, 1].$$

The weight for each layer i is given by

$$w_i = \exp(\alpha[(1 - \text{progress})(b - 1 - i) + \text{progress} \cdot i]).$$

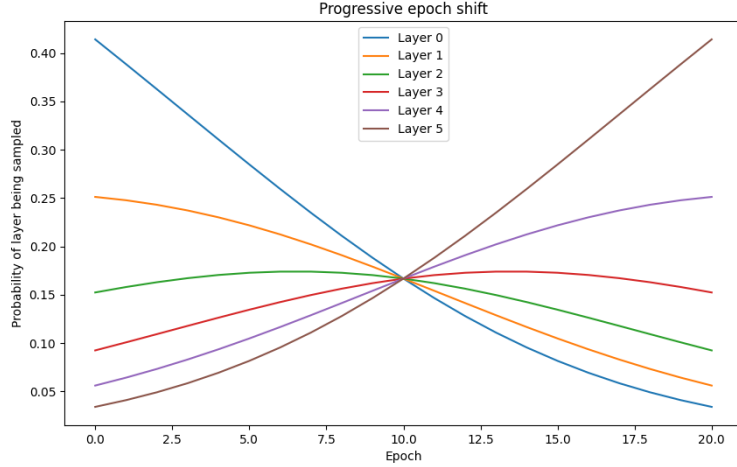


Figure 4: Evolution of the layer sampling distribution as a function of the epochs. Shallow layers are more trained in the first epochs but their probabilities of being sampled decrease with the epochs. This effect can be amplified or reduced by varying the value of α ; here we chose $\alpha = 0.5$.

We then normalize the weights to obtain a valid probability distribution:

$$W = \sum_{i=0}^{b-1} w_i, \quad p_i = \frac{w_i}{W}.$$

By adjusting α , one can control the rate at which training emphasis shifts from shallow to deeper layers. In Figure 4, we set $\alpha = 0.5$.

G.3 Evaluation Metrics

We repeat every experiment over multiple random seeds. When reporting aggregated results, we use two complementary procedures:

- **Normalized curve averaging:** We normalize wall-clock time for each run so that Muon always ends at $t = 1$, interpolate both Muon and Drop-Muon curves onto this grid, and average over seeds.
- **Time-to-target evaluation:** We measure the wall-clock time to reach fixed accuracy thresholds (e.g. 60%, 70%, 80%, ...), reporting the ratio between Muon and Drop-Muon. This metric is interpolation-free and directly quantifies practical speedup.

G.4 Results on MNIST

We first evaluate Drop-Muon on the MNIST dataset. Figure 1 shows a representative run comparing standard full-network Muon training with Drop-Muon using a uniform layer sampling distribution. Although the per-epoch train accuracy of Drop-Muon is initially lower than that of full-layer Muon, the wall-clock training time tells a different story: for training times up to approximately 150 seconds, Drop-Muon consistently achieves higher accuracy. In practice, this means that reaching a train accuracy of 95% or less is faster with Drop-Muon, highlighting its efficiency advantage.

To account for variability across independent runs, we evaluate two aggregation strategies. Figure 5 shows normalized curve averaging, where each run is rescaled to a common time grid. Figure 2 (left) presents averaged time-to-target evaluation across multiple seeds, reporting the ratio of Muon time to Drop-Muon time for fixed accuracy thresholds.

The time-to-target results clearly demonstrate a speedup of up to $1.4\times$ across thresholds of 60%, 70%, 80%, and 99%, with slightly lower speedup for the 90% threshold. We note variability across seeds, emphasizing the stochastic nature of layer sampling. Despite this, Drop-Muon consistently provides practical acceleration across training.

