Rescaled Asynchronous SGD: Optimal Distributed Optimization under Data and System Heterogeneity

Ammar Mahran [1] **Artavazd Maranjyan** [1] **Peter Richtárik** [1]

Abstract

Asynchronous stochastic gradient descent (ASGD) is a standard way to exploit heterogeneous compute resources in distributed learning: instead of forcing fast workers to wait for slow ones, the server updates the model whenever a gradient arrives. Vanilla ASGD applies each arriving gradient with the same weight. When local data distributions are heterogeneous, this becomes problematic: faster workers contribute more updates, and we show theoretically that the method is biased toward a frequency-weighted average of the local objectives rather than the desired global objective. Existing remedies typically move away from the simple ASGD template by introducing gathering phases, buffering, or extra memory. We show that this is unnecessary. Keeping the standard ASGD mechanism, we recover the correct objective by rescaling worker-specific stepsizes in proportion to their computation times, so that each worker contributes the same aggregate learning rate over a cycle. In the non-convex setting, under smoothness and bounded heterogeneity assumptions, we prove that the resulting method, Rescaled ASGD, converges to stationary points of the correct global objective in the fixed-computation model. Its time complexity matches the known lower bound in the leading term, while the effects of staleness and data heterogeneity appear only in lower-order terms. Experiments confirm that the method converges to the correct objective and is competitive with state-of-the-art baselines.

[1]King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. Correspondence to: Ammar Mahran <majiedammar.mahran@kaust.edu.sa>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

1. Introduction

We consider the non-convex distributed optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left\{ F(\mathbf{x}) \coloneqq \frac{1}{n}\sum_{i=1}^n F_i(\mathbf{x}) \right\}, \tag{1}$$

where $F_i : \mathbb{R}^d \to \mathbb{R}, \mathbf{x} \mapsto \mathbb{E}_{\xi\sim\mathcal{D}_i}\left[f_i(\mathbf{x};\xi)\right]$ denotes the local objective function of worker $i = 1, \ldots, n$, which is the expected value of the sample loss $f_i(\mathbf{x};\xi)$ over data points ξ drawn from their local distribution $\mathcal{D}_i$. In parallel stochastic gradient descent methods, workers collaboratively solve this problem by computing stochastic gradients of their local objectives, which are then processed by a central server to find an ε-stationary point, that is, a random vector $\bar{\mathbf{x}} \in \mathbb{R}^d$ such that $\mathbb{E}\left\|\nabla F(\bar{\mathbf{x}})\right\|^2 \leq \varepsilon$.

In this model, workers differ across two dimensions. First, each worker has their own local objective function F_i which may differ from one another and the global objective function F. These objective functions vary across workers, typically due to highly heterogeneous local data distributions $\mathcal{D}_i$, as is common in federated learning (McMahan et al., 2017; Caldas et al., 2019; Kairouz et al., 2021; Wang et al., 2021), for example. While early asynchronous literature primarily focused on the homogeneous data setting (e.g., Tyurin & Richtárik, 2023; Maranjyan et al., 2025d), recent extensions to the heterogeneous setting (Mishchenko et al., 2022a; Koloskova et al., 2022; Nguyen et al., 2022) often require restrictive similarity assumptions or algorithmic modifications. In the present work, our focus is on the general case allowing for data heterogeneity ($F_i \neq F_j$) under a relaxed similarity assumption (Assumption 2.5).

Second, workers may also differ in the time it takes them to compute stochastic gradients, due to, for example, hardware differences (Dutta et al., 2018; Li et al., 2020; Maranjyan et al., 2025a;b;c). We denote the time needed for worker i to compute a stochastic gradient by τ_i throughout the text and explicitly allow for these to differ across workers.

In Naive Minibatch SGD (Cotter et al., 2011; Dekel et al., 2012), one of the simplest distributed methods for such problems, each worker i computes a stochastic gradient in τ_i units of time and sends it to the central server. Once the server has received gradients from all workers, gradients

Algorithm 1 Asynchronous SGD

Input: initial point $\mathbf{x}_0 \in \mathbb{R}^d$, stepsizes $\gamma_k > 0$
Set $\mathbf{y}_{0,i} = \mathbf{x}_0, \quad \forall i$
Each worker i begins calculation of $\nabla f_i(\mathbf{y}_{0,i})$
for $k = 0, 1, \ldots$ **do**
  Some worker i_k delivers stochastic gradient $\nabla f_{i_k}(\mathbf{y}_{k,i_k})$
  Server updates $\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla f_{i_k}(\mathbf{y}_{k,i_k})$
  Update worker model:
    $\mathbf{y}_{k+1,i_k} = \mathbf{x}_{k+1}$
    $\mathbf{y}_{k+1,j} = \mathbf{y}_{k,j}, \quad \forall j \neq i_k$
  Worker i_k begins calculating $\nabla f_{i_k}(\mathbf{x}_{k+1})$
end for

are averaged, and the model is updated and sent back to the workers for computation of another round of gradients. Each round thus takes $\tau_{\max} := \max_i \tau_i$ units of time, leaving faster workers idle after delivering their gradients while they wait for the global model update. While this fully synchronized procedure simplifies the theoretical analysis to that of standard SGD, it severely underutilizes fast workers whose idle time could be spent more productively.

