DROP-MUON: UPDATE LESS, CONVERGE FASTER

Anonymous authors
Paper under double-blind review

ABSTRACT

Conventional wisdom in deep learning optimization dictates updating all layers at every step – a principle followed by recent 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 experiments on NanoGPT and controlled CNN studies, 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 & Ba, 2015; Loshchilov & 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 $X = [X_1, \ldots, X_b]$ denote the parameters of a b-layer model, with X_i indexing the parameters of layer $i \in [b] := \{1, \ldots, b\}$. Each of the aforementioned optimizers can be viewed as an instance of the general update rule

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

(see Section B.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 ith layer, M_i^k is a momentum term, and the linear minimization oracle is defined as $\mathrm{LMO}_{\mathcal{B}_i(X_i,t_i)}(M_i) := \arg\min_{Z_i \in \mathcal{B}_i(X_i,t_i)} \langle M_i, Z_i \rangle$. 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.*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Algorithm 1 Drop-Muon

Input: initial iterate $X^0 = [X_1^0, \ldots, X_b^0] \in \mathcal{X}$; momentum $M^0 = [M_1^0, \ldots, M_b^0] \in \mathcal{X}$; stepsizes $\gamma_i^k > 0$; momentum parameters $\beta^k \in [0, 1)$
for $k = 0, \ldots, K-1$ **do**
  Sample $\xi^k \sim \mathcal{P}$ and the set of *active* layers $S^k \sim \mathcal{D}$
  for $i \notin S^k$ **do** ▷ Freeze layers not selected as *active*
    $M_i^k = M_i^{k-1}$
    $X_i^{k+1} = X_i^k$
  end for
  for $i \in S^k$ **do** ▷ Update *active* layers
    Update momentum via $M_i^k = (1-\beta_i)M_i^{k-1} + \beta_i \nabla_i f(X^k; \xi^k)$
    Update parameters via

$$X_i^{k+1} = \mathrm{LMO}_{\mathcal{B}_i(X_i^k, t_i^k)}(M_i^k) = X_i^k - \gamma_i^k \left(M_i^k\right)^\sharp \tag{3}$$

  end for
end for

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

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

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 $X = [X_1, \ldots, X_b]$. In this context, $\mathcal{X}$ is the d-dimensional product space

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

where $d := \sum_{i=1}^b m_i n_i$. Each function $f(\cdot; \xi) : \mathcal{X} \to \mathbb{R}$ 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 ith layer, so that $\nabla f(X) = [\nabla_1 f(X), \ldots, \nabla_b f(X)] \in \mathcal{X}$. Each space $\mathcal{X}_i$ is equipped with the trace inner product, defined as $\langle X_i, Y_i \rangle_{(i)} := \mathrm{tr}(X_i^\top Y_i)$ 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 $\|\cdot\|_{(i)\star}$ be the dual norm associated with $\|\cdot\|_{(i)}$ (i.e., $\|X_i\|_{(i)\star} := \sup_{\|Z_i\|_{(i)} \leq 1} \langle X_i, Z_i \rangle_{(i)}$ 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[1] (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 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),

[1] Algorithm 1 is not limited to the spectral norm used in Muon, but instead supports *arbitrary* norms in the LMO constraint. Consequently, it defines a strictly more general class of algorithms, which we nevertheless collectively term Drop-Muon for simplicity.

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 $\{X_i\}_{i \notin S^k}$ ($X_i^{k+1} = X_i^k \; \forall i \notin S^k$)
5: Update the layers $\{X_i\}_{i \in S^k}$ via

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

6: **end for**

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 $M^\sharp = -\|M\|_\star \, \mathrm{LMO}_{\mathcal{B}(0,1)}(M)$, and hence

$$X_i^{k+1} = X_i^k + t_i^k \mathrm{LMO}_{\mathcal{B}_i(0,1)}(M_i^k) = X_i^k - \frac{t_i^k}{\|M_i^k\|_{(i)\star}} \left(M_i^k\right)^\sharp, \tag{4}$$

which corresponds to a layer-wise normalized *steepest descent* step with stepsize $\gamma_i^k := t_i^k / \|M_i^k\|_{(i)\star}$. 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 & 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 $c_{\mathrm{ov}} \geq 0$ 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

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

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

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

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 (Riabinin 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 D). 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 F.1), which is unlikely in practice. This formally justifies selective layer updates as the compute-optimal default.

5. **Empirical validation.** Experiments on language models, ResNet50, and small CNNs (Sections 5 and H) show that Drop-Muon consistently removes measurable training work. Whether this translates into better time-to-target depends on the task: Drop-Muon reaches target accuracies earlier in the CNN and ResNet50 experiments, and makes each language model training step cheaper.

4 Randomized Progressive Training

The general framework in Algorithm 2 allows virtually any sampling strategy (see Section D). 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 $[\nabla_{s^k} f(X^k), \ldots, \nabla_b f(X^k)]$.

Formally, we can define the sampling distribution $\mathcal{D}$ as follows: at each iteration k, sample $s^k \in [b]$ with probabilities $p_i := \mathbb{P}\left(s^k = i\right)$, where $\sum_{i=1}^b p_i = 1$ and $p_1 > 0$, and set $S^k = \{s^k, \ldots, b\}$. Algorithms 1 and 2 then update the layers $[X^k_{s^k}, \ldots, X^k_b]$, while $[X^k_1, \ldots, X^k_{s^k-1}]$ remain frozen at their previous values (with the convention that $[X^k_1, X^k_0] = \emptyset$ when $s^k = 1$). We refer to this sampling scheme as *Randomized Progressive Training* (RPT), or Drop-training for short (see Section B.4).

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 $f^\star \in \mathbb{R}$ such that $f(X) \geq f^\star$ 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 $L^0 := \{(L^0_{1,S}, \ldots, L^0_{b,S})\}_{S \in \text{supp}(\mathcal{D})}$, $(L^0_{1,S}, \ldots, L^0_{b,S}) \in \mathbb{R}^b_+$, i.e., for any $S \in \text{supp}(\mathcal{D})$,

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

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

We take L^0 to be the collection of the smallest constants satisfying the above. Throughout, we use 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 $\text{supp}(\mathcal{D}) = \{\{j, \ldots, b\}, j \in [b]\}$. By definition, $L^0_{i,S} = 0$ whenever $i \notin S$. Moreover, if $S_1, S_2 \in \text{supp}(\mathcal{D})$ are such that $S_1 \subseteq S_2$, then $L^0_{i,S_1} \leq L^0_{i,S_2}$ (see Lemma C.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 & Takáč, 2014a; Fercoq & Richtárik, 2015; Qu & Richtárik, 2016) in the special case when $\text{supp}(\mathcal{D}) = \{\{1\}, \{2\}, \ldots, \{b\}\}$. 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 $\gamma^k_i = 1/L^0_{i,S^k}$. Then

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

where $w_i := \sum_{s=1}^{i} \frac{p_s}{2L^0_{i,\{s,\ldots,b\}}}$ and $\bar{w} := \frac{1}{b} \sum_{j=1}^{b} w_j$.

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 & 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 $L^0_{i,S}$, but also on the magnitude of the layer's gradient via additional terms $L^1_{i,S} \|\nabla_i f(X)\|_{(i)\star}$.

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 $L^\alpha := \{(L^\alpha_{1,S}, \ldots, L^\alpha_{b,S})\}_{S \in \text{supp}(\mathcal{D})}$, $(L^\alpha_{1,S}, \ldots, L^\alpha_{b,S}) \in \mathbb{R}^b_+$, $\alpha \in \{0, 1\}$, i.e., for any $S \in \text{supp}(\mathcal{D})$,

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

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

Following the methodology of Riabinin et al. (2025) adapted to our layer-wise setting, we estimate these smoothness constants in Section H.6.1.

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 $\gamma^k_i = \left(L^0_{i,S^k} + L^1_{i,S^k} \left\|\nabla_i f(X^k)\right\|_{(i)\star}\right)^{-1}$. Then, to guarantee that

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

it suffices to run the algorithm for

$$K = \left\lceil \frac{2\delta^0 \sum_{i=1}^{b} \frac{\left(\sum_{s=1}^{i} p_s\right)^2 \left(\sum_{s=1}^{i} p_s L^0_{i,\{s,\ldots,b\}}\right)}{\left(\sum_{s=1}^{i} p_s L^1_{i,\{s,\ldots,b\}}\right)^2}}{\varepsilon^2 \bar{w}^2} + \frac{2\delta^0}{\varepsilon \bar{w}} \right\rceil$$

iterations, where $\delta^0 := f(X^0) - f^\star$, $w_i := \frac{\left(\sum_{s=1}^{i} p_s\right)^2}{\sum_{s=1}^{i} p_s L^1_{i,\{s,\ldots,b\}}}$, and $\bar{w} := \frac{1}{b} \sum_{j=1}^{b} w_j$.

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 $(p_1, p_2, \ldots, p_b) = (1, 0, \ldots, 0)$, corresponding to full-network training, the weights simplify to $w_i = 1/L^1_{i,[b]}$, exactly recovering the convergence rate of deterministic Gluon (Riabinin 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.

[2]Theorem 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.

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

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

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

$$\text{cost}_\varepsilon(\mathcal{D}) \overset{(7)}{=} K \times \mathbb{E}\left[\text{cost}(\hat{S})\right] \propto \frac{c_{\text{ov}}+\sum_{i=1}^{b}(c_i+c_i^\sharp)\sum_{s=1}^{i}p_s}{\min_{i\in[b]}\left[\sum\limits_{s=1}^{i}\frac{p_s}{2L^0_{i,\{s,\ldots,b\}}}\right]} \tag{8}$$

(see Section F.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 $(p_1, p_2, \ldots, p_b) = (1, 0, \ldots, 0)$ 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 $(p_1, p_2, \ldots, p_b) = (1, 0, \ldots, 0)$ **if and only if**

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

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 8). The layer-wise (L^0, L^1)–smooth case (Theorem F.1) leads to the same conclusion: **full-network updates are optimal only if the first layer is associated with the largest smoothness constant**–a 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 $L^\alpha := \{(L^\alpha_{1,S}, \ldots, L^\alpha_{b,S})\}_{S\in\text{supp}(\mathcal{D})}$, $(L^\alpha_{1,S}, \ldots, L^\alpha_{b,S}) \in \mathbb{R}^b_+$, $\alpha \in \{0, 1\}$, i.e., for any $S \in \text{supp}(\mathcal{D})$,

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

for all $X = [X_1, \ldots, X_b] \in \mathcal{X}$ and $\Gamma = [\Gamma_1, \ldots, \Gamma_b] \in \mathcal{X}$ 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 I.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., $\mathbb{E}_{\xi\sim\mathcal{P}}\left[\nabla f(X;\xi)\right] = \nabla f(X)$ for all $X \in \mathcal{X}$ and there exist $\sigma_i \geq 0$ such that $\mathbb{E}_{\xi\sim\mathcal{P}}\left[\left\|\nabla_i f(X;\xi) - \nabla_i f(X)\right\|_2^2\right] \leq \sigma_i^2$ for all $X \in \mathcal{X}$ and $i = 1, \ldots, b$.

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 $\underline{\rho}_i \|X_i\|_{(i)} \le \|X_i\|_2 \le \bar{\rho}_i \|X_i\|_{(i)}$ for all $X_i \in \mathcal{X}_i$ (equivalently, $\underline{\rho}_i \|X_i\|_2 \le \|X_i\|_{(i)\star} \le \bar{\rho}_i \|X_i\|_2$).

Theorem 4.4. Let Assumptions 4.1, 4.4, and 4.5 hold. Let $\{X^k\}_{k=0}^K$, $K \ge 0$, 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 $\beta_i \equiv \beta = (K+1)^{-1/2}$ and $0 < \eta_i^2 \le \min\left\{ \frac{(K+1)^{1/2}}{4C_i^1} \left(\sum_{s=1}^b p_s \max_{i\in[b]} L^1_{i,\{s,\dots,b\}} \right)^{-1}, \right.$ $\left. \frac{\underline{\rho}_i p_1}{16\bar{\rho}_i(1-\beta)C_iC_i^1} \left(\sum_{s=1}^b p_s \max_{i\in[b]} L^1_{i,\{s,\dots,b\}} \right)^{-1}, 1 \right\}$. Then

$$\min_{k=0,\dots,K} \sum_{i=1}^b \frac{C_i\eta_i}{C} \mathbb{E}\left[\left\| \nabla_i f(X^k) \right\|_{(i)\star} \right] \le \frac{3\delta^0}{(K+1)^{1/4}C} + \sum_{i=1}^b \frac{6\eta_i\bar{\rho}_i\sigma_i}{(K+1)^{1/2}C} + \sum_{i=1}^b \frac{2\bar{\rho}_i\sigma_iC_i\eta_i}{(K+1)^{1/4}C}$$
$$+ \sum_{i=1}^b \frac{2\bar{\rho}_i\eta_i^2 \left(C_i^0 + C_i^1 \left(\sum_{s=1}^i p_s \frac{L^0_{i,\{s,\dots,b\}}}{L^1_{i,\{s,\dots,b\}}} \right) \right)}{\underline{\rho}_i(K+1)^{1/4}C} + \sum_{i=1}^b \frac{\eta_i^2 \left(C_i^0 + C_i^1 \left(\sum_{s=1}^i p_s \frac{L^0_{i,\{s,\dots,b\}}}{L^1_{i,\{s,\dots,b\}}} \right) \right)}{2(K+1)^{3/4}C},$$

where $C := \frac{1}{b}\sum_{l=1}^b \sum_{s=1}^l p_s\eta_l$, $C_i := \sum_{s=1}^i p_s$, $C_i^0 := \sum_{s=1}^i p_s L^0_{i,\{s,\dots,b\}}$, and $C_i^1 := \sum_{s=1}^i p_s L^1_{i,\{s,\dots,b\}}$.