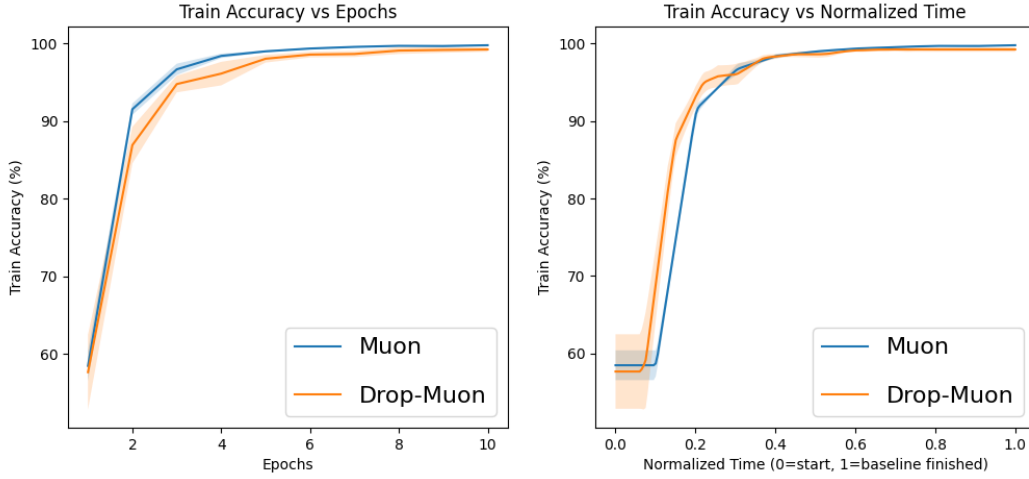


Figure 5: Normalized curve averaging of several runs of Muon and Drop-Muon with uniform index sampling on MNIST. Batch size = 8192, learning rate = 0.1, channels = [64, 128, 256].

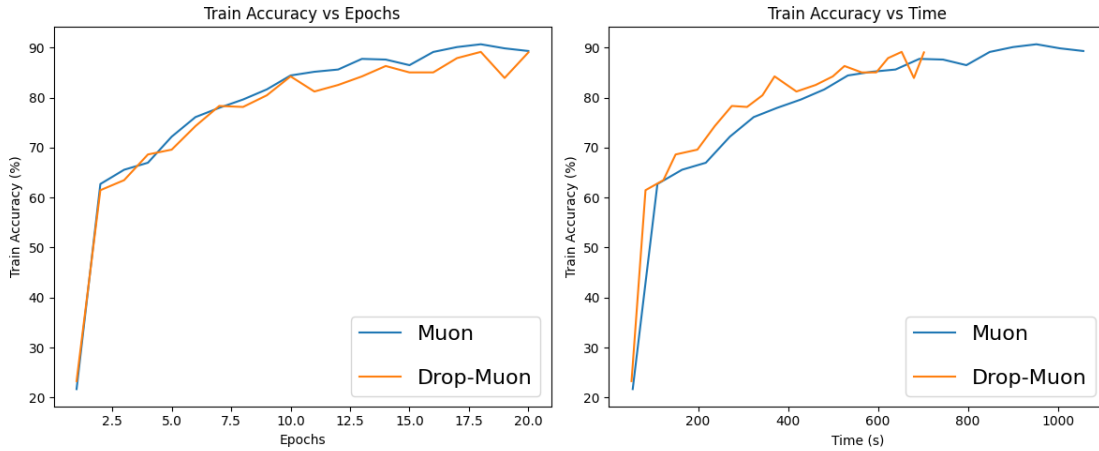


Figure 6: Evolution of the training accuracy for Muon and Drop-Muon with epoch-shift index sampling on Fashion-MNIST. Batch size = 32768, learning rate = 0.1, channels = [64, 128, 256].

G.5 Results on Fashion-MNIST

Figure 6 shows a typical run comparing standard Muon training with Drop-Muon using the epoch-shift layer sampling distribution. Early in training, Drop-Muon achieves faster progress in wall-clock time, even if per-epoch accuracy is slightly lower.

Normalized curve averaging across multiple seeds is presented in Figure 7. Although the curves appear smoother at higher accuracy, this should not be interpreted as a reduction in variance; early-stage training is inherently noisier due to layer sampling, whereas later training reflects fewer updates per wall-clock time.

Figure 2 (right) reports the averaged time-to-target evaluation. Drop-Muon consistently achieves speedups of approximately 1.2 across all considered accuracy thresholds. The 99% threshold is omitted as neither Muon nor Drop-Muon reach this level within the plotted runs.

Overall, Fashion-MNIST experiments reinforce that Drop-Muon provides consistent practical speedups, similar to MNIST, while maintaining comparable final accuracy.