In contrast, Asynchronous SGD never synchronizes workers nor delays the application of updates from any workers. Instead, the server updates the model whenever it receives a gradient from a worker, and immediately forwards the updated model to the same worker for computation of their next stochastic gradient. This method is described in Algorithm 1.

Asynchronous SGD avoids the problem of idle workers, but introduces two novel problems: **staleness** and **objective inconsistency**. Because the server updates its model whenever a gradient arrives from any worker, the gradient applied in a given iteration typically does not correspond to the model of the previous iteration; in other words, the server performs model updates with stale gradients. Moreover, as faster workers contribute more gradient updates than slower ones, the search trajectory may be biased towards faster workers' local objective functions, resulting in the wrong objective function being targeted.

To a first-order approximation, computation times may, in many settings, be assumed to be fixed over time. To see how this added structure can be leveraged, let us first consider what happens when Naive Minibatch SGD takes a step of stepsize α: The model $\mathbf{x}^0$ is moved in direction $-\alpha \cdot \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}^0)$, so that the contribution of worker i to this update is $-\alpha/n \nabla f_i(\mathbf{x}^0)$. In expectation, this becomes $-\alpha/n \nabla F_i(\mathbf{x}^0)$.

Next, consider what Asynchronous SGD can do in the same time, and let us assume, for simplicity, that worker i computes K_i stochastic gradients at points $\mathbf{y}_0^0, \ldots, \mathbf{y}_{K_i-1}^0$ in the time needed for one Naive Minibatch SGD update. If each step of worker i has stepsize γ_i, then the contribution of worker i over this time is $-\sum_{k=0}^{K_i-1} \gamma_i \nabla f_i(\mathbf{y}_k^0)$. Provided that the K_i points $\mathbf{y}_0^0, \ldots, \mathbf{y}_{K_i-1}^0$ are close to $\mathbf{x}^0$ and the evaluated gradients do not change abruptly, this contribution is approximately equal to $-K_i\gamma_i \nabla F_i(\mathbf{x}^0)$ in expectation. Comparison with the contribution made by worker i in Naive Minibatch SGD suggests choosing a worker-specific stepsize $\gamma_i = \alpha/nK_i$ to approximate the ideal gradient step $-\alpha \nabla F(\mathbf{x}^0)$ through the sum of all workers' contributions.

The sequence of stochastic gradient steps thus computed by Asynchronous SGD is, in general, not an unbiased estimator of the true gradient step $-\alpha \nabla F(\mathbf{x}^0)$. However, our analysis shows that this bias can be controlled efficiently. Moreover, by letting faster workers take more steps we achieve greater variance reduction and faster convergence.

Our primary conceptual insight is a shift in perspective: Asynchronous methods do not need to approximate the true global gradient direction in every single iteration. Instead, we show that by approximating the global gradient *step*—split across workers and over time—we can establish convergence to an ε-stationary point.

We show that rescaling worker-specific stepsizes (Rescaled ASGD) neutralizes objective inconsistency under data and system heterogeneity. In the fixed-computation model (Tyurin & Richtárik, 2023), Rescaled ASGD achieves a near-optimal wall-clock time complexity to reach an ε-stationary point, matching known lower bounds in the leading term (cf. Table 1). As an additional result of our analysis, we show that Vanilla ASGD targets a frequency-weighted average of the local objectives.

Rescaled ASGD maintains the standard ASGD template (Algorithm 1) without introducing memory overhead, gathering phases, or worker idleness. Through proof-of-concept experiments on heterogeneous data, we confirm that Rescaled ASGD accurately targets the global objective and remains competitive with memory- and synchronization-heavy baselines.

2. Problem Setup

As is standard, we assume the samples ξ used to compute the stochastic gradients $\nabla f_i(\mathbf{x}, \xi)$ to be independent. Moreover, we make the following assumptions:

Assumption 2.1 (Unbiased Gradients)**.** *The stochastic gradients are unbiased, that is, for all $\mathbf{x} \in \mathbb{R}^d$ and all i,*

$$\mathbb{E}_{\xi \sim \mathcal{D}_i}\left[\nabla f_i(\mathbf{x}, \xi)\right] = \nabla F_i(\mathbf{x}).$$

Assumption 2.2 (Bounded Variance)**.** *The stochastic gradients have bounded variance $\sigma^2 \geq 0$, that is, for all $\mathbf{x} \in \mathbb{R}^d$*

and all i,

$$\mathbb{E}_{\xi\sim\mathcal{D}_i}\|\nabla f_i(\mathbf{x},\xi)-\nabla F_i(\mathbf{x})\|^2\leq\sigma^2.$$

Assumption 2.3 (Global Objective Function)**.** *The global objective function F is differentiable and L-smooth, and bounded below by $F^*:=\inf_{\mathbf{x}\in\mathbb{R}^d}F(\mathbf{x})>-\infty$.*

We denote the initial optimality gap by $\Delta:=F(\mathbf{x}^0)-F^*$.