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 & 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 4.1, our experiments (Section 5) clearly confirm that the approach remains effective in the stochastic setting, too.

5 Experiments

We compare Drop-Muon with full-network Muon under matched training conditions: the same model, data, initialization, batch size, learning-rate schedule, and hardware. For the language models, the learning rate is tuned independently for each method; all other hyperparameters are tuned for Muon and shared (Section H). Drop-Muon applies exactly the same update to the active suffix. Unless noted otherwise, it samples a cutoff using the rule

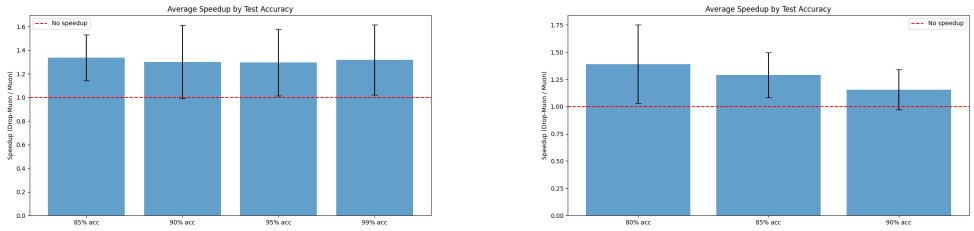
$$p_i = \frac{2(b-i)}{b(b-1)}, \qquad i \in \{1, \dots, b\}. \tag{9}$$

This is the $s = -1$ member of the family defined in Section H. In the language model experiments, Drop-Muon first performs full-network updates during a "late wake-up" phase covering the first 35% (NanoGPT-162M) or 30% (24-layer models) of training; the wake-up fraction is a tuned hyperparameter (see Section H). We report quality against cumulative training time, and time-to-target ratios $T_{\text{Muon}}/T_{\text{Drop-Muon}}$ computed within matched pairs. Full implementation and protocol details appear in Section H.

Small CNNs. Figures 1a and 1b show the ratio $T_{\text{Muon}}/T_{\text{Drop-Muon}}$, comparing the running times of the two algorithms at fixed test-accuracy targets, averaged over repeated paired runs with a constant learning rate. The mean ratios are approximately 1.30–$1.34\times$ on MNIST and 1.15–$1.39\times$ on Fashion-MNIST, with the displayed error bars showing appreciable run-to-run variability.

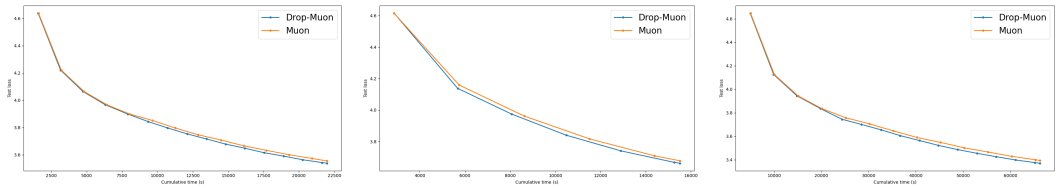
Tiny ImageNet. We train ResNet50 (23.9M parameters) from scratch on Tiny ImageNet with batch size 256 and learning rate 0.02 on one V100 GPU (Figure 3 in Section H). Per epoch, the two methods reach nearly the same training accuracy, but Drop-Muon's epochs are about 20% shorter, so it reaches any given training accuracy earlier in wall-clock time.

Language modeling. We train GPT-style models with 162M, 247M, and 405M parameters on FineWeb10B (Figure 2). Over the displayed range, Drop-Muon reaches a lower test loss than Muon at the same cumulative training time.



(a) MNIST (b) Fashion-MNIST

Figure 1: Average time-to-target speedup of Drop-Muon over Muon.



(a) NanoGPT-162M (b) GPT-24L-247M (c) GPT-24L-405M

Figure 2: Test losses of Drop-Muon and Muon against cumulative training time. Both methods process the same data for the same number of steps; each panel ends when Drop-Muon completes its run.

Where the speedup comes from. Synchronized component profiling of NanoGPT-162M on four V100 GPUs (global batch 256, dropping active from the first step, three paired seeds) confirms the intended mechanism: the forward pass is essentially unchanged, while backward computation, Newton–Schulz orthogonalization, optimizer updates, and gradient communication become cheaper for inactive prefixes. Clean step time falls from 6.482 to 5.646 seconds (12.9%), and a forced-cutoff model predicts stochastic Drop-Muon latency within about 1.2%.

Overall, randomized suffix updates remove measurable training work. In the CNN and ResNet50 experiments, this translates into faster time-to-accuracy. Better sampling schedules and task-specific tuning remain important directions.

AI USE STATEMENT

In this work, we used generative AI tools for refining the prose, designing or providing feedback on research methodology or experiments and implementing methods. We have not used generative AI tools for generating synthetic data sets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, assisting with translation, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, and interpreting results. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pp. 560–569. PMLR, 2018. URL https://arxiv.org/abs/1802.04434. (Cited on page 17)

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 16)

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 18)

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*, pp. 5396–5427. PMLR, 2023b. URL https://arxiv.org/abs/2303.02854. (Cited on page 6 and 17)

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 17)

Ashok Cutkosky and Harsh Mehta. Momentum improves normalized SGD. In *International conference on machine learning*, pp. 2260–2268. PMLR, 2020. URL https://arxiv.org/abs/2002.03305. (Cited on page 3, 8, and 51)

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 18)

Yury Demidovich, Grigory Malinovsky, Egor Shulgin, and Peter Richtárik. MAST: Model-agnostic sparsified training. *arXiv preprint arXiv:2311.16086*, 2023. (Cited on page 18)

Werner Dinkelbach. On nonlinear fractional programming. *Management Science*, 13:492–498, 1967. URL https://api.semanticscholar.org/CorpusID:119939254. (Cited on page 30)

Chen Dun, Cameron R Wolfe, Christopher M Jermaine, and Anastasios Kyrillidis. ResIST: Layer-wise decomposition of resnets for distributed training. In *Uncertainty in Artificial Intelligence*, pp. 610–620. PMLR, 2022. (Cited on page 18)

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 18)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. (Cited on page 18)

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 17)

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 16)

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 18)

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*, pp. 217–226. SIAM, 2014. URL https://arxiv.org/abs/1304.2338. (Cited on page 2)

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)

TN Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. (Cited on page 18)

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 16)

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 17)

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 17)

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 16)

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 2, 17, and 18)

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 18)

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 16)

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 17)

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 18)

Artem Riabinin, Egor Shulgin, Kaja Gruntkowska, 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, 6, 7, 8, 16, 17, 25, 65, and 67)

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 17 and 18)

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 18)

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)

Egor Shulgin and Peter Richtárik. Towards a better theoretical understanding of independent subnetwork training. *arXiv preprint arXiv:2306.16484*, 2023. (Cited on page 18)

Richard V. Southwell. *Relaxation Methods in Engineering Science: A Treatise on Approximate Computation*. Read Books, 1940. (Cited on page 17)

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 (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33077–33099. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/sun23l.html. (Cited on page 8)

Rafał Szlendak, Alexander Tyurin, and Peter Richtárik. Permutation compressors for provably faster distributed nonconvex optimization. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2110.03300. (Cited on page 18)

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*, pp. 2161–2169. PMLR, 2024. URL https://arxiv.org/abs/2306.03626. (Cited on page 4 and 18)

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 (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23034–23054. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wang22y.html. (Cited on page 18)

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)

Cameron R Wolfe, Jingkang Yang, Fangshuo Liao, Arindam Chowdhury, Chen Dun, Artun Bayer, Santiago Segarra, and Anastasios Kyrillidis. GIST: Distributed training for large-scale graph convolutional networks. *Journal of Applied and Computational Topology*, 8(5):1363–1415, 2024. (Cited on page 18)

Stephen J. Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, 2015. URL https://arxiv.org/abs/1502.04759. (Cited on page 5, 17, and 18)

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 17)

Binhang Yuan, Cameron R Wolfe, Chen Dun, Yuxin Tang, Anastasios Kyrillidis, and Christopher M Jermaine. Distributed learning of deep neural networks using independent subnet training. *arXiv preprint arXiv:1910.02120*, 2019. (Cited on page 18)

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 6 and 17)

A 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 . . . 7

4.2 Iteration Complexity – Stochastic Gradient Setting . . . 7

5 Experiments **8**

A Appendix **14**

B Additional Literature Review **16**

B.1 Muon (and Friends) . . . 16

B.2 Generalized Smoothness . . . 17

B.3 Randomized Block Coordinate Descent . . . 17

B.4 Prior Work on Progressive Training . . . 18

B.5 Independent Subnet Training . . . 18

C More on Smoothness Assumptions **20**

D Arbitrary Sampling **22**

E Convergence Results – Deterministic Gradient Case **23**

E.1 Layer-wise Smooth Case . . . 23

E.2 Layer-wise Generalized Smooth Case . . . 24

F Optimizing the Cost – Deterministic Gradient Case **28**

F.1 Smooth case . . . 29

F.1.1 Randomized Progressive Training . . . 30

F.1.2 τ-nice Sampling . . . 32

F.1.3 τ-submodel Sampling . . . 35

F.1.4 Arbitrary Submodel Sampling . . . 38

F.2 Generalized Smooth Case . . . 41

F.2.1 Randomized Progressive Training . . . 43

F.2.2 τ-nice Sampling 46
F.2.3 τ-submodel Sampling 47
F.2.4 Arbitrary Submodel Sampling 49

G Convergence Results – Stochastic Gradient Case **50**

H Experiments **63**
H.1 Experimental design 63
H.2 Drop-Muon implementation 63
H.3 Layer-sampling schedule 63
H.4 Metrics 63
H.5 Vision experiments 64
H.6 Language modeling 64
H.6.1 Estimating the layer-wise smoothness constants 65
H.7 Component profiling 65
H.8 Interpretation and limitations 66

I Useful Facts **67**

B Additional Literature Review

B.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 (Riabinin 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 & Hutter, 2019)). Given a layer X_i and the associated (stochastic) gradient G_i, the update is obtained by solving the constrained optimization problem

$$\min_{\Delta X_i} \langle G_i, \Delta X_i \rangle \quad \text{subject to} \quad \|\Delta X_i\|_{2\to 2} \le t_i, \tag{10}$$

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

$$\Delta X_i = -t_i U_i V_i^\top, \qquad X_i^{k+1} = X_i^k + \Delta X_i. \tag{11}$$

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 & Bowie, 1971) to approximate the orthogonalization. In practice, the algorithm also incorporates momentum, yielding the update

$$M_i^k = (1-\beta_i) M_i^{k-1} + \beta_i G_i^k,$$
$$X_i^{k+1} = X_i^k - t_i^k \mathrm{NewtonSchulz}(M_i^k),$$

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 (11) 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

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

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

In particular, Muon's update in (10) corresponds to an LMO over the spectral norm ball $\mathcal{B}_i^{2\to 2}(0, t_i) := \{Z_i \in \mathcal{X}_i : \|Z_i\|_{2\to 2} \le t_i\}$, leading to the equivalent update

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

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 $\|\cdot\|_{1\to\infty}$ norm for embedding and output layers).

Finally, Gluon (Riabinin 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 B.2). As such, Gluon can be seen as the theoretical foundation unifying the entire class of LMO-based optimizers for training deep networks.

Algorithm 3 Randomized Block Coordinate Descent

1: **Input:** Initial iterate $X^0 = [X_1^0, \ldots, X_b^0] \in \mathcal{X}$; stepsizes $\gamma_i^k > 0$, $i \in [b]$, $k \geq 0$
2: **for** $k = 0, \ldots, K-1$ **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**

B.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 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

$$\|\nabla_i f(X) - \nabla_i f(Y)\|_{(i)\star} \leq \left(L_i^0 + L_i^1 \|\nabla_i f(X)\|_{(i)\star}\right) \|X_i - Y_i\|_{(i)} \tag{12}$$

for all $i \in [b]$ and all $X = [X_1, \ldots, X_p] \in \mathcal{X}$, $Y = [Y_1, \ldots, Y_p] \in \mathcal{X}$. Condition (12) 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 (12), 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 & 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).