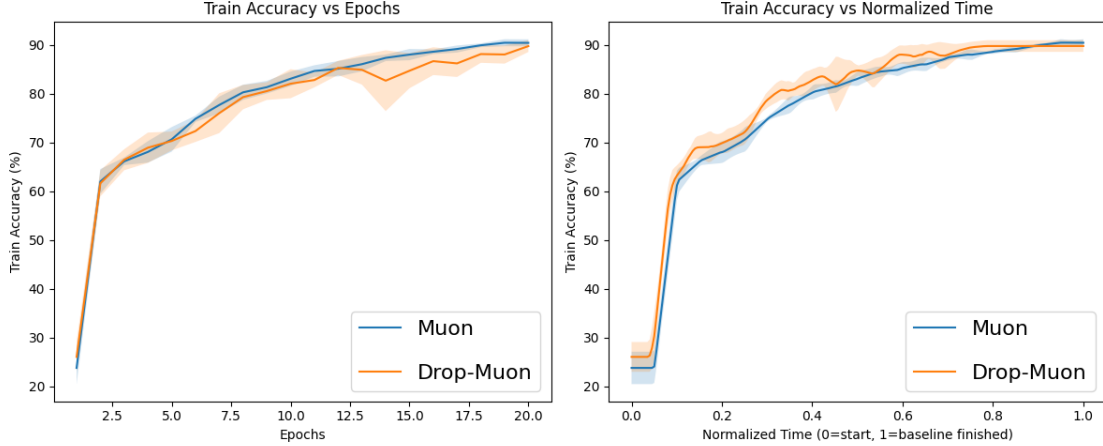


Figure 7: Normalized curve averaging of several runs of Muon and Drop-Muon with epoch-shift index sampling on Fashion-MNIST. Batch size = 32768, learning rate = 0.1, channels = [64, 128, 256].

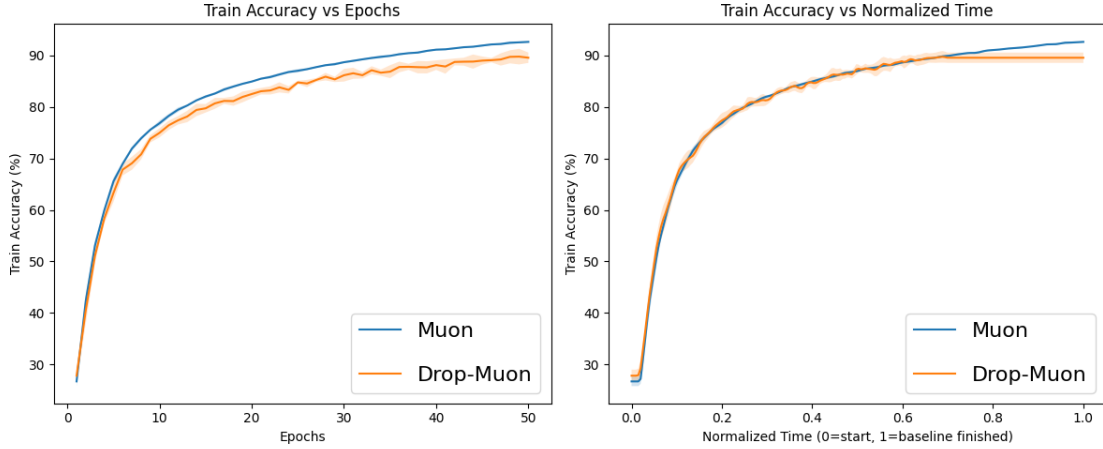


Figure 8: Normalized curve averaging of several runs of Muon and Drop-Muon with epoch-shift index sampling on CIFAR-10. Batch size = 8192, learning rate = 0.1, channels = [128, 256, 512].

G.6 Results on CIFAR-10

CIFAR-10 requires more sophisticated network architectures. We use a CNN with channels [128, 256, 512] and insert batch normalization layers between each convolution and ReLU activation to improve training stability.