Assumption 2.4 (Local Objective Functions)**.** *Each local objective function F_i is differentiable and L_i-smooth.*

We denote by $L_{\max}:=\max_i L_i$ the maximum of these local smoothness constants.

Assumption 2.5 (Bounded Heterogeneity)**.** *There exist constants $\zeta,\rho\geq 0$ such that, for all $\mathbf{x}\in\mathbb{R}^d$ and all i*

$$\|\nabla F_i(\mathbf{x})\|^2\leq\zeta^2+\rho^2\|\nabla F(\mathbf{x})\|^2.$$

3. Asynchronous SGD with Cyclic Update Schedule and Rescaled ASGD

Following Mishchenko et al. (2022a); Tyurin & Richtárik (2023), we assume worker i to take a fixed amount of time, τ_i, to compute one stochastic gradient. These times may differ across workers, but do not vary over time for a given worker. To facilitate the theoretical analysis and derive the rigorous convergence guarantees in Section 4, we must ensure that workers deliver a fixed number of updates over a given time horizon. To that end, we make the following assumption:

Assumption 3.1 (Harmonic Periods)**.** *The set of computation times $\{\tau_1,\ldots,\tau_n\}$ satisfies $\frac{\tau_i}{\tau_j}\in\mathbb{N}$ or $\frac{\tau_j}{\tau_i}\in\mathbb{N}$ for all $i,j\in\{1,\ldots,n\}$.*

Under Assumption 3.1, Asynchronous SGD as presented in Algorithm 1 becomes more structured. Assuming, for ease of exposition, that the server processes updates delivered by multiple workers at the same time in a fixed order, the sequence of worker indices i_k is fully deterministic and follows a cyclic update schedule. After $\tau_{\max}$ time units, each worker will have sent $K_i:=\frac{\tau_{\max}}{\tau_i}$ updates in a fixed order that will be repeated exactly over the next $\tau_{\max}$ time units.

We refer to one such pass as a *cycle*, to $\tau_{\max}$ as the (wall-clock) cycle duration, and define $K:=\sum_{i=1}^n K_i$ as the total number of updates received from all workers over the course of one cycle. We may now reformulate Algorithm 1 in this more structured form as shown in Algorithm 2. Here, m denotes the cycle and k the iteration within a cycle.

Algorithm 2 initializes a shared point $\mathbf{x}^0$, after which all workers proceed asynchronously without any synchronization or idle periods. We refer to $\{\mathbf{x}^m\}_{m\in\mathbb{N}}$ as the cycle iterates and to $\{\mathbf{x}_k^m\}_{k=1,\ldots,K}$ as the inner iterates within cycle m.

Algorithm 2 Asynchronous SGD with Cyclic Update Schedule

Input: initial point $\mathbf{x}^0\in\mathbb{R}^d$, stepsizes $\gamma_k^m>0$
Set $\mathbf{x}_0^0=\mathbf{x}^0$
Set $\mathbf{y}_{0,i}^0=\mathbf{x}^0,\quad\forall i$
Each worker i begins calculation of $\nabla f_i(\mathbf{y}_{0,i}^0)$
for $m=0,1,2,\ldots$ **do**
  for $k=0,\ldots,K-1$ **do**
    Worker i_k delivers stochastic gradient $\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)$
    Server updates $\mathbf{x}_{k+1}^m=\mathbf{x}_k^m-\gamma_k^m\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)$
    Update worker model:
      $\mathbf{y}_{k+1,i_k}^m=\mathbf{x}_{k+1}^m$
      $\mathbf{y}_{k+1,j}^m=\mathbf{y}_{k,j}^m,\quad\forall j\neq i_k$
    Worker i_k begins calculating $\nabla f_{i_k}(\mathbf{x}_{k+1}^m)$
  end for
  Set $\mathbf{y}_{0,i}^{m+1}=\mathbf{y}_{K,i}^m,\quad\forall i$
  Set $\mathbf{x}^{m+1}=\mathbf{x}_K^m$
  Set $\mathbf{x}_0^{m+1}=\mathbf{x}_K^m$
end for

Let $\mathbf{y}_{k,i}^m$ be the local model held by worker i at the start of inner iteration k in cycle m. This model corresponds to a past server iterate $\mathbf{x}_{k'}^{m'}$, originating either from the previous cycle ($m'=m-1$, $k'\geq k$) or the current one ($m'=m$, $k'<k$). As a result, the gradient staleness, or delays—that is, the number of server updates between a worker reading the model and applying its gradient—is bounded by K. Bounded staleness of this form is a well-established setting in the asynchronous optimization literature (Agarwal & Duchi, 2011; Recht et al., 2011; Lian et al., 2015).