B.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 $\{\mathbf{U}_i : \mathcal{X}_i \to \mathcal{X} : i \in [b]\}$ (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:

$$\|\nabla_i f(X + \mathbf{U}_i(\Gamma_i)) - \nabla_i f(X)\|_{(i)\star} \leq L_i \|\Gamma_i\|_{(i)}, \quad \forall i \in [b],\ X \in \mathcal{X},\ \Gamma_i \in \mathcal{X}_i. \tag{13}$$

From (13), one obtains the standard quadratic upper bound

$$f(X + \mathbf{U}_i(\Gamma_i)) - f(X) - \langle \nabla_i f(X), \Gamma_i \rangle \leq \frac{L_i}{2} \|\Gamma_i\|_{(i)}^2. \tag{14}$$

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 (13) 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 & Takáč (2016); Qu & Richtárik (2016). For the nonconvex case considered in this paper, iteration complexity results of Algorithm 2 are only known for serial sampling (Patrascu & Necoara, 2015; Dang & 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.4 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 & 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.

B.5 Independent Subnet Training

A separate RPT-related line of work has developed around the idea of *Independent Subnet Training* (IST). The main idea of IST is to decompose a large neural network into multiple submodels, train these submodels independently, typically in parallel across multiple devices, and periodically aggregate their parameters to update the full model. This paradigm was formalized by Yuan et al. (2019), who introduced the method together with an accompanying convergence analysis. While IST has demonstrated strong empirical performance, its theoretical guarantees rely on restrictive assumptions. In particular, the analysis of Yuan et al. (2019) assumes strong regularity properties of the objective, including Lipschitz continuity and certain bounds on the stochastic gradient noise. Further analyses of IST arise as special cases of the framework proposed by Demidovich et al. (2023). Moreover, Shulgin & Richtárik (2023) analyzed the convergence of IST for quadratic objectives using permutation-based sketches (Szlendak et al., 2021).

Variants of the IST paradigm have also been explored in applied settings. For example, Dun et al. (2022) adapted a similar strategy for training residual networks (He et al., 2016), while Wolfe et al. (2024) employed it for large-scale graph convolutional networks (Kipf, 2016).

Despite several similarities, IST and RPT differ in several fundamental respects. The IST framework partitions the model into multiple sparse components that are optimized in parallel, followed by a synchronization step that aggregates the independently trained submodels. In contrast, RPT performs a single update per iteration, modifying a randomly selected subnetwork. Moreover, IST is naturally a distributed method, whereas our work focuses on the single-machine setting.

C More on Smoothness Assumptions

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

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

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

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

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

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

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

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

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

Therefore, one can always choose $L^0_{i,S_1} \leq L^0_{i,S_2}$. □

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

$$L^0_{i,S} \leq C(|S|) L^0_{i,\{i\}} \tag{15}$$

holds for all f. Consider the case $b = 2$ with scalar blocks and the Euclidean norm, suppose that $\mathrm{supp}(\mathcal{D}) = \{\{1\}, \{2\}, \{1, 2\}\}$, 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 $L^0_{1,\{1\}} = 0$. By symmetry, $L^0_{2,\{2\}} = 0$ 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 (15) holds for some function $C : [b] \to \mathbb{R}$. Then, since $L^0_{1,\{1\}} = 0$, it must be that $L^0_{1,\{1,2\}} = 0$. By Assumption 4.2, this requires

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

But the right-hand side is independent of γ_1, whereas the left-hand side grows without bound as $\gamma_1 \to \infty$ whenever $\gamma_2 \neq 0$. Hence no such constant $L^0_{2,\{1,2\}}$ can exist. This proves that no uniform bound of the form (15) can hold without additional assumptions.

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

$$L^0_{i,S_2} \leq C(|S_1|, |S_2|) L^0_{i,S_1}. \tag{16}$$

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 $f : \mathbb{R}^b \to \mathbb{R}$ 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 $L^0_{i,S_1} = 0$ 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

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

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 (16) holds with some finite $C(|S_1|, |S_2|)$. Since $L^0_{i,S_1} = 0$ for every $i \in S_1$, the inequality forces $L^0_{i,S_2} = 0$ for all $i \in S_1$. But then Assumption 4.2 would yield

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

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

D 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 E.1, E.2 and G.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

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

with support $\operatorname{supp}(\mathcal{D}) := \{S \in \mathfrak{P}([b]) : \mathcal{D}(S) > 0\}$, 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 G.1) and 2 (Theorems E.1 and E.2). 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 4, 6, and 10.

E CONVERGENCE RESULTS – DETERMINISTIC GRADIENT CASE

E.1 LAYER-WISE SMOOTH CASE

Theorem E.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 $\gamma_i^k = 1/L^0_{i,S^k}$. Then

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

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

Remark 3. *For Theorem E.1 to be meaningful, the sampling must ensure that $w_i > 0$ for every $i \in [b]$, i.e.,*

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

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 $\mathbb{P}\left(i\in\hat{S}\right)=0$, then the expectation vanishes, so $w_i>0$ necessarily implies $\mathbb{P}\left(i\in\hat{S}\right)>0$ for all $i\in[b]$.

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

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

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

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

$$\begin{aligned}
& f(X^{k+1}) \\
\leq\ & f(X^k) + \left\langle \nabla f(X^k), X^{k+1}-X^k\right\rangle + \sum_{i\in S^k}\frac{L^0_{i,S^k}}{2}\left\|X_i^k - X_i^{k+1}\right\|^2_{(i)} \\
=\ & f(X^k) + \sum_{i\in S^k}\left(\left\langle \nabla_i f(X^k), X_i^{k+1}-X_i^k\right\rangle_{(i)} + \frac{L^0_{i,S^k}}{2}\left\|X_i^k - X_i^{k+1}\right\|^2_{(i)}\right) \\
=\ & f(X^k) + \sum_{i\in S^k}\left(-\gamma_i^k\left\langle \nabla_i f(X^k), \left(\nabla_i f(X^k)\right)^\sharp\right\rangle_{(i)} + \frac{L^0_{i,S^k}(\gamma_i^k)^2}{2}\left\|\left(\nabla_i f(X^k)\right)^\sharp\right\|^2_{(i)}\right) \\
\overset{(53),(54)}{=}\ & f(X^k) + \sum_{i\in S^k}\left(-\gamma_i^k\left\|\nabla_i f(X^k)\right\|^2_{(i)\star} + \frac{L^0_{i,S^k}(\gamma_i^k)^2}{2}\left\|\nabla_i f(X^k)\right\|^2_{(i)\star}\right).
\end{aligned}$$

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

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

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

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

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

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

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

□

E.2 Layer-wise Generalized Smooth Case

Theorem E.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 $\gamma_i^k = \left(L^0_{i,S^k} + L^1_{i,S^k}\left\|\nabla_i f(X^k)\right\|_{(i)\star}\right)^{-1}$. Then, to guarantee that

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

it suffices to run the algorithm for

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

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

Remark 5. *The guarantee in Theorem E.2 is meaningful only if $w_i > 0$ for all $i \in [b]$, which is equivalent to requiring that $\mathbb{P}\left(i\in\hat{S}\right) > 0$ for all $i\in[b]$.*

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

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

and

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

for $\alpha\in\{0,1\}$. Hence the weights from Theorem E.2 become

$$
\frac{\mathbb{P}\left(i\in\hat{S}\right)}{\mathbb{E}\left[L^1_{i,\hat{S}}\middle| i\in\hat{S}\right]} = \frac{\left(\sum_{s=1}^{i} p_s\right)^2}{\sum_{s=1}^{i} p_s L^\alpha_{i,\{s,\ldots,b\}}}.
$$

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

Remark 7. *When $\hat{S} = [b]$ with probability 1, the weights become $w_i = {}^1/_{L^1_{i,[b]}}$, and hence Theorem E.2 guarantees that*

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

after

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

iterations, recovering the rate of Gluon *(Riabinin et al., 2025).*

Proof of Theorem E.2. Starting with Assumption 4.3, we have

$$\begin{aligned}
& f(X^{k+1}) \\
\leq\ & f(X^k) + \sum_{i\in S^k} \Bigg(\left\langle \nabla_i f(X^k), X_i^{k+1} - X_i^k \right\rangle_{(i)} \\
& \qquad + \frac{L^0_{i,S^k} + L^1_{i,S^k} \left\|\nabla_i f(X^k)\right\|_{(i)\star}}{2} \left\|X_i^{k+1} - X_i^k\right\|^2_{(i)} \Bigg) \\
=\ & f(X^k) + \sum_{i\in S^k} \Bigg(-\gamma_i^k \left\langle \nabla_i f(X^k), \left(\nabla_i f(X^k)\right)^\sharp \right\rangle_{(i)} \\
& \qquad + \frac{L^0_{i,S^k} + L^1_{i,S^k} \left\|\nabla_i f(X^k)\right\|_{(i)\star}}{2} (\gamma_i^k)^2 \left\|\left(\nabla_i f(X^k)\right)^\sharp\right\|^2_{(i)} \Bigg) \\
\overset{(53),(54)}{=}\ & f(X^k) + \sum_{i\in S^k} \Bigg(-\gamma_i^k \left\|\nabla_i f(X^k)\right\|^2_{(i)\star} \\
& \qquad + \frac{L^0_{i,S^k} + L^1_{i,S^k} \left\|\nabla_i f(X^k)\right\|_{(i)\star}}{2} (\gamma_i^k)^2 \left\|\nabla_i f(X^k)\right\|^2_{(i)\star} \Bigg).
\end{aligned}$$

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

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

and hence

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

Taking expectation

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

$$
\begin{aligned}
&= \sum_{k=0}^{K-1}\sum_{i=1}^{b}\mathbb{E}\left[\mathbb{P}\left(i\in S^k\right)\mathbb{E}\left[\frac{\left\|\nabla_i f(X^k)\right\|^2_{(i)\star}}{2\left(L^0_{i,S^k}+L^1_{i,S^k}\left\|\nabla_i f(X^k)\right\|_{(i)\star}\right)}\middle| X^k,\left\{i\in S^k\right\}\right]\right]\\
&\overset{(i)}{\geq} \sum_{k=0}^{K-1}\sum_{i=1}^{b}\mathbb{E}\left[\frac{\left\|\nabla_i f(X^k)\right\|^2_{(i)\star}\mathbb{P}\left(i\in S^k\right)}{2\left(\mathbb{E}\left[L^0_{i,S^k}\middle| X^k,\left\{i\in S^k\right\}\right]+\mathbb{E}\left[L^1_{i,S^k}\middle| X^k,\left\{i\in S^k\right\}\right]\left\|\nabla_i f(X^k)\right\|_{(i)\star}\right)}\right]\\
&\overset{(ii)}{=} \frac{1}{2}\sum_{k=0}^{K-1}\sum_{i=1}^{b}\mathbb{E}\left[\frac{\left\|\nabla_i f(X^k)\right\|^2_{(i)\star}\mathbb{P}\left(i\in \hat{S}\right)}{\mathbb{E}\left[L^0_{i,\hat{S}}\middle| i\in\hat{S}\right]+\mathbb{E}\left[L^1_{i,\hat{S}}\middle| i\in\hat{S}\right]\left\|\nabla_i f(X^k)\right\|_{(i)\star}}\right]\\
&\overset{(iii)}{\geq} \frac{1}{2}\sum_{k=0}^{K-1}\sum_{i=1}^{b}\frac{\mathbb{E}\left[\left\|\nabla_i f(X^k)\right\|_{(i)\star}\right]^2\mathbb{P}\left(i\in\hat{S}\right)}{\mathbb{E}\left[L^0_{i,\hat{S}}\middle| i\in\hat{S}\right]+\mathbb{E}\left[L^1_{i,\hat{S}}\middle| i\in\hat{S}\right]\mathbb{E}\left[\left\|\nabla_i f(X^k)\right\|_{(i)\star}\right]},
\end{aligned}
\tag{17}
$$

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 $t\mapsto\frac{t^2\mathbb{P}\left(i\in S^k\right)}{\mathbb{E}\left[L^0_{i,\hat{S}}\middle| i\in\hat{S}\right]+\mathbb{E}\left[L^1_{i,\hat{S}}\middle| i\in\hat{S}\right]t}$ and Jensen's inequality.