Figure 3 shows an example run comparing standard full-layer Muon training with Drop-Muon using the epoch-shift layer sampling distribution. While Drop-Muon has slightly lower per-epoch training accuracy, it achieves faster progress in terms of wall-clock time. Figure 8 shows normalized curve averaging across multiple seeds (the flat end of the curve reflects the earlier completion of Drop-Muon).

Time-to-target results in Figure 9 indicate a notable speedup at the 90% train accuracy threshold. However, for lower thresholds (60%, 70%, 80%), speedups are minimal or absent for this configuration. We attribute this limitation primarily to sub-optimal hyperparameter choices (see Section 6).

G.7 Discussion and Practical Remarks

We summarize several key observations and practical considerations arising from our experiments:

- **Simplicity of implementation:** Drop-Muon can be implemented in just a few lines of code, making it easy to integrate into existing training pipelines.

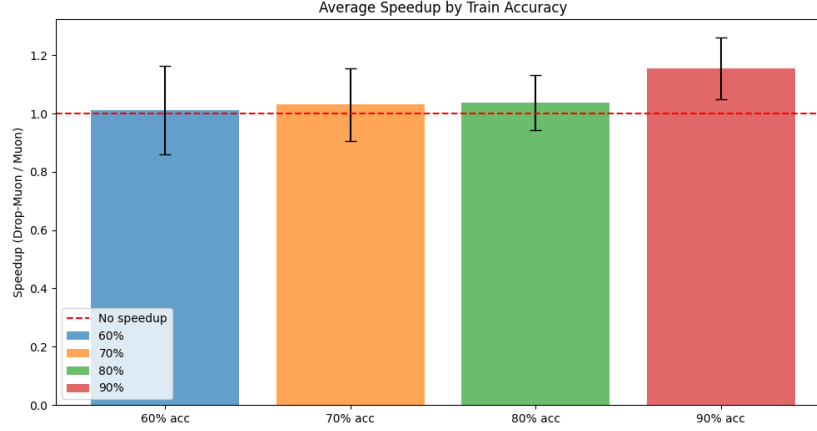


Figure 9: Averaged time-to-target speed-up over multiple runs comparing Muon and Drop-Muon with epoch-shift index sampling on CIFAR-10 with batch size 8192, learning rate 0.1, and channels [128, 256, 512].

- **Further benefits from tuning:** Performance can be further improved through dedicated tuning of hyperparameters (see Section 6).
- **Potential for implementation improvements:**
 - Turning off gradient computations at every iteration can be costly for large models. Sampling the cutoff layer only every few iterations can reduce overhead and improve efficiency.
 - When the batch size or model size is small, the gradient computation may be dominated by the Newton-Schulz routine. In such cases, computing all gradients but orthogonalizing only a subset can be advantageous.
- **Variance across seeds:** The method exhibits non-negligible variance, particularly on smaller datasets. Stabilization mechanisms could mitigate this.
- **Adaptive learning of sampling distributions:** A promising future direction is to learn the layer sampling distribution online, so that it automatically adapts to the training dynamics of a given dataset and architecture.

H Useful Facts

For all $X, Y \in \mathcal{X}$ and $t > 0$, we have:

$$\|\text{LMO}_{\mathcal{B}(0,t)}(G)\| = t \quad (48)$$

$$\langle G, \text{LMO}_{\mathcal{B}(X,t)}(G) \rangle = -t \|G\|_{\star} \quad (49)$$

$$\langle X, X^{\sharp} \rangle = \|X^{\sharp}\|^2, \quad (50)$$

$$\|X\|_{\star} = \|X^{\sharp}\|, \quad (51)$$

$$\langle H, X \rangle = \|X\|_{\star}, \quad \|H\| = 1, \quad \forall X \neq 0. \quad (52)$$

where $H \in \partial \|\cdot\|_{\star}(X)$ belongs to the subdifferential of the dual norm.