Our proposed method sets the stepsizes in Algorithm 2 to $\gamma_k^m=\gamma_{i_k}\propto\tau_{i_k}$. By doing so, even though the number of gradients delivered varies across workers, the aggregate stepsizes taken along each worker's descent direction over a cycle, $\gamma_i K_i=\gamma_i\cdot\tau_{\max}/\tau_i$, are equal. The accumulated gradient update taken by Rescaled ASGD over the course of a cycle m can then be decomposed (cf. Lemma C.7) into the exact scaled global gradient $\alpha\nabla F(\mathbf{x}^m)$ with $\alpha:=\sum_{k=0}^{K-1}\gamma_{i_k}$ being the cycle stepsize, a bias term, and a noise term that vanishes in expectation. As we will see next, the effect of the bias can be controlled efficiently.

4. Convergence Guarantees

Our main result establishes that Rescaled ASGD with properly chosen, worker-specific stepsizes targets the equal-weighted average F defined in (1). For a chosen stepsize parameter $\gamma>0$, we set the worker-specific stepsizes as

$$\gamma_i:=\gamma\cdot\tau_i\cdot\frac{\tau_H}{n\tau_{\max}},\tag{2}$$

where the last factor serves as a normalizing constant. Under this rescaling, we obtain the following convergence guarantee.

Theorem 4.1 (Convergence to the Equal-Weighted Average)**.** *Let the worker-specific stepsizes be chosen according to (2) and suppose Assumptions 2.1 to 2.5 and 3.1 hold. If the stepsize parameter satisfies* $0 < \gamma \leq \min\left\{\frac{1}{6L\tau_H}, \frac{1}{5L_{\max}\rho\tau_{\max}}\right\}$*, then, after* $M \geq 1$ *cycles, the cycle iterates* $\mathbf{x}^m$ *generated by Algorithm 2 satisfy*

$$\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right] \leq c_0 \cdot \frac{\Delta}{\gamma\tau_H M} + c_1 \cdot \frac{\gamma\tau_A L\sigma^2}{K} + c_2 \cdot \gamma^2\tau_A^2 L_{\max}^2\left(\sigma^2+\zeta^2\right)$$

for some absolute constants $c_0, c_1, c_2 > 0$*.*

The last term reflects the effect of the cycle bias and scales with γ^2. By scaling down the stepsize parameter γ, we can shrink this term and achieve ε-stationarity for arbitrarily small $\varepsilon > 0$. The following corollary provides a bound on the worst-case time complexity to reach such an ε-stationary point.

Corollary 4.2 (Time Complexity for the Equal-Weighted Average)**.** *In the setting of Theorem 4.1, the worst-case wall-clock time complexity to find an* ε*-stationary point of* F *is*

$$\mathcal{O}\left(\frac{\Delta L\sigma^2}{n\varepsilon^2}\tau_A + \frac{\Delta L_{\max}\sqrt{\sigma^2+\zeta^2}}{\varepsilon^{1.5}}\frac{\tau_{\max}}{\tau_H}\tau_A + \frac{\Delta L_{\max}\rho}{\varepsilon}\frac{\tau_{\max}}{\tau_H}\tau_{\max} + \frac{\Delta L}{\varepsilon}\tau_{\max}\right).$$

For small ε, the leading term dominates the complexity bound. This scales efficiently with the arithmetic mean of computation times, τ_A, rather than their maximum, $\tau_{\max}$, and matches that of the theoretical lower bound (see Table 1). The local function parameters ($L_{\max}$) and data heterogeneity measures (ζ, ρ) affect only lower-order terms, showing that the slowdown caused by function heterogeneity and gradient staleness is not of first-order concern.

Objective Inconsistency under Equal Stepsizes Our analysis allows us to precisely quantify the objective inconsistency for other stepsize choices. In Vanilla ASGD, all workers use the same stepsize $\gamma_i = \gamma/K$, so that $\gamma_i/\tau_i \propto 1/\tau_i$. Consequently, Algorithm 2 now targets a frequency-weighted average of the local functions,

$$\tilde{F}(\mathbf{x}) := \sum_{i=1}^n \tilde{w}_i F_i(\mathbf{x}), \quad \tilde{w}_i := \frac{\tau_i^{-1}}{\sum_{j=1}^n \tau_j^{-1}} \propto \tau_i^{-1}. \quad (3)$$

Our next theorem makes this precise.