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

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

Applying this in (17), we get

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

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

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

$$\geq \frac{K}{2}\psi\left(\min_{k=0,\ldots,K-1}\sum_{i=1}^{b}\frac{\mathbb{P}\left(i\in\hat{S}\right)}{\mathbb{E}\left[L^1_{i,\hat{S}}\middle| i\in\hat{S}\right]}\mathbb{E}\left[\left\|\nabla_i f(X^k)\right\|_{(i)\star}\right]\right).$$

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

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

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

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

it suffices to run the algorithm for

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

iterations. □

F 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

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

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

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

and

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

the expected cost of one iteration is

$$\mathbb{E}\left[\text{cost}(\hat{S})\right] = c_{\text{ov}} + \sum_{i=1}^{b} c_i F_i + \sum_{i=1}^{b} c_i^\sharp Q_i. \tag{18}$$

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 F.1) and the generalized layer-wise smooth setting (see Section F.2).

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

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

   Therefore,

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

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

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

   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

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

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

$$F_i = \mathbb{P}(\hat{s} \le i) = \sum_{j=1}^{\min\{i, b-\tau+1\}} p_j,$$

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

The expected per-iteration cost is therefore

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

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 $\underline{b}_k := \min B_k$. Then

$$F_i = \mathbb{P}(\hat{s} \le i) = \sum_{j : \underline{b}_j \le i} p_j,$$

$$Q_i = \mathbb{P}\left(i \in \hat{S}\right) = p_{k(i)}.$$

The expected cost per iteration is

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

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

F.1 Smooth case

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

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

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

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

iterations. Now, recall from (18) that the expected cost of a single iteration is

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

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

$$\begin{aligned}\mathrm{cost}_\varepsilon(\mathcal{D}) &= K \times \mathbb{E}\left[\mathrm{cost}(\hat{S})\right] \\ &= K \times \left(c_{\mathrm{ov}} + \sum_{i=1}^{b} c_i \mathbb{P}\left(\min \hat{S} \leq i\right) + \sum_{i=1}^{b} c_i^\sharp \mathbb{P}\left(i \in \hat{S}\right)\right) \\ &\propto \frac{c_{\mathrm{ov}} + \sum_{i=1}^{b} c_i \mathbb{P}\left(\min \hat{S} \leq i\right) + \sum_{i=1}^{b} c_i^\sharp \mathbb{P}\left(i \in \hat{S}\right)}{\min_{i\in[b]} \mathbb{E}\left[\frac{\mathbb{I}\left(i\in\hat{S}\right)}{2L^0_{i,\hat{S}}}\right]},\end{aligned}$$

and the cost minimization problem to be solved is

$$\min_{\mathcal{D}:\mathfrak{P}([b])\to[0,1],\sum_{S\subseteq[b]}\mathcal{D}(S)=1} \frac{c_{\mathrm{ov}} + \sum_{i=1}^{b} c_i \mathbb{P}\left(\min \hat{S} \leq i\right) + \sum_{i=1}^{b} c_i^\sharp \mathbb{P}\left(i \in \hat{S}\right)}{\min_{i\in[b]} \mathbb{E}\left[\frac{\mathbb{I}\left(i\in\hat{S}\right)}{2L^0_{i,\hat{S}}}\right]}. \tag{19}$$

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.

F.1.1 Randomized Progressive Training

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

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

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

and hence

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

Combining it with the fact that

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

we get

$$\mathrm{cost}_\varepsilon(\mathcal{D}) \propto \frac{\mathbb{E}\left[\mathrm{cost}(\hat{S})\right]}{\min_{i\in[b]} \mathbb{E}\left[\frac{\mathbb{I}\left(i\in\hat{S}\right)}{2L^0_{i,\hat{S}}}\right]} = \frac{c_{\mathrm{ov}} + \sum_{i=1}^{b}(c_i + c_i^\sharp)\sum_{j=1}^{i} p_j}{\min_{i\in[b]}\left\{\sum_{s=1}^{i} \frac{p_s}{2L^0_{i,\{s,\ldots,b\}}}\right\}}. \tag{20}$$

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

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

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

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

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

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

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

$$d_1 > d_2 > \ldots > d_b \tag{21}$$

and

$$\delta_{i,1} \leq \delta_{i,2} \leq \ldots \leq \delta_{i,i} \qquad \forall i \in [b]. \tag{22}$$

Based on (20), the search for optimal probabilities reduces to solving the following linear fractional program:

$$\begin{aligned} \min_{p,t} \quad & \frac{d_1 p_1 + \ldots + d_b p_b}{t} \\ \text{s.t.} \quad & p_1, \ldots, p_b \geq 0 \\ & p_1 + \ldots + p_b = 1 \\ & t \geq 0 \\ & t \leq \delta_{1,1} p_1 \\ & \vdots \\ & t \leq \delta_{i,1} p_1 + \ldots + \delta_{i,i} p_i \\ & \vdots \\ & t \leq \delta_{b,1} p_1 + \ldots + \delta_{b,b} p_b. \end{aligned} \tag{23}$$

This program can be written equivalently as

$$\begin{aligned} \min_{q} \quad & d_1 q_1 + \ldots + d_b q_b \\ \text{s.t.} \quad & q_1, \ldots, q_b \geq 0 \\ & \delta_{1,1} q_1 \geq 1 \\ & \vdots \\ & \delta_{i,1} q_1 + \ldots + \delta_{i,i} q_i \geq 1 \\ & \vdots \\ & \delta_{b,1} q_1 + \ldots + \delta_{b,b} q_b \geq 1. \end{aligned} \tag{24}$$

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 F.1. Let q be an optimal point of (24) and fix $i \in [b-1]$. If $\sum_{s=1}^{i} \delta_{i,s} q_s > 1$, then $q_i = 0$. Equivalently,

$$q_i > 0 \qquad \Longrightarrow \qquad \sum_{s=1}^{i} \delta_{i,s} q_s = 1. \tag{25}$$

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 $q_i = 0$. □

Let us now use Lemma F.1 to derive a recursive construction of the optimal probabilities. Let $q^\star = (q_1^\star, \ldots, q_b^\star)$ be a solution of (24). As established in (25), any positive coordinate forces its constraint to be tight. Constraint 1 is $\delta_{1,1} q_1^\star \geq 1$, meaning that $q_1^\star > 0$, and so $\delta_{1,1} q_1^\star = 1$. Thus

$$q_1^\star = \frac{1}{\delta_{1,1}} = 2L_{1,\{1,\ldots,b\}}.$$

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

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

Define the residual

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

If $r_i^\star \leq 0$, then $(q_1^\star, \ldots, q_{i-1}^\star)$ already satisfy the constraint, so by (25), we must have $q_i^\star = 0$. If $r_i^\star > 0$, then for $(q_1^\star, \ldots, q_i^\star)$ to satisfy the constraint, we need $q_i^\star > 0$, and hence by Lemma F.1, we have $\sum_{s=1}^{i} \delta_{i,s} q_s^\star = 1$, meaning that

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

Combining the above yields the recursion

$$q_1^\star = 2L_{1,\{1,\ldots,b\}},$$

$$q_i^\star = 2[r_i^\star]_+ L_{i,\{i,\ldots,b\}}, \qquad r_i^\star = 1 - \sum_{s=1}^{i-1} \frac{q_s^\star}{2L_{i,\{s,\ldots,b\}}}, \quad i \in \{2, \ldots, b\},$$

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

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

Remark 8. *Note that $(p_1^\star, p_2^\star, \ldots, p_b^\star) = (1, 0, \ldots, 0)$ if and only if $q_1^\star = \sum_{j=1}^{b} q_j^\star$, i.e., $q_2^\star = \ldots = q_b^\star = 0$. But for that to be the case, we need $r_i^\star \leq 0$ for all $i \in \{2, \ldots, b\}$, so*

$$1 \leq \sum_{s=1}^{i-1} \delta_{i,s} q_s^\star = \delta_{i,1} q_1^\star = \frac{\delta_{i,1}}{\delta_{1,1}} = \frac{L_{1,\{1,\ldots,b\}}}{L_{i,\{1,\ldots,b\}}} \qquad \forall i \in \{2, \ldots, b\}.$$

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

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

which proves Theorem 4.3.

F.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

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

and hence

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

In general there is no closed-form expression for τ minimizing this cost because the denominator depends on $\{L^0_{i,S}\}_{S\subseteq[b]}$ 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

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

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

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

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

Define

$$A(\tau) := \max_{i\in[b]} L^0_{i,\tau}, \qquad B(\tau) := \frac{b}{\tau}c_{\mathrm{ov}} + \sum_{j=1}^b c_j \left(\frac{b}{\tau} - \frac{\binom{b-j}{\tau}}{\binom{b-1}{\tau-1}}\right) + \sum_{j=1}^b c_j^\sharp.$$

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

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

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,

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

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 $L^0_{i,\tau}$ scales linearly with τ, i.e, $A(\tau) = \max_{i\in[b]} \tau L^0_i$ for some $L^0_i \geq 0$. Then

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

Now, consider the function

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

Note that

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

(with the convention that the right-hand side is 0 when $b-j-\tau<0$). Moreover, the increment is strictly positive if either $\sum_{j=1}^{b} c_j^\sharp > 0$ 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 $\tau^\star = 1$.

F.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

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

Hence the total expected cost is proportional to

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

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

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

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

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

Plugging this choice into the general submodel expressions immediately gives

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

Therefore the full cost reduces to

$$\begin{aligned}\mathrm{cost}_\varepsilon(\mathcal{D}) &\propto \frac{c_{\mathrm{ov}} + \sum_{j=1}^b c_j \left(\sum_{k=1}^{\min\{m, \lfloor (j-1)/\tau \rfloor + 1\}} p_{s_k}\right) + \sum_{j=1}^b c_j^\sharp p_{s_{\lceil j/\tau \rceil}}}{\min_{i\in[b]} \left\{ \frac{p_{s_{\lceil i/\tau \rceil}}}{2L^0_{i,B_{\lceil i/\tau \rceil}}} \right\}} \\ &= \frac{c_{\mathrm{ov}} + \sum_{j=1}^b c_j \left(\sum_{k=1}^{\lceil j/\tau \rceil} p_{s_k}\right) + \sum_{j=1}^b c_j^\sharp p_{s_{\lceil j/\tau \rceil}}}{\min_{i\in[b]} \left\{ \frac{p_{s_{\lceil i/\tau \rceil}}}{2L^0_{i,B_{\lceil i/\tau \rceil}}} \right\}}. \end{aligned} \tag{26}$$

Now, note that

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

and

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

Hence

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

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

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

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

and set

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

Then the expected cost can be represented as

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

This corresponds to the linear fractional program

$$\begin{aligned} \min_{p,t} \quad & \frac{d_1 p_{s_1} + \cdots + d_m p_{s_m}}{t} \\ \text{s.t.} \quad & p_{s_1}, \ldots, p_{s_m} \geq 0, \\ & p_{s_1} + \cdots + p_{s_m} = 1, \\ & t \geq 0, \\ & t \leq \delta_{i,k} p_{s_k}, \qquad k \in [m], i \in B_k. \end{aligned} \tag{27}$$

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

$$\begin{aligned} \min_{q} \quad & d_1 q_1 + \cdots + d_m q_m \\ \text{s.t.} \quad & q_1, \ldots, q_m \geq 0, \\ & \delta_{i,k} q_k \geq 1 \qquad k \in [m], i \in B_k. \end{aligned} \tag{28}$$

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

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

Hence (28) is separable and its optimal solution is

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

with optimal objective value

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

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 (28) is

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

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

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

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 $q^\star$ is optimal.

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

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

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

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

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.,

$$c_j \equiv c, \qquad c_j^\sharp \equiv c^\sharp,$$

in which case

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

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

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

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 $\tau^\star = b$.
- If $L^0(\tau)$ grows sublinearly, then $L^0(\tau)\frac{b}{\tau}\left(c_{\mathrm{ov}} + \frac{bc}{2}\right)$ is decreasing in τ, while $L^0(\tau)\left(\frac{cb}{2} + bc^\sharp\right)$ 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 $\tau^\star = 1$.

F.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 $\underline{b}_k := \min B_k$.