Lemma H.1 (Variance decomposition). For any random vector $X \in \mathcal{X}$ and any non-random $c \in \mathcal{X}$, we have

$$\mathbb{E} [\|X - c\|_2^2] = \mathbb{E} [\|X - \mathbb{E}[X]\|_2^2] + \|\mathbb{E}[X] - c\|_2^2.$$

Lemma H.2 (Riabini et al. [2025], Lemma 3). Suppose that $x_1, \dots, x_p, y_1, \dots, y_p \in \mathbb{R}$, $\max_{i \in [b]} |x_i| > 0$ and $z_1, \dots, z_p > 0$. Then

$$\sum_{i=1}^b \frac{y_i^2}{z_i} \geq \frac{\left(\sum_{i=1}^b x_i y_i\right)^2}{\sum_{i=1}^b z_i x_i^2}.$$

Lemma H.3. Let Assumption 4.4 hold and let $S \subseteq [b]$. Then, for any vectors $X = [X_1, \dots, X_b] \in \mathcal{X}$ and $\Gamma = [\Gamma_1, \dots, \Gamma_b] \in \mathcal{X}$ such that $\Gamma_i = 0$ for all $i \notin S$,

$$|f(X + \Gamma) - f(X) - \langle \nabla f(X), \Gamma \rangle| \leq \sum_{i \in S} \frac{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}}{2} \|\Gamma_i\|_{(i)}^2.$$

Proof. For all $X \in \mathcal{X}$ we have

$$\begin{aligned} f(X + \Gamma) &= f(X) + \int_0^1 \langle \nabla f(X + \tau\Gamma), \Gamma \rangle d\tau \\ &= f(X) + \int_0^1 \langle \nabla f(X + \tau\Gamma) - \nabla f(X), \Gamma \rangle d\tau + \langle \nabla f(X), \Gamma \rangle. \end{aligned}$$

Therefore, using the Cauchy-Schwarz inequality

$$\begin{aligned} |f(X + \Gamma) - f(X) - \langle \nabla f(X), \Gamma \rangle| &= \left| \int_0^1 \sum_{i=1}^b \langle \nabla_i f(X + \tau\Gamma) - \nabla_i f(X), \Gamma_i \rangle_{(i)} d\tau \right| \\ &= \left| \int_0^1 \sum_{i \in S} \langle \nabla_i f(X + \tau\Gamma) - \nabla_i f(X), \Gamma_i \rangle_{(i)} d\tau \right| \\ &\leq \int_0^1 \sum_{i \in S} \left| \langle \nabla_i f(X + \tau\Gamma) - \nabla_i f(X), \Gamma_i \rangle_{(i)} \right| d\tau \\ &\leq \int_0^1 \sum_{i \in S} \|\nabla_i f(X + \tau\Gamma) - \nabla_i f(X)\|_{(i)\star} \|\Gamma_i\|_{(i)} d\tau \\ &\stackrel{(4.4)}{\leq} \int_0^1 \sum_{i \in S} \tau \left(L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star} \right) \|\Gamma_i\|_{(i)}^2 d\tau \\ &= \sum_{i \in S} \frac{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}}{2} \|\Gamma_i\|_{(i)}^2. \end{aligned}$$

□

Lemma H.4. Let Assumptions 4.1 and Assumption 4.3 hold and let $S \subseteq [b]$. Then

$$\sum_{i \in S} \frac{\|\nabla_i f(X)\|_{(i)\star}^2}{2 \left(L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star} \right)} \leq f(X) - f^\star$$

for all $X \in \mathcal{X}$.

Proof. Let $Y = [Y_1, \dots, Y_b] \in \mathcal{X}$, where $Y_i = X_i - \frac{\|\nabla_i f(X)\|_{(i)\star}}{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}} H_i$ for some $H_i \in \partial \|\cdot\|_{(i)\star}(\nabla_i f(X))$ for $i \in S$ and $Y_i = X_i$ otherwise. By Lemma H.3