Theorem 4.3 (Convergence to the Frequency-Weighted Average)**.** *Let the workers' stepsizes be equal,* $\gamma_i = \gamma/K$*, and suppose Assumptions 2.1 to 2.5 and 3.1 hold for* $\tilde{F}$*.*[1] *If the stepsize parameter satisfies* $0 < \gamma \leq \min\left\{\frac{1}{6\tilde{L}}, \frac{1}{5L_{\max}\tilde{\rho}}\right\}$*, then, after* $M \geq 1$ *cycles, the cycle iterates* $\mathbf{x}^m$ *generated by Algorithm 2 satisfy*

$$\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla\tilde{F}(\mathbf{x}^m)\right\|^2\right] \leq c_0 \cdot \frac{\tilde{\Delta}}{\gamma M} + c_1 \cdot \frac{\gamma\tilde{L}\sigma^2}{K} + c_2 \cdot \gamma^2 L_{\max}^2\left(\sigma^2+\tilde{\zeta}^2\right)$$

for some absolute constants $c_0, c_1, c_2 > 0$*.*

To better understand the workings of Algorithm 2 with equal stepsizes, we must consider its wall-clock time complexity.

Corollary 4.4 (Time Complexity for the Frequency-Weighted Average)**.** *In the setting of Theorem 4.3, the worst-case wall-clock time complexity to find an* ε*-stationary point of* $\tilde{F}$ *is*

$$\mathcal{O}\left(\frac{\tilde{\Delta}\tilde{L}\sigma^2}{n\varepsilon^2}\tau_H + \frac{\tilde{\Delta}L_{\max}\sqrt{\sigma^2+\tilde{\zeta}^2}}{\varepsilon^{1.5}}\tau_{\max} + \frac{\tilde{\Delta}L_{\max}\tilde{\rho}}{\varepsilon}\tau_{\max} + \frac{\tilde{\Delta}\tilde{L}}{\varepsilon}\tau_{\max}\right).$$

Note that the leading term of this complexity bound now scales with the harmonic mean τ_H. This captures the price of neutralizing objective inconsistency: In Vanilla ASGD, fast workers take unscaled steps, driving the global model forward at a higher rate governed by the harmonic mean of the workers' computation speeds. With rescaled stepsizes, on the other hand, stepsizes of fast workers are shrunk ($\gamma_i \propto \tau_i$) to prevent them from dominating the optimization trajectory. While this ensures convergence to the equal-weighted average, scaling down the updates from faster workers slows the global learning progress, shifting the time complexity bottleneck from the smaller harmonic mean τ_H to the larger arithmetic mean τ_A.

5. Conclusion

In this work, we introduced Rescaled ASGD to address the problem of objective inconsistency in asynchronous optimization caused by the interplay of data and system heterogeneity. By proportionally rescaling worker-specific stepsizes, we proved that Asynchronous SGD converges to the true global objective without introducing memory overhead, gathering phases, or worker idle times.

[1] That is, we replace $F, \Delta, L, \zeta, \rho$ by $\tilde{F}, \tilde{\Delta}, \tilde{L}, \tilde{\zeta}, \tilde{\rho}$ in Assumptions 2.3 and 2.5.

Acknowledgments

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

Impact Statement

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

References

Agarwal, A. and Duchi, J. C. Distributed delayed stochastic optimization. *Advances in Neural Information Processing Systems*, 24, 2011.

Baudet, G. M. Asynchronous iterative methods for multiprocessors. *Journal of the ACM (JACM)*, 25(2):226–244, 1978.

Bertsekas, D. and Tsitsiklis, J. *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., and Talwalkar, A. LEAF: A Benchmark for Federated Settings, December 2019. arXiv:1812.01097.

Chazan, D. and Miranker, W. Chaotic relaxation. *Linear algebra and its applications*, 2(2):199–222, 1969.

Cotter, A., Shamir, O., Srebro, N., and Sridharan, K. Better mini-batch algorithms via accelerated gradient methods. *Advances in Neural Information Processing Systems*, 24, 2011.

Dekel, O., Gilad-Bachrach, R., Shamir, O., and Xiao, L. Optimal Distributed Online Prediction using Mini-Batches, January 2012. arXiv:1012.1367.

Dutta, S., Joshi, G., Ghosh, S., Dube, P., and Nagpurkar, P. Slow and stale gradients can win the race: Error-runtime trade-offs in distributed SGD. In *International Conference on Artificial Intelligence and Statistics*, pp. 803–812. PMLR, 2018.

Gorbunov, E., Hanzely, F., and Richtárik, P. Local SGD: Unified Theory and New Efficient Methods, November 2020. arXiv:2011.02828.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification, September 2019. arXiv:1909.06335.

Islamov, R., Safaryan, M., and Alistarh, D. AsGrad: A sharp unified analysis of asynchronous-SGD algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 649–657. PMLR, 2024.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., and others. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning, April 2021. arXiv:1910.06378.

Koloskova, A., Stich, S. U., and Jaggi, M. Sharper convergence guarantees for asynchronous SGD for distributed and federated learning. *Advances in Neural Information Processing Systems*, 35:17202–17215, 2022.

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

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

Lian, X., Huang, Y., Li, Y., and Liu, J. Asynchronous parallel stochastic gradient for nonconvex optimization. *Advances in Neural Information Processing Systems*, 28, 2015.

Mangasarian, L. Parallel gradient distribution in unconstrained optimization. *SIAM Journal on Control and Optimization*, 33(6):1916–1925, 1995.