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

$$\begin{aligned}\mathbb{E}\left[\mathrm{cost}(\hat{S})\right] &= c_{\mathrm{ov}} + \sum_{i=1}^{b} c_i \left(\sum_{k:\underline{b}_k \le i} p_k\right) + \sum_{i=1}^{b} c_i^\sharp p_{k(i)} \\ &= c_{\mathrm{ov}} + \sum_{k=1}^{m} \sum_{i \ge \underline{b}_k} c_i p_k + \sum_{k=1}^{m} \sum_{i \in B_k} c_i^\sharp p_k \\ &= \sum_{k=1}^{m} d_k p_k,\end{aligned}$$

where $d_k := c_{\mathrm{ov}} + \sum_{j \ge \underline{b}_k} c_j + \sum_{j \in B_k} c_j^\sharp$. We also have

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

and hence

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

Now, define

$$\delta_{i,k} := \tfrac{1}{2L^0_{i,B_k}}, \qquad \delta_k^{\min} := \min_{i\in B_k} \delta_{i,k}.$$

By the same linear fractional reduction as in (27), the unique optimal solution is

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

With this choice, the objective is

$$\mathrm{cost}_\varepsilon(\mathcal{D}) \quad \propto \quad 2\sum_{k=1}^{m} d_k \max_{i\in B_k} L^0_{i,B_k} = 2\sum_{k=1}^{m} \left(c_{\mathrm{ov}} + \sum_{j\ge\underline{b}_k} c_j + \sum_{j\in B_k} c_j^\sharp\right) \max_{i\in B_k} L^0_{i,B_k}. \tag{29}$$

Let us compare it with the cost for full model training

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

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*

$$c_j \equiv \underline{c}_k, \qquad c_j^\sharp \equiv \underline{c}_k^\sharp.$$

Then

$$\mathrm{cost}_\varepsilon(\mathcal{D}) \quad \propto \quad 2\sum_{k=1}^{m} \left(c_{\mathrm{ov}} + \sum_{j\ge\underline{b}_k} c_j + \sum_{j\in B_k} c_j^\sharp\right) \max_{i\in B_k} L^0_{i,B_k}$$

$$= \quad 2\sum_{k=1}^{m}\left(c_{\text{ov}} + \sum_{j\geq \underline{b}_k} c_j + |B_k|\underline{c}_k^{\sharp}\right)\max_{i\in B_k} L_{i,B_k}^0$$

and

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

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

$$\sum_{k=1}^{m}\max_{i\in B_k} L_{i,B_k}^0\left(c_{\text{ov}} + \sum_{j\geq \underline{b}_k} c_j + |B_k|\underline{c}_k^{\sharp}\right) \leq \max_{i\in[b]} L_{i,[b]}^0\left(c_{\text{ov}} + \sum_{k=1}^{m}|B_k|\left(\underline{c}_k + \underline{c}_k^{\sharp}\right)\right). \tag{30}$$

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* (30) *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*

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

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

Example 2 (Grouping layers across transformer blocks)**.** *In their general forms,* (29) *and* (30) *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* $l \in \{1,\ldots,L\}$ *inside transformer block* $t \in [T]$ *via* $j = l + (t-1)L$*.*

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

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

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 is 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

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

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

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

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

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

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

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

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

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

Comparing the two, partitioned sampling is better if

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

Now, let us denote

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

Then (31) is equivalent to

$$\sum_{l=1}^{L} L_l^0 \left(c_{\text{ov}} + \sum_{j=l}^{L} \underline{c}_j \right) + T \sum_{l=1}^{L} L_l^0 \left(\frac{T-1}{T} \underline{C} + \underline{c}_l^\sharp \right) \leq T \left(\underline{C} + \underline{C}^\sharp \right) \max_{i \in [b]} L_{i,[b]}^0 + c_{\text{ov}} \max_{i \in [b]} L_{i,[b]}^0. \tag{32}$$

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

$$\sum_{l=1}^{L} L_l^0 \left(\underline{C} + \underline{c}_l^\sharp \right) \leq \max_{i \in [b]} L_{i,[b]}^0 (\underline{C} + \underline{C}^\sharp), \tag{33}$$

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

F.2 Generalized Smooth Case

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

$$\min_{k=0,\ldots,K-1} \sum_{i=1}^{b} \left[\frac{w_i}{\frac{1}{b} \sum_{l=1}^{b} w_l} \mathbb{E}\left[\left\| \nabla_i f(X^k) \right\|_{(i)\star} \right] \right] \leq \varepsilon, \tag{34}$$

after

$$K = \left\lceil \frac{2\delta^0 \sum_{i=1}^b \frac{\mathbb{P}\left(i\in\hat{S}\right)\mathbb{E}\left[L^0_{i,\hat{S}}\middle| i\in\hat{S}\right]}{\left(\mathbb{E}\left[L^1_{i,\hat{S}}\middle| i\in\hat{S}\right]\right)^2}}{\underline{\varepsilon}^2\left(\frac{1}{b}\sum_{l=1}^b \frac{\mathbb{P}\left(l\in\hat{S}\right)}{\mathbb{E}\left[L^1_{l,\hat{S}}\middle| l\in\hat{S}\right]}\right)^2} + \frac{2\delta^0}{\underline{\varepsilon}\left(\frac{1}{b}\sum_{l=1}^b \frac{\mathbb{P}\left(l\in\hat{S}\right)}{\mathbb{E}\left[L^1_{l,\hat{S}}\middle| l\in\hat{S}\right]}\right)}\right\rceil \tag{35}$$

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

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

and hence, substituting in (34) and (35), we have

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

after

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

iterations. Moreover, recall from (18) that the expected cost of a single iteration is

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

Hence, using the fact that

$$\mathbb{E}\left[L^\alpha_{i,\hat{S}}\middle| i\in\hat{S}\right] = \frac{\mathbb{E}\left[L^\alpha_{i,\hat{S}}\mathbb{I}\left(i\in\hat{S}\right)\right]}{\mathbb{P}\left(i\in\hat{S}\right)}$$

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

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

We will consider two regimes.

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

$$\min_{\mathcal{D}:\mathfrak{P}([b])\to[0,1],\sum_{S\subseteq[b]}\mathcal{D}(S)=1} \underbrace{\frac{\sum_{i=1}^b \frac{\mathbb{P}\left(i\in\hat{S}\right)^2\mathbb{E}\left[L^0_{i,\hat{S}}\mathbb{I}\left(i\in\hat{S}\right)\right]}{\left(\mathbb{E}\left[L^1_{i,\hat{S}}\mathbb{I}\left(i\in\hat{S}\right)\right]\right)^2}}{\min_{i\in[b]}\left(\frac{\mathbb{P}\left(i\in\hat{S}\right)^2}{\mathbb{E}\left[L^1_{i,\hat{S}}\mathbb{I}\left(i\in\hat{S}\right)\right]}\right)^2}\left(c_{\rm ov}+\sum_{i=1}^b c_iF_i+\sum_{i=1}^b c_i^\sharp Q_i\right)}_{\propto\,\mathbb{E}\left[\mathrm{cost}_{\varepsilon^2}(K)\right]}. \tag{36}$$

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

$$\min_{\mathcal{D}:\mathfrak{P}([b])\to[0,1],\sum_{S\subseteq[b]}\mathcal{D}(S)=1} \underbrace{\frac{1}{\min_{i\in[b]}\frac{\mathbb{P}\left(i\in\hat{S}\right)^2}{\mathbb{E}\left[L^1_{i,\hat{S}}\mathbb{I}\left(i\in\hat{S}\right)\right]}}\left(c_{\rm ov}+\sum_{i=1}^b c_iF_i+\sum_{i=1}^b c_i^\sharp Q_i\right)}_{\propto\,\mathbb{E}[\mathrm{cost}_{\varepsilon}(K)]}. \tag{37}$$

As in Section F.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.

F.2.1 Randomized Progressive Training

Sample $\hat{s}\in\{1,\dots,b\}$, where $p_i=\mathbb{P}\left(\hat{s}=i\right)$, 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

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

and

$$\mathbb{E}\left[L^\alpha_{i,\hat{S}}\mathbb{I}\left(i\in\hat{S}\right)\right]=\sum_{s=1}^i p_sL^\alpha_{i,\{s,\dots,b\}}$$

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

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

Hence, substituting into (36) and (37), the respective optimization problems reduce to minimizing

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

and

$$\mathbb{E}\left[\mathrm{cost}_{\varepsilon}(K)\right] \propto \frac{1}{\min_{i\in[b]}\frac{\mathbb{P}\left(i\in\hat{S}\right)^2}{\mathbb{E}\left[L^1_{i,\hat{S}}\mathbb{I}\left(i\in\hat{S}\right)\right]}}\left(\sum_{i=1}^b\left[c_{\rm ov}+\sum_{j\ge i}(c_j+c_j^\sharp)\right]p_i\right)$$

$$= \frac{1}{\min_{i\in[b]} \frac{\left(\sum_{s=1}^{i} p_s\right)^2}{\sum_{s=1}^{i} p_s L^1_{i,\{s,\dots,b\}}}} \left(\sum_{i=1}^{b} \left[c_{\mathrm{ov}} + \sum_{j\geq i} (c_j + c_j^\sharp) \right] p_i \right).$$

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

$$\begin{aligned} \min_{p,t} \quad & \frac{d_1 p_1 + \cdots + d_b p_b}{t} \\ \text{s.t.} \quad & p_1, \dots, p_b \geq 0 \\ & p_1 + \cdots + p_b = 1 \\ & t \geq 0 \\ & t \leq \frac{\left(\sum_{s=1}^{i} p_s\right)^2}{\sum_{s=1}^{i} p_s L^1_{i,\{s,\dots,b\}}}, \quad i \in [b]. \end{aligned} \tag{38}$$

This program can be written equivalently as

$$\begin{aligned} \min_{q} \quad & d_1 q_1 + \cdots + d_b q_b \\ \text{s.t.} \quad & q_1, \dots, q_b \geq 0 \\ & g_i(q) := \left(\sum_{s=1}^{i} q_s \right)^2 - \sum_{s=1}^{i} q_s L^1_{i,\{s,\dots,b\}} \geq 0, \quad i \in [b]. \end{aligned} \tag{39}$$

We now write the KKT conditions for (39). 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

$$\frac{\partial g_i(q)}{\partial q_k} = \begin{cases} 2\sum_{s=1}^{i} q_s - L^1_{i,\{k,\dots,b\}}, & k \leq i, \\ 0, & k > i. \end{cases} \tag{40}$$

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

$$0 = d_k - \sum_{i=k}^{b} \lambda_i \left(2\sum_{s=1}^{i} q_s - L^1_{i,\{k,\dots,b\}} \right) - \eta_k, \tag{41}$$

and complementarity and sign conditions are

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

Lemma F.2. Let $q^\star = (q_1^\star, \dots, q_p^\star)$ be a local minimizer of (39) and let $\mathrm{supp}(q^\star) := \{k : q_k^\star > 0\}$. Then, for every index $k \in \mathrm{supp}(q^\star)$, 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 $k \in \mathrm{supp}(q^\star)$. If $\lambda_i = 0$ for all $i \geq k$, then (41) at k reduces to

$$0 = d_k - \eta_k.$$

By complementary slackness $\eta_k q_k^\star = 0$, and since $q_k^\star > 0$, we must have $\eta_k = 0$, so $d_k = 0$ as well. But by definition $d_k = c_{\mathrm{ov}} + \sum_{j\geq k}(c_j + c_j^\sharp)$, 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. □

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

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

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

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

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

Lemma F.4. Let $q^\star$ be a KKT point of (39) such that $\operatorname{supp}(q^\star) = \{k\}$. Then

$$q_k^\star = \max_{i \geq k} L^1_{i,\{k,\dots,b\}}.$$

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

$$q_k^\star > \max_{i \geq k} L^1_{i,\{k,\dots,b\}}.$$

Then for every $i \geq k$ we have

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

Since by complementary slackness $\lambda_i g_i(q^\star) = 0$, we get $\lambda_i = 0$ for all $i \geq k$. Plugging this into the stationarity condition (41) yields

$$0 = d_k - \eta_k.$$

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

$$q_k^\star = \max_{i \geq k} L^1_{i,\{k,\dots,b\}}.$$

□

Theorem F.1. Unless

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

$(p_1, p_2, \dots, p_b) = (1, 0, \dots, 0)$ is *not* an optimal solution of (38).