$$\begin{aligned} f(Y) &\leq f(X) + \langle \nabla f(X), Y - X \rangle + \sum_{i \in S} \frac{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}}{2} \|X_i - Y_i\|_{(i)}^2 \\ &= f(X) + \sum_{i \in S} \langle \nabla_i f(X), Y_i - X_i \rangle_{(i)} + \sum_{i \in S} \frac{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}}{2} \|X_i - Y_i\|_{(i)}^2 \\ &= f(X) - \sum_{i \in S} \frac{\|\nabla_i f(X)\|_{(i)\star}}{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}} \langle \nabla_i f(X), H_i \rangle_{(i)} \\ &\quad + \sum_{i \in S} \left(\frac{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}}{2} \frac{\|\nabla_i f(X)\|_{(i)\star}^2}{\left(L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star} \right)^2} \|H_i\|_{(i)}^2 \right) \\ &\stackrel{(52)}{=} f(X) + \sum_{i \in S} \left(-\frac{\|\nabla_i f(X)\|_{(i)\star}^2}{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}} + \frac{\|\nabla_i f(X)\|_{(i)\star}^2}{2 \left(L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star} \right)} \right) \\ &= f(X) - \sum_{i \in S} \frac{\|\nabla_i f(X)\|_{(i)\star}^2}{2 \left(L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star} \right)}, \end{aligned}$$

and hence

$$\sum_{i \in S} \frac{\|\nabla_i f(X)\|_{(i)\star}^2}{2 \left(L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star} \right)} \leq f(X) - f(Y) \leq f(X) - f^\star.$$

□

Lemma H.5. Let Assumptions 4.1 and Assumption 4.3 hold and let $S \subseteq [b]$. Then, for any $x_i > 0$, $i \in [b]$, we have

$$\sum_{i \in S} x_i \|\nabla_i f(X)\|_{(i)\star} \leq 4 \max_{i \in S} (x_i L_{i,S}^1) (f(X) - f^\star) + \sum_{i \in S} \frac{x_i L_{i,S}^0}{L_{i,S}^1}$$

for all $X \in \mathcal{X}$.

Proof. Applying Lemma H.4 and Lemma H.2 with $y_i = \|\nabla_i f(X)\|_{(i)\star}$, $z_i = L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}$ and any $x_i > 0$, we have

$$\begin{aligned} 2(f(X) - f^\star) &\geq \sum_{i \in S} \frac{\|\nabla_i f(X)\|_{(i)\star}^2}{L_{i,S}^0 + L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}} \\ &\geq \frac{\left(\sum_{i \in S} x_i \|\nabla_i f(X)\|_{(i)\star} \right)^2}{\sum_{i \in S} x_i^2 L_{i,S}^0 + \sum_{i \in S} x_i^2 L_{i,S}^1 \|\nabla_i f(X)\|_{(i)\star}} \\ &\geq \frac{\left(\sum_{i \in S} x_i \|\nabla_i f(X)\|_{(i)\star} \right)^2}{\sum_{i \in S} x_i^2 L_{i,S}^0 + \max_{i \in S} (x_i L_{i,S}^1) \sum_{i \in S} x_i \|\nabla_i f(X)\|_{(i)\star}} \end{aligned}$$

$$\geq \begin{cases} \frac{(\sum_{i \in S} x_i \|\nabla_i f(X)\|_{(i)\star})^2}{2 \sum_{i \in S} x_i^2 L_{i,S}^0} & \text{if } \frac{\sum_{i \in S} x_i^2 L_{i,S}^0}{\max_{i \in S} (x_i L_{i,S}^1)} \geq \sum_{i \in S} x_i \|\nabla_i f(X)\|_{(i)\star}, \\ \frac{\sum_{i \in S} x_i \|\nabla_i f(X)\|_{(i)\star}}{2 \max_{i \in S} (x_i L_{i,S}^1)} & \text{otherwise.} \end{cases}$$

Therefore,

$$\begin{aligned} \sum_{i \in S} x_i \|\nabla_i f(X)\|_{(i)\star} &\leq \max \left\{ 4 \max_{i \in S} (x_i L_{i,S}^1) (f(X) - f^\star), \frac{\sum_{i \in S} x_i^2 L_{i,S}^0}{\max_{i \in S} (x_i L_{i,S}^1)} \right\} \\ &\leq 4 \max_{i \in S} (x_i L_{i,S}^1) (f(X) - f^\star) + \frac{\sum_{i \in S} x_i^2 L_{i,S}^0}{\max_{i \in S} (x_i L_{i,S}^1)} \\ &\leq 4 \max_{i \in S} (x_i L_{i,S}^1) (f(X) - f^\star) + \sum_{i \in S} \frac{x_i L_{i,S}^0}{L_{i,S}^1}. \end{aligned}$$

□