Maranjyan, A. and Richtárik, P. Ringleader ASGD: The First Asynchronous SGD with Optimal Time Complexity under Data Heterogeneity. In *The Fourteenth International Conference on Learning Representations*, 2026.

Maranjyan, A., Omar, O. S., and Richtárik, P. MindFlayer SGD: Efficient Parallel SGD in the Presence of Heterogeneous and Random Worker Compute Times. In *The 41st Conference on Uncertainty in Artificial Intelligence*, 2025a.

Maranjyan, A., Saad, E. M., Richtárik, P., and Orabona, F. ATA: Adaptive Task Allocation for Efficient Resource Management in Distributed Machine Learning. In *International Conference on Machine Learning*, 2025b.

Maranjyan, A., Safaryan, M., and Richtárik, P. GradSkip: Communication-Accelerated Local Gradient Methods with Better Computational Complexity. *Transactions on Machine Learning Research*, 2025c. ISSN 2835-8856.

Maranjyan, A., Tyurin, A., and Richtárik, P. Ringmaster ASGD: The First Asynchronous SGD with Optimal Time Complexity. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 43120–43139. PMLR, October 2025d.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Mishchenko, K., Bach, F., Even, M., and Woodworth, B. E. Asynchronous SGD beats minibatch SGD under arbitrary delays. *Advances in Neural Information Processing Systems*, 35:420–433, 2022a.

Mishchenko, K., Malinovsky, G., Stich, S., and Richtárik, P. ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally! In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15750–15769. PMLR, July 2022b.

Nesterov, Y. *Lectures on Convex Optimization*, volume 137. Springer, 2018.

Nguyen, J., Malik, K., Zhan, H., Yousefpour, A., Rabbat, M., Malek, M., and Huba, D. Federated learning with buffered asynchronous aggregation. In *International Conference on Artificial Intelligence and Statistics*, pp. 3581–3607. PMLR, 2022.

Recht, B., Re, C., Wright, S., and Niu, F. Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

Stich, S. U. Local SGD Converges Fast and Communicates Little, May 2019. arXiv:1805.09767.

Tyurin, A. and Richtárik, P. Optimal Time Complexities of Parallel Stochastic Optimization Methods Under a Fixed Computation Model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 16515–16577, 2023.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pp. 7611–7623, Red Hook, NY, USA, December 2020. Curran Associates Inc. ISBN 978-1-7138-2954-6.

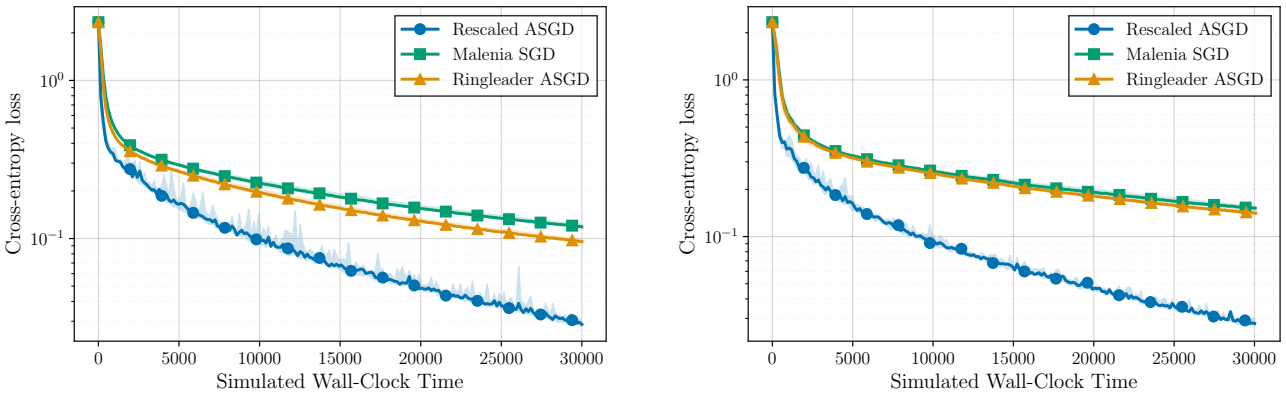
Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., and others. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

Woodworth, B., Patel, K. K., Stich, S., Dai, Z., Bullins, B., Mcmahan, B., Shamir, O., and Srebro, N. Is local SGD better than minibatch SGD? In *International Conference on Machine Learning*, pp. 10334–10343. PMLR, 2020.

Xie, C., Koyejo, S., and Gupta, I. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*, 2019.

A. Experiments

We compare Rescaled ASGD against Malenia SGD and Ringleader ASGD, two state-of-the-art methods with optimal or near-optimal theoretical wall-clock time complexities, in a data-heterogeneous setup. We train a two-layer neural network on MNIST (LeCun et al., 2010). To enforce maximal heterogeneity, we partition the data by label (Hsu et al., 2019) across $n = 10$ workers, such that each worker holds images from exactly one class.