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

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

and hence we must have $p_1 > 0$. Thus, the support of the solution $p^\star$ of (38) (which coincides with the support of the solution $q^\star$ of (39)) can only be a singleton if $\operatorname{supp}(p^\star) = \operatorname{supp}(q^\star) = \{1\}$. But then, by Lemma F.4, we must have

$$q_1^\star = \max_{i \in [b]} L^1_{i,[b]}, \quad q_2^\star = \dots = q_p^\star = 0,$$

which in turn implies that

$$0 \le g_i(q^\star) = \left(\sum_{s=1}^{i} q_s^\star\right)^2 - \sum_{s=1}^{i} q_s^\star L^1_{i,\{s,\dots,b\}} = q_1^\star \left(q_1^\star - L^1_{i,[b]}\right)$$

for all $i \in [b]$, so in particular $q_1^\star \left(q_1^\star - L^1_{1,[b]}\right) \ge 0$. Now, suppose $L^1_{1,[b]} \ne \max_{i\in[b]} L^1_{i,[b]}$. Then the inequality is strict and, by complementary slackness, $\lambda_1 = \eta_1 = 0$, so

$$d_1 = \sum_{i=2}^{b} \lambda_i \left(2q_1^\star - L^1_{i,\{1,\dots,b\}}\right) \le \sum_{i=2}^{b} \lambda_i \left(2q_1^\star - L^1_{i,\{2,\dots,b\}}\right) = d_2 - \eta_2 \le d_2,$$

which is a contradiction. Thus, similar to the result in Remark 8, we must have $L^1_{1,[b]} = \max_{i\in[b]} L^1_{i,[b]}$. □

F.2.2 τ-NICE SAMPLING

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

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

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

and

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

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

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

Hence, substituting into (36) and (37), the objective functions to minimize are

$$\begin{aligned}
&\mathbb{E}\left[\mathrm{cost}_{\varepsilon^2}(K)\right] \\
&\propto \frac{\sum_{i=1}^{b} \frac{\mathbb{P}\left(i\in\hat{S}\right)^2 \mathbb{E}\left[L^0_{i,\hat{S}} \mathbb{I}\left(i\in\hat{S}\right)\right]}{\left(\mathbb{E}\left[L^1_{i,\hat{S}} \mathbb{I}\left(i\in\hat{S}\right)\right]\right)^2}}{\min_{i\in[b]} \left(\frac{\mathbb{P}\left(i\in\hat{S}\right)^2}{\mathbb{E}\left[L^1_{i,\hat{S}} \mathbb{I}\left(i\in\hat{S}\right)\right]}\right)^2} \left(c_{\mathrm{ov}} + \sum_{j=1}^{b} c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}}\right) + \frac{\tau}{b} \sum_{j=1}^{b} c_j^\sharp\right) \\
&= \frac{\max_{i\in[b]} \left(\sum_{\substack{S\subseteq[b]\\|S|=\tau}} L^1_{i,S} \mathbb{I}\left(i \in S\right)\right)^2}{\frac{\tau}{b}\binom{b-1}{\tau-1}} \\
&\qquad \times \sum_{i=1}^{b} \frac{\sum_{\substack{S\subseteq[b]\\|S|=\tau}} L^0_{i,S} \mathbb{I}\left(i \in S\right)}{\left(\sum_{\substack{S\subseteq[b]\\|S|=\tau}} L^1_{i,S} \mathbb{I}\left(i \in S\right)\right)^2} \left(c_{\mathrm{ov}} + \sum_{j=1}^{b} c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}}\right) + \frac{\tau}{b} \sum_{j=1}^{b} c_j^\sharp\right)
\end{aligned}$$

and

$$\mathbb{E}\left[\mathrm{cost}_{\varepsilon}(K)\right] \propto \frac{1}{\min_{i\in[b]} \frac{\mathbb{P}\left(i\in\hat{S}\right)^2}{\mathbb{E}\left[L^1_{i,\hat{S}} \mathbb{I}\left(i\in\hat{S}\right)\right]}} \left(c_{\mathrm{ov}} + \sum_{j=1}^{b} c_j \left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}}\right) + \frac{\tau}{b} \sum_{j=1}^{b} c_j^\sharp\right)$$

$$= \frac{\max_{i\in[b]} \sum_{\substack{S\subseteq[b]\\|S|=\tau}} L^1_{i,S}\mathbb{I}\left(i\in S\right)}{\left(\frac{\tau}{b}\right)^2\binom{b}{\tau}}\left(c_{\text{ov}} + \sum_{j=1}^{b} c_j\left(1 - \frac{\binom{b-j}{\tau}}{\binom{b}{\tau}}\right) + \frac{\tau}{b}\sum_{j=1}^{b} c_j^\sharp\right).$$

As in Section F.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 $L^0_{i,S}$ depends only on i and $|S|$ and define $L^0_{i,\tau} := L^0_{i,S}$ for $|S| = \tau$. Then

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

and

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

We have already established in Section F.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 $A(\tau) := \max_{i\in[b]} L^0_{i,\tau}$. Since $L^1_{i,\tau}$, like $L^0_{i,\tau}$, is non-decreasing in τ, the same reasoning as in Section F.1.2 applies to $\mathbb{E}\left[\text{cost}_\varepsilon(K)\right]$. On the other hand, the dependence of $A_{\varepsilon^2}(\tau)$ on τ can be arbitrary depending on the scaling between $L^0_{i,\tau}$ and $L^1_{i,\tau}$.

Consequently, both objectives $\mathbb{E}\left[\text{cost}_\varepsilon(K)\right]$ and $\mathbb{E}\left[\text{cost}_{\varepsilon^2}(K)\right]$ 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.

F.2.3 τ-SUBMODEL SAMPLING

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

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

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

and

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

We have also already shown that

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

Partitioned τ-submodel sampling. Mimicking Section F.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

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

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

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

and

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

for $\alpha \in \{0, 1\}$. Plugging this choice into (36) and (37) gives

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

and

$$\begin{aligned}
\mathbb{E}\left[\mathrm{cost}_{\varepsilon}(K)\right] \quad &\propto \quad \frac{1}{\min_{i\in[b]} \frac{\mathbb{P}\left(i\in\hat{S}\right)^2}{\mathbb{E}\left[L^1_{i,\hat{S}}\mathbb{I}\left(i\in\hat{S}\right)\right]}} \\
&\qquad \times \left(c_{\mathrm{ov}} + \sum_{j=1}^{b} c_j \left(\sum_{i=1}^{\min\{j,b-\tau+1\}} p_i\right) + \sum_{j=1}^{b} c_j^\sharp \left(\sum_{i=\max\{1,j-\tau+1\}}^{\min\{j,b-\tau+1\}} p_i\right)\right)
\end{aligned}$$

$$= \frac{1}{\min_{i\in[b]} \frac{p_{s_{\lceil i/\tau \rceil}}}{L^1_{i,B_{\lceil i/\tau \rceil}}}} \left(c_{\mathrm{ov}} + \sum_{j=1}^{b} c_j \left(\sum_{k=1}^{\lceil j/\tau \rceil} p_{s_k} \right) + \sum_{j=1}^{b} c_j^\sharp p_{s_{\lceil j/\tau \rceil}} \right).$$

The expression for $\mathbb{E}\left[\mathrm{cost}_\varepsilon(K)\right]$ is entirely analogous to (26) derived in the smooth case, with $L^0_{i,B_{\lceil i/\tau \rceil}}$ replaced by $L^1_{i,B_{\lceil i/\tau \rceil}}$. Consequently, for a fixed τ, the optimal block probabilities are

$$p^\star_{s_k} = \frac{\max_{i\in B_k} L^1_{i,B_k}}{\sum_{l=1}^m \max_{i\in B_l} L^1_{i,B_l}},$$

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

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

F.2.4 Arbitrary Submodel Sampling

Let $\{B_1, \ldots, B_m\}$ 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

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

and

$$\mathbb{E}\left[L^\alpha_{i,\hat{S}} \mathbb{I}\left(i \in \hat{S}\right)\right] = p_{k(i)} L^\alpha_{i,B_{k(i)}}$$

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

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

the total expected costs become

$$\mathbb{E}\left[\mathrm{cost}_{\varepsilon^2}(K)\right] \propto \frac{\sum_{i=1}^b \frac{\mathbb{P}\left(i\in\hat{S}\right)^2 \mathbb{E}\left[L^0_{i,\hat{S}} \mathbb{I}\left(i\in\hat{S}\right)\right]}{\left(\mathbb{E}\left[L^1_{i,\hat{S}} \mathbb{I}\left(i\in\hat{S}\right)\right]\right)^2}}{\min_{i\in[b]} \left(\frac{\mathbb{P}\left(i\in\hat{S}\right)^2}{\mathbb{E}\left[L^1_{i,\hat{S}} \mathbb{I}\left(i\in\hat{S}\right)\right]} \right)^2} \left(c_{\mathrm{ov}} + \sum_{j=1}^{b} c_j \left(\sum_{k:\underline{b}_k \le j} p_k \right) + \sum_{j=1}^{b} c_j^\sharp p_{k(j)} \right)$$

$$= \frac{\sum_{i=1}^b \frac{p_{k(i)} L^0_{i,B_{k(i)}}}{\left(L^1_{i,B_{k(i)}}\right)^2}}{\min_{i\in[b]} \left(\frac{p_{k(i)}}{L^1_{i,B_{k(i)}}} \right)^2} \left(c_{\mathrm{ov}} + \sum_{j=1}^{b} c_j \left(\sum_{k:\underline{b}_k \le j} p_k \right) + \sum_{j=1}^{b} c_j^\sharp p_{k(j)} \right)$$

and

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

$$= \frac{1}{\min_{i\in[b]} \frac{p_{k(i)}}{L^1_{i,B_{k(i)}}}} \left(c_{\mathrm{ov}} + \sum_{j=1}^{b} c_j \left(\sum_{k:\underline{b}_k \le j} p_k \right) + \sum_{j=1}^{b} c_j^\sharp p_{k(j)} \right).$$

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

$$p^\star_k = \frac{1/\delta^{\min}_k}{\sum_{l=1}^m 1/\delta^{\min}_l} = \frac{\max_{i\in B_k} L^0_{i,B_k}}{\sum_{l=1}^m \max_{i\in B_l} L^0_{i,B_l}},$$

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

G Convergence Results – Stochastic Gradient Case

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

Lemma G.1 (Descent Lemma)**.** Let Assumption 4.4 hold and consider the update rule $X_i^{k+1} = \mathrm{LMO}_{\mathcal{B}(X_i^k, t_i^k)}(M_i^k)$, $i = 1, \dots, b$, 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

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

Proof. By Lemma I.3

$$\begin{aligned}
& f(X^{k+1}) \\
\leq\ & f(X^k) + \left\langle \nabla f(X^k), X^{k+1} - X^k \right\rangle + \sum_{i \in S^k} \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \left\|\nabla_i f(X^k)\right\|_{(i)\star}}{2} \left\|X_i^k - X_i^{k+1}\right\|_{(i)}^2 \\
\overset{(51)}{=}\ & f(X^k) + \sum_{i \in S^k} \left(\left\langle \nabla_i f(X^k) - M_i^k, X_i^{k+1} - X_i^k \right\rangle_{(i)} + \left\langle M_i^k, X_i^{k+1} - X_i^k \right\rangle_{(i)} \right) \\
& + \sum_{i \in S^k} \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \left\|\nabla_i f(X^k)\right\|_{(i)\star}}{2} (t_i^k)^2 \\
\overset{(52)}{=}\ & f(X^k) + \sum_{i \in S^k} \left(\left\langle \nabla_i f(X^k) - M_i^k, X_i^{k+1} - X_i^k \right\rangle_{(i)} - t_i^k \left\|M_i^k\right\|_{(i)\star} \right) \\
& + \sum_{i \in S^k} \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \left\|\nabla_i f(X^k)\right\|_{(i)\star}}{2} (t_i^k)^2 \\
\leq\ & f(X^k) + \sum_{i \in S^k} \left(t_i^k \left\|\nabla_i f(X^k) - M_i^k\right\|_{(i)\star} - t_i^k \left\|M_i^k\right\|_{(i)\star} \right. \\
& \left. + \frac{L_{i,S^k}^0 + L_{i,S^k}^1 \left\|\nabla_i f(X^k)\right\|_{(i)\star}}{2} (t_i^k)^2 \right),
\end{aligned}$$

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

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

as required. □

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

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

Proof. The proof is inspired by Cutkosky & Mehta (2020, Theorem 1). First, using the momentum update rule, we have

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

where $\beta_i^k := \mathbb{I}\left(i \in S^k\right)\beta_i$. 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

$$U_{1,i}^{k+1} = \left(1 - \beta_i^k\right) U_{1,i}^k + \left(1 - \beta_i^k\right) U_{2,i}^k + \beta_i^k U_{3,i}^{k+1}. \tag{42}$$

Unrolling the recursion in (42), 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,

$$\begin{aligned}
\mathbb{E}\left[\left\|U_{1,i}^{k+1}\right\|_2\right] &\leq \mathbb{E}\left[\left\|\left(\prod_{m=0}^{k}(1 - \beta_i^m)\right) U_{1,i}^0\right\|_2\right] + \mathbb{E}\left[\left\|\sum_{l=0}^{k}\left(\prod_{m=l}^{k}(1 - \beta_i^m)\right) U_{2,i}^l\right\|_2\right] \\
&\quad + \mathbb{E}\left[\left\|\sum_{l=0}^{k}\left(\beta_i^l \prod_{m=l+1}^{k}(1 - \beta_i^m)\right) U_{3,i}^{l+1}\right\|_2\right].
\end{aligned} \tag{43}$$

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

$$\begin{aligned}
\mathbb{E}\left[\left\|\left(\prod_{m=0}^{k}(1 - \beta_i^m)\right) U_{1,i}^0\right\|_2\right] &\leq \mathbb{E}\left[\left(\prod_{m=0}^{k}(1 - \beta_i^m)\right)\left\|U_{1,i}^0\right\|_2\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\prod_{m=0}^{k}(1 - \beta_i^m)\right)\middle| X^k, M_i^k\right]\left\|U_{1,i}^0\right\|_2\right] \\
&= \prod_{m=0}^{k}\mathbb{E}\left[1 - \beta_i^m\right]\mathbb{E}\left[\left\|U_{1,i}^0\right\|_2\right].
\end{aligned} \tag{44}$$