(a) Fixed Times. *(b)* Fluctuating Times.

Figure 1. Solid lines denote the median loss across five random seeds, with shaded regions indicating the minimum and maximum. Malenia SGD and Ringleader ASGD are slowed down under fluctuating computation times as the gradient gathering phase takes longer. Rescaled ASGD shows virtually identical performance in both settings.

We evaluate two computation time settings. First, fixed harmonic periods (Assumption 3.1) matching our theoretical setup. We assign $\tau_i \in \{1, 2, 4, 8, 16\}$ to two workers each, making the fastest workers 16 times faster than the slowest. Second, a fluctuating setting where computation times of worker i are sampled from an exponential distribution with mean τ_i. While outside our theoretical framework, the underlying mechanism of equalized long-run learning progress remains.

Figure 1 shows the loss trajectory. Under fluctuating times, Malenia SGD and Ringleader ASGD degrade slightly due to stragglers slowing the gathering phase, whereas Rescaled ASGD remains unaffected.

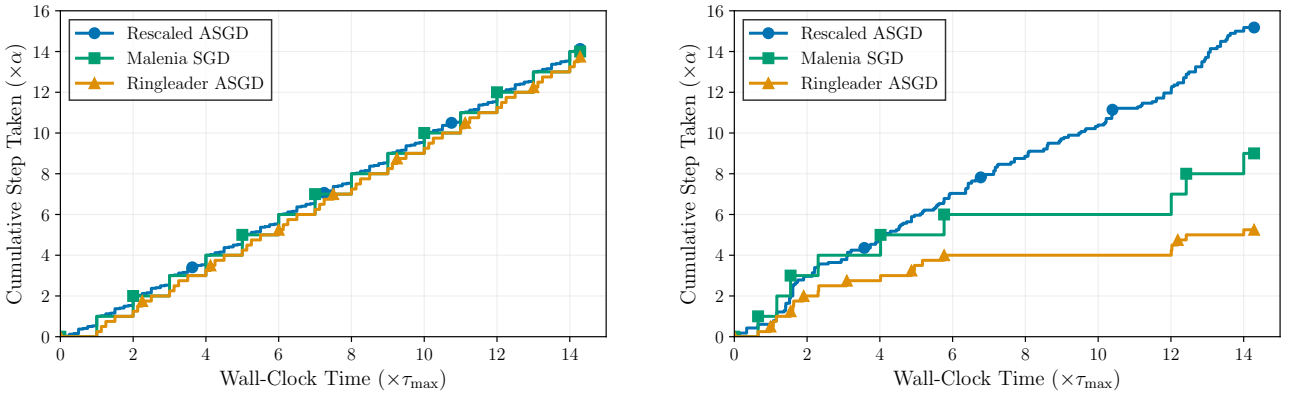
To illustrate algorithmic differences, Figure 2 plots the cumulative stepsizes over time. For a fair comparison, worker i takes a step of $\alpha\tau_i/n\tau_{\max}$ in Rescaled ASGD, ensuring the total stepsize per cycle, taking $\tau_{\max}$ units of time, is α. Ringleader ASGD workers take steps of α/n to match this sum, while Malenia SGD requires no rescaling as it takes one step per cycle.

In the fixed setup (a), all methods share an average progress rate of α per $\tau_{\max}$ units of time, with Rescaled ASGD and Ringleader ASGD yielding smoother trajectories. Under fluctuating times (b), Malenia SGD and Ringleader ASGD are bottlenecked by the slowest worker due to their gathering phases. Although faster workers compute continuously, the server model stalls until the straggler finishes, making the expected gathering duration greater than $\tau_{\max}$. Rescaled ASGD continuously applies updates, exploring the search space unimpeded.

B. Related Work

Asynchronous Optimization Asynchronous methods have been a cornerstone of parallel computing for over half a century (Chazan & Miranker, 1969; Baudet, 1978; Bertsekas & Tsitsiklis, 2015). Early research focused primarily on the homogeneous data setting, where the central challenge is managing stale gradients to ensure they do not derail the learning trajectory. In these settings, the assumption of bounded delays has proven a fruitful abstraction for analysis (Agarwal & Duchi, 2011; Recht et al., 2011), a property that is naturally satisfied within the structured computation model we employ. The more complex interplay between asynchronous updates and heterogeneous data has received comparatively less attention.

Mishchenko et al. (2022a) examine a framework with arbitrary, unbounded delays. While their results for homogeneous data yield convergence rates to ε-stationary points comparable to ours, the heterogeneous case presents a fundamental challenge. Specifically, if worker computation times can vary without structure, the optimization trajectory becomes biased



(a) Fixed. *(b)* Fluctuating.

Figure 2. Cumulative stepsize taken over wall-clock time. The gathering phases dilate under fluctuating computation times, causing Malenia SGD and Ringleader ASGD to stall in the search space. Rescaled ASGD takes steps at an unimpeded rate of $\alpha/\tau_{\max}$.

toward faster workers. Consequently, their guarantees in this setting necessitate a strong similarity assumption between local objectives and do not allow for convergence to an arbitrary ε-stationary point. By contrast, the structured nature of our computation model allows us to neutralize this bias through stepsize rescaling, enabling stronger guarantees without requiring local objectives to be nearly identical.

Koloskova et al. (2022) also explore asynchronous SGD under heterogeneous data but utilize a distinct mechanism that deviates from the standard Asynchronous SGD formulation (Algorithm 1). In their model, the server assigns gradient computation tasks to workers via uniform random sampling. This process allows for multiple gradient tasks to be queued for a single worker. While this ensures that all workers contribute an equal number of gradients in the long run–matching the core motivation of our approach–it does so without utilizing faster workers' speed to increase throughput. In their model, faster workers are characterized by lower average delays rather than higher update frequencies. While this enables convergence to arbitrary ε-stationarity, the queuing mechanism does not reflect the common physical reality of asynchronous systems where workers deliver results as soon as they are computed and the server applies them as soon as they arrive. Our approach, Rescaled ASGD, allows workers to operate at their maximum physical frequency while using rescaled stepsizes to ensure that the resulting trajectory targets the correct global objective.

Objective Inconsistency The phenomenon wherein distributed methods fail to minimize the global objective (1) when workers have different computation times is a well-documented challenge in distributed optimization (Wang et al., 2020; Islamov et al., 2024). This inconsistency arises because the stochastic process underlying the optimization becomes implicitly weighted by the relative frequencies of worker updates, effectively targeting a surrogate objective $\tilde{F}$ rather than the equal-weighted average F.

Wang et al. (2020) address this challenge within the context of *local SGD*, a framework where workers perform multiple local updates before periodically synchronizing with a central server (Mangasarian, 1995; Stich, 2019; Gorbunov et al., 2020; Woodworth et al., 2020). In their analysis, the number of local steps taken by a worker is intrinsically linked to its computation speed. To neutralize the resulting bias, they propose a mechanism conceptually similar to ours: rescaling the aggregate model difference from each worker by the reciprocal of the number of local steps performed. This normalization ensures that fast workers do not pull the global model disproportionately toward their local optima, thereby enabling the algorithm to reach arbitrary ε-stationarity for the true objective.

Other strategies designed to mitigate bias from heterogeneous worker participation include the use of proximal regularization to constrain local drift (Li et al., 2020), control variates to correct for client-server residuals (Karimireddy et al., 2021; Mishchenko et al., 2022b), and staleness-aware mixing coefficients that de-weight delayed updates (Xie et al., 2019). While effective at improving stability, these methods typically focus on reducing the variance or the impact of stale information rather than enforcing the structural update parity required to eliminate objective inconsistency. In the absence of explicit rescaling or contribution equalization, the underlying optimization target remains skewed toward more active participants.

Our work extends the rescaling principle to the asynchronous, single-update regime, providing a memory-efficient solution.

Table 1. Comparison of worst-case wall-clock time complexities for parallel stochastic first-order methods in the fixed-computation time model with data and system heterogeneity to achieve ε-stationarity in (1). We denote the arithmetic mean and maximum, respectively, of the workers' computation times by $\tau_A, \tau_{\max}$. Problem parameters include the initial optimality gap $\Delta \coloneqq F(\mathbf{x}^0) - F^*$ and global smoothness constant L (Assumption 2.3), the target stationarity $\varepsilon > 0$, and a bound on the stochastic gradient variance σ^2 (Assumption 2.2). **Asymptotic Optimality:** time complexity matches the lower bound established by Tyurin & Richtárik (2023) in the leading term and achieves ε-stationarity for arbitrarily small ε. **No Idle Workers:** all workers remain busy and computational resources are fully utilized. **No Memory Overhead:** no gradients or model iterates need to be stored for later use.

Method (Reference)	Time Complexity (Leading Term)	Asymptotic Optimality	No Idle Workers	No Memory Overhead
Naive Minibatch SGD (Cotter et al., 2011; Dekel et al., 2012)	$\frac{\Delta L \sigma^2}{n\varepsilon^2}\tau_{\max}$	✘	✘	✔
Malenia SGD (Tyurin & Richtárik, 2023)	$\frac{\Delta L \sigma^2}{n\varepsilon^2}\tau_A$	✔	✔	✘
Ringleader ASGD (Maranjyan & Richtárik, 2026)	$\frac{\Delta L' \sigma^2}{n\varepsilon^2}\tau_A$ (†)	✘	✔	✘
Concurrent ASGD (Koloskova et al., 2022)	$\frac{\Delta L (\sigma^2+\zeta^2)}{n\varepsilon^2}\tau_{\max}$ (‡)	✘	✘	✘
Delay-Adaptive ASGD (Mishchenko et al., 2022a)	$\frac{\Delta L \sigma^2}{n\varepsilon^2}\tau_A$	✘(‡)	✔	✔
Rescaled