Next, by Assumption 4.4

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

$$
\begin{aligned}
&\leq \mathbb{E}\left[\sum_{l=0}^{k}\left(\prod_{m=l}^{k}(1-\beta_i^m)\right)\left\|U_{2,i}^l\right\|_2\right] \\
&\leq \frac{1}{\underline{\rho}_i}\sum_{l=0}^{k}\mathbb{E}\left[\left(\prod_{m=l}^{k}(1-\beta_i^m)\right)\left\|\nabla_i f(X^l)-\nabla_i f(X^{l+1})\right\|_{(i)\star}\right] \\
&\overset{(4.3)}{\leq} \frac{1}{\underline{\rho}_i}\sum_{l=0}^{k}\mathbb{E}\left[\left(\prod_{m=l}^{k}(1-\beta_i^m)\right)\left(L_{i,S^l}^0+L_{i,S^l}^1\left\|\nabla_i f(X^l)\right\|_{(i)\star}\right)\left\|X_i^l-X_i^{l+1}\right\|_{(i)}\right].
\end{aligned}
$$

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,

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

where

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

Thus

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\sum_{l=0}^{k}\left(\prod_{m=l}^{k}(1-\beta_i^m)\right)U_{2,i}^l\right\|_2\right] \qquad (45)\\
&\leq \frac{t_i}{\underline{\rho}_i}\sum_{l=0}^{k}\left(\prod_{m=l+1}^{k}\mathbb{E}\left[1-\beta_i^m\right]\right)\left(\mathbb{E}\left[(1-\beta_i^l)L_{i,S^l}^0\right]+\mathbb{E}\left[(1-\beta_i^l)L_{i,S^l}^1\right]\mathbb{E}\left[\left\|\nabla_i f(X^l)\right\|_{(i)\star}\right]\right).
\end{aligned}
$$

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

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

$$
\begin{aligned}
&= \sigma_i \sqrt{\sum_{l=0}^{k} \left(\beta_i^2 \mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right] \mathbb{E}\left[\left(1 - \mathbb{I}\left(i \in \hat{S}\right)\beta_i\right)^2\right]^{k-l} \right)} \\
&\leq \sigma_i \sqrt{\frac{\beta_i^2 \mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right]}{1 - \mathbb{E}\left[\left(1 - \mathbb{I}\left(i \in \hat{S}\right)\beta_i\right)^2\right]}} \\
&= \sigma_i \beta_i \sqrt{\frac{\mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right]}{\mathbb{E}\left[(2 - \beta_i)\,\mathbb{I}\left(i \in \hat{S}\right)\beta_i\right]}} = \frac{\sigma_i \beta_i}{\sqrt{(2-\beta_i)\,\beta_i}} \leq \sigma_i \sqrt{\beta_i}. \qquad (46)
\end{aligned}
$$

Applying (44), (45) and (46) in (43) and noting that $\mathbb{E}\left[\beta_i^k\right] = \mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right]\beta_i$ yields

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

□

Theorem G.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

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

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

Then

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

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

Remark 9. *For Theorem G.1 to be meaningful, the sampling must be such that* $\mathbb{E}\left[\mathbb{I}\left(i\in\hat{S}\right)\right]\eta_i>0$ *for every* $i\in[b]$*. Thus, every layer has to be sampled with positive probability.*

Remark 10. *Note that under Assumption 4.4, without loss of generality we can set* $L^0_{i,S}=L^1_{i,S}=0$ *whenever* $i\notin S$*. Hence, in the case of* RPT *(see Section 4), we have*

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

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

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

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

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

Hence, taking full expectation

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

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

$$\begin{aligned}
\delta^{k+1} \leq\ & \delta^k - \sum_{i=1}^{b} t_i \mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right] \mathbb{E}\left[\left\|\nabla_i f(X^k)\right\|_{(i)\star}\right] + \sum_{i=1}^{b} 2t_i\bar{\rho}_i \mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right] P_i^k \\
& + \frac{1}{2}\sum_{i=1}^{b} t_i^2 \mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right) L_{i,\hat{S}}^0\right] + \frac{1}{2}\sum_{i=1}^{b} t_i^2 \mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right) L_{i,\hat{S}}^1\right] \mathbb{E}\left[\left\|\nabla_i f(X^k)\right\|_{(i)\star}\right], \qquad (47) \\
P_i^k \leq\ & \left(1 - \mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right]\beta_i\right)^k P_i^0 + \frac{t_i \mathbb{E}\left[\left(1 - \mathbb{I}\left(i \in \hat{S}\right)\beta_i\right) L_{i,\hat{S}}^0\right]}{\underline{\rho}_i \beta_i \mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right]} \\
& + \frac{t_i}{\underline{\rho}_i} \mathbb{E}\left[\left(1 - \mathbb{I}\left(i \in \hat{S}\right)\beta_i\right) L_{i,\hat{S}}^1\right] \sum_{l=0}^{k-1} \left(1 - \mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right]\beta_i\right)^{k-1-l} \mathbb{E}\left[\left\|\nabla_i f(X^l)\right\|_{(i)\star}\right] \\
& + \sigma_i\sqrt{\beta_i}. \qquad (48)
\end{aligned}$$

Applying (48) in (47),

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

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

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

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

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

and

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

Thus

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

$$
\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\underline{\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\underline{\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\underline{\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}\\
=\ &\left(1+c_1\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]\\
&+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

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

Now, let us introduce a weighting sequence $w^k := w^{k-1}\left(1 + c_1 + \frac{c_2}{\min_{i\in[b]}\left(\mathbb{E}\left[\mathbb{I}\left(i\in\hat{S}\right)\right]\beta_i\right)}\right)^{-1}$, 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

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

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

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

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

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

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

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

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

we have

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

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

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

meaning that

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

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

$$
\begin{aligned}
&\min_{k=0,\ldots,K} \sum_{i=1}^{b} \frac{\mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right] \eta_i}{\frac{1}{b}\sum_{l=1}^{b} \mathbb{E}\left[\mathbb{I}\left(l \in \hat{S}\right)\right] \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} \mathbb{E}\left[\mathbb{I}\left(l \in \hat{S}\right)\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}\left(l \in \hat{S}\right)\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}\left(l \in \hat{S}\right)\right] \eta_l} \frac{2\bar{\rho}_i}{\beta_i \underline{\rho}_i} \\
&\qquad + \times \left(\mathbb{E}\left[\left(1 - \mathbb{I}\left(i \in \hat{S}\right) \beta_i\right) L_{i,\hat{S}}^0\right] + \mathbb{E}\left[\left(1 - \mathbb{I}\left(i \in \hat{S}\right) \beta_i\right) L_{i,\hat{S}}^1\right] \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}\left[\mathbb{I}\left(l \in \hat{S}\right)\right] \eta_l} \left(\mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right) L_{i,\hat{S}}^0\right] + \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] \right) \\
&\quad + \sum_{i=1}^{b} \frac{2\mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right] \eta_i}{\frac{1}{b}\sum_{l=1}^{b} \mathbb{E}\left[\mathbb{I}\left(l \in \hat{S}\right)\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}\left(l \in \hat{S}\right)\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}\left(l \in \hat{S}\right)\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}\left(l \in \hat{S}\right)\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] \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}\left[\mathbb{I}\left(l \in \hat{S}\right)\right] \eta_l} \left(\mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right) L_{i,\hat{S}}^0\right] + \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] \right) \\
&\quad + \sum_{i=1}^{b} \frac{2\mathbb{E}\left[\mathbb{I}\left(i \in \hat{S}\right)\right] \eta_i}{(K+1)^{1/4} \frac{1}{b}\sum_{l=1}^{b} \mathbb{E}\left[\mathbb{I}\left(l \in \hat{S}\right)\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

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

□

H EXPERIMENTS

H.1 EXPERIMENTAL DESIGN

Every comparison pairs a Muon run with a Drop-Muon run that uses the same model, data, initialization, batch size, learning-rate schedule, and hardware. In the language model experiments, the learning rate is tuned independently for each method; all other hyperparameters are tuned for Muon and shared by both methods. Both runs process the same data for the same number of optimizer steps. We record quality metrics together with cumulative training time, and compute time-to-target ratios within each matched pair before averaging across pairs.

H.2 DROP-MUON IMPLEMENTATION

The network is partitioned into ordered trainable groups from input to output. At optimizer step k, a cutoff s_k freezes groups $1, \dots, s_k - 1$ and activates the suffix $s_k, \dots, b$; the cutoff is held fixed across all gradient-accumulation micro-batches of that step. Frozen parameters produce no gradients and receive no optimizer or momentum update. The frozen prefix is evaluated without building its backward graph, and the graph is detached at the prefix–suffix boundary; the stochastic forward pass itself is unchanged. In distributed training, gradients are synchronized only for parameters that received a gradient, so frozen groups incur no gradient communication. Cutoff sampling uses a dedicated random-number generator, so it does not perturb data ordering or augmentation randomness.

For GPT models, the $b = n_{\text{layer}} + 2$ groups are the token embedding, each Transformer block, and the output head. The embedding and head are deliberately untied: tying them would make one parameter simultaneously belong to the frozen prefix and active suffix. For convolutional models, we use the natural block structure of the network, grouping convolutional blocks and the classifier head.

H.3 LAYER-SAMPLING SCHEDULE

We use the pure-linear family

$$w_i(s) = 1 + s\left(\frac{2(i-1)}{b-1} - 1\right), \qquad p_i(s) = \frac{w_i(s)}{\sum_{j=1}^{b} w_j(s)}, \qquad s \in [-1, 1]. \tag{49}$$

All principal results use $s = -1$, which gives $p_i = 2(b-i)/[b(b-1)]$. In particular, $p_b = 0$, so the output group is updated at every step. No extra mixture of full-network updates is used.

Late wake-up. In the language model experiments, Drop-Muon performs full-network Muon updates ($s_k = 1$) during the first fraction ρ of training, after which the progressive sampling rule is activated. Early in training, the early layers strongly influence the learned representations, so we first let the full model adapt before applying partial updates. We use $\rho = 0.35$ for NanoGPT-162M and $\rho = 0.30$ for the 24-layer models, selected from $\rho \in \{0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.60\}$. The wake-up fraction is therefore a tuned hyperparameter of the method.

Late wake-up ablation. On NanoGPT-162M (one seed, all other settings fixed), we varied $\rho \in \{0.10, 0.20, 0.30, 0.40, 0.50\}$. The wake-up fraction trades computation for per-step progress: moving from $\rho = 0.10$ to $\rho = 0.50$ lowers the runtime reduction relative to Muon from 13.4% to 8.0%, while the final loss gap to Muon at a fixed number of steps decreases monotonically from 0.039 to 0.014.

H.4 METRICS

Let $T_{\mathsf{alg}}(\tau)$ be the first cumulative training time at which method alg reaches target loss or accuracy τ. We report

$$\text{Speedup}(\tau) = \frac{T_{\mathsf{Muon}}(\tau)}{T_{\mathsf{Drop\text{-}Muon}}(\tau)}. \tag{50}$$

A value larger than one favors Drop-Muon. When results are averaged across runs, we first compute the speedup within each matched pair and then average the ratios. Cumulative training time is synchronized training-loop time and excludes validation.

H.5 Vision experiments

The small controlled experiments use three-layer CNNs on MNIST and Fashion-MNIST with a constant learning rate. We search batch sizes $\{64, 512, 8192, 16384, 32768\}$ and learning rates $\{0.1, 0.01, 0.001\}$, use five Newton–Schulz iterations, and set the momentum parameter to 0.5. The configurations displayed in Figure 1 use batch size 8192 for MNIST and 32768 for Fashion-MNIST, with learning rate 0.1; the MNIST bars average seven paired runs. The bars show paired time-to-target ratios rather than final-accuracy improvements.

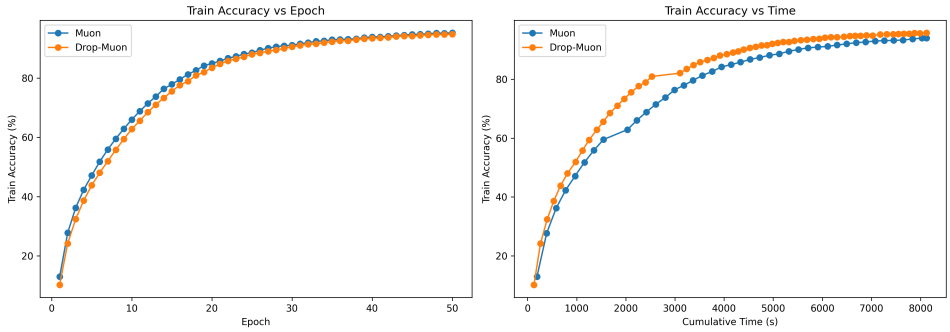


Figure 3: Training accuracy of Drop-Muon and Muon on ResNet50/Tiny ImageNet against epochs (left) and cumulative training time (right). Learning rate 0.02, batch size 256, one V100 GPU, single seed. In the right panel, both curves end when Drop-Muon completes its 50 epochs.

For the larger vision experiment, we train a randomly initialized ResNet50 (23.9M parameters) on Tiny ImageNet for 50 epochs with batch size 256 and learning rate 0.02 for both methods, on one V100 GPU. Figure 3 shows training accuracy against epochs and against cumulative training time. Per epoch, Drop-Muon is slightly behind Muon early in training and matches it by the end, while its epochs are about 20% shorter; it therefore reaches every training-accuracy level earlier in wall-clock time.

H.6 Language modeling

We train decoder-only GPT models on pre-tokenized FineWeb10B using sequence length 1024 and vocabulary size $50{,}304$. Each block uses pre-norm residual connections with parameter-free RMSNorm, rotary position embeddings with RMS-normalized queries and keys, and an MLP with expansion factor four and squared-ReLU activation. Bias parameters are omitted, a final RMSNorm precedes the output head, and the token embedding and language modeling head are independent. Table 1 gives the model dimensions.

Table 1: GPT model configurations.

Model	Parameters	Layers	Heads	Width	MLP width
NanoGPT-162M	162,201,600	12	6	768	3072
GPT-24L-247M	247,136,256	24	6	768	3072
GPT-24L-405M	405,012,480	24	16	1024	4096

The learning rate is tuned independently for each method over $\{2.6, 3.0, 3.24, 3.42, 3.6, 3.78, 4.0, 4.2, 4.6, 5.6, 7.6\} \times 10^{-4}$, and both methods select 3.6×10^{-4}. All other hyperparameters are tuned for Muon and shared by both methods; in particular, the global batch size, selected from $\{64, 128, 256, 512, 1024\}$, is 256 sequences ($262{,}144$ tokens per optimizer step). The learning rate is constant for the first 40% of training and then decays linearly to 10% of its initial value. The momentum buffer is $m_k = 0.9\, m_{k-1} + 0.1\, g_k$. Transformer matrices use

the spectral LMO (five Newton–Schulz iterations, update rescaled by $\sqrt{d_{\text{out}}/d_{\text{in}}}$) with scale 50; the token embedding uses a column-wise RMS-normalized update with scale 0.01, selected from $\{0.01, 0.5, 1, 5, 10, 20, 40, 50, 60, 80, 100, 120, 140, 200, 500, 5000\}$; and the output head uses a sign update normalized by $1/d_{\text{in}}$ with scale 3000. We use gradient clipping with threshold 5×10^4 as a numerical safeguard only, and no weight decay. Transformer-block weights are cast to BF16 inside the forward pass, while master weights and optimizer states are kept in FP32; PyTorch automatic mixed precision is not used.

The runs in Figure 2 use four V100 or A100 GPUs (the same GPU type within each pair) with a per-device micro-batch of 4 sequences and 16 gradient-accumulation steps. NanoGPT-162M is trained on 20% of FineWeb10B, and the two 24-layer models on 10%. The training-loss curves are downsampled by a factor of 10 (NanoGPT-162M) or 5 (24-layer models) and smoothed with an exponential moving average with parameter 0.02 (NanoGPT-162M and GPT-24L-247M) or 0.01 (GPT-24L-405M) for display; the test-loss curves in Figure 2 are shown as recorded.

H.6.1 Estimating the layer-wise smoothness constants

Assumptions 4.3 and 4.4 posit constants $L^0_{i,S}, L^1_{i,S}$ for each layer $i \in [b]$ and each active subset $S \in \text{supp}(\mathcal{D})$. We estimate these along the training trajectory following a procedure analogous to that of Riabinin et al. (2025), adapted to the subset-dependent setting considered here: whereas they fit a single pair (L^0_i, L^1_i) per layer, we fit one pair per (i, S). We refer the reader to Riabinin et al. (2025) for further details on the procedure.

We train `NanoGPT-162M` with full-network Muon using the configuration above. We probe the model at the current iterate X^k. For each cutoff $j \in [b]$ we form the hypothetical iterate $X^{k+1}(S_j)$ obtained by applying the update to the suffix $S_j = \{j, \dots, b\}$ only, and for every active layer $i \in S_j$ we record

$$\hat{L}_{i,S_j}[k] := \frac{\left\|\nabla_i f(X^{k+1}(S_j)) - \nabla_i f(X^k)\right\|_{(i)\star}}{\left\|X_i^{k+1}(S_j) - X_i^k\right\|_{(i)}}, \qquad G_i[k] := \left\|\nabla_i f(X^k)\right\|_{(i)\star}.$$

Both gradients are evaluated on the same minibatch.

Fitting. For each pair (i, S_j) we fit $\hat{L} \approx L^0 + L^1 G$ by minimizing the squared error with an additional hinge penalty on underestimation,

$$\sum_k \left(\hat{L}_{i,S_j}[k] - L^0 - L^1 G_i[k]\right)^2 + \lambda \sum_k \left[\hat{L}_{i,S_j}[k] - L^0 - L^1 G_i[k]\right]_+^2,$$

over $L^0, L^1 \geq 0$, with $\lambda = 1$. The parameter λ trades goodness of fit against a bias towards not underestimating the smoothness.

Results. Table 2 reports the fitted L^1_{i,S_j}. First, the fit returns $L^0_{i,S} = 0$ in 102 of the 105 pairs, so the model effectively reduces to a slope; we therefore omit L^0 from the table. This mirrors the finding of Riabinin et al. (2025) that $L^0_i \approx 0$ across the architecture. Second, every row of the table is nonincreasing from left to right, with no exceptions among the 91 adjacent pairs: as one may expect, the effective smoothness of a layer grows with the set of layers updated alongside it. Third, and most relevant to the results of Section 4, the first column gives

$$L^1_{1,[b]} = 35.2 \;<\; \max_{i \in [b]} L^1_{i,[b]} = 52.3,$$

supporting the fact that full-network updates are not optimal here.

H.7 Component profiling

We profile NanoGPT-162M on four V100 GPUs with global batch size 256 and a per-device micro-batch of 16, with dropping active from the first step (no wake-up), so the profile corresponds to the post-wake-up regime. Each primary profiling run uses 150 warm-up steps, 100 uninstrumented steps for end-to-end ("clean") latency, and 400 instrumented steps for the component breakdown, with synchronized GPU timing. We run three paired seeds, a Drop-Muon control with the cutoff

Table 2: Fitted smoothness constants L^1_{i,S_j} for `NanoGPT-162M` along the training trajectory of full-network Muon, where $S_j = \{j, \ldots, b\}$ and $b = 14$. Layer 1 is the token embedding, layers 2–13 are the transformer blocks, and layer 14 is the language modeling head. Entries above the diagonal are empty because layer i is inactive whenever $j > i$. The corresponding $L^0_{i,S_j} \approx 0$ in all but three pairs and is omitted.

	Active subset													
Layer i	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}
1	35.25													
2	49.05	35.70												
3	52.34	38.46	35.50											
4	51.21	38.20	35.72	33.49										
5	48.74	36.61	34.38	32.53	30.45									
6	44.94	33.72	31.71	30.19	28.43	26.20								
7	41.98	31.70	29.90	28.63	27.13	25.18	23.00							
8	40.27	30.90	29.33	28.26	26.97	25.25	23.31	20.85						
9	36.52	28.20	26.84	25.94	24.84	23.36	21.68	19.56	16.80					
10	33.19	25.91	24.77	24.01	23.09	21.83	20.38	18.53	16.09	13.56				
11	30.95	24.20	23.16	22.49	21.67	20.54	19.23	17.56	15.37	13.07	10.77			
12	28.64	22.36	21.43	20.82	20.08	19.07	17.90	16.42	14.52	12.51	10.49	8.38		
13	27.72	22.10	21.24	20.68	20.02	19.12	18.08	16.77	15.11	13.34	11.57	9.65	7.80	
14	0.41	0.30	0.29	0.28	0.27	0.25	0.24	0.22	0.20	0.18	0.16	0.14	0.13	0.11

forced to the full network, and forced cutoffs sweeping the size of the active suffix. Stochastic Drop-Muon reduces clean step time from 6482 ms to 5646 ms (12.9%); the main reductions are backward (716 ms), communication (83 ms), and Newton–Schulz (54 ms), while forward time is unchanged. The forced-cutoff sweep is nearly linear in the active backward fraction and predicts the stochastic latency within approximately 1.2%. The cutoff-zero control differs from Muon by about 1.3%, indicating small controller overhead.

H.8 Interpretation and limitations

The experiments establish that Drop-Muon removes the intended computation, and that this yields faster time-to-accuracy in the CNN and ResNet50 experiments. The ResNet50 and language model comparisons each use one run per method. We therefore avoid extrapolating a single speedup across tasks.

I USEFUL FACTS

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

$$\left\|\mathrm{LMO}_{\mathcal{B}(0,t)}(G)\right\| = t \tag{51}$$

$$\left\langle G, \mathrm{LMO}_{\mathcal{B}(X,t)}(G)\right\rangle = -t\left\|G\right\|_\star \tag{52}$$

$$\left\langle X, X^\sharp\right\rangle = \left\|X^\sharp\right\|^2, \tag{53}$$

$$\left\|X\right\|_\star = \left\|X^\sharp\right\|, \tag{54}$$

$$\left\langle H, X\right\rangle = \left\|X\right\|_\star, \quad \left\|H\right\| = 1, \qquad \forall X \neq 0. \tag{55}$$

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

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

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

Lemma I.2 (Riabinin et al. (2025), Lemma 3)**.** Suppose that $x_1, \ldots, x_p, y_1, \ldots, y_p \in \mathbb{R}$, $\max_{i\in[b]}|x_i| > 0$ and $z_1, \ldots, z_p > 0$. 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 I.3. Let Assumption 4.4 hold and let $S \subseteq [b]$. Then, for any vectors $X = [X_1, \ldots, X_b] \in \mathcal{X}$ and $\Gamma = [\Gamma_1, \ldots, \Gamma_b] \in \mathcal{X}$ such that $\Gamma_i = 0$ for all $i \notin S$,

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

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

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

Therefore, using the Cauchy-Schwarz inequality

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

□

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

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

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

Proof. Let $Y = [Y_1, \ldots, Y_b] \in \mathcal{X}$, where $Y_i = X_i - \frac{\|\nabla_i f(X)\|_{(i)\star}}{L^0_{i,S} + L^1_{i,S}\|\nabla_i f(X)\|_{(i)\star}} H_i$ 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 I.3

$$\begin{aligned}
f(Y) &\leq f(X) + \langle \nabla f(X), Y - X\rangle + \sum_{i \in S} \frac{L^0_{i,S} + L^1_{i,S}\|\nabla_i f(X)\|_{(i)\star}}{2}\|X_i - Y_i\|^2_{(i)} \\
&= f(X) + \sum_{i \in S}\langle \nabla_i f(X), Y_i - X_i\rangle_{(i)} + \sum_{i \in S} \frac{L^0_{i,S} + L^1_{i,S}\|\nabla_i f(X)\|_{(i)\star}}{2}\|X_i - Y_i\|^2_{(i)} \\
&= f(X) - \sum_{i \in S} \frac{\|\nabla_i f(X)\|_{(i)\star}}{L^0_{i,S} + L^1_{i,S}\|\nabla_i f(X)\|_{(i)\star}}\langle \nabla_i f(X), H_i\rangle_{(i)} \\
&\quad + \sum_{i \in S}\left(\frac{L^0_{i,S} + L^1_{i,S}\|\nabla_i f(X)\|_{(i)\star}}{2}\frac{\|\nabla_i f(X)\|^2_{(i)\star}}{\left(L^0_{i,S} + L^1_{i,S}\|\nabla_i f(X)\|_{(i)\star}\right)^2}\|H_i\|^2_{(i)}\right) \\
&\overset{(55)}{=} f(X) + \sum_{i \in S}\left(-\frac{\|\nabla_i f(X)\|^2_{(i)\star}}{L^0_{i,S} + L^1_{i,S}\|\nabla_i f(X)\|_{(i)\star}} + \frac{\|\nabla_i f(X)\|^2_{(i)\star}}{2\left(L^0_{i,S} + L^1_{i,S}\|\nabla_i f(X)\|_{(i)\star}\right)}\right) \\
&= f(X) - \sum_{i \in S} \frac{\|\nabla_i f(X)\|^2_{(i)\star}}{2\left(L^0_{i,S} + L^1_{i,S}\|\nabla_i f(X)\|_{(i)\star}\right)},
\end{aligned}$$

and hence

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

□

Lemma I.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

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

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

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

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

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

Therefore,

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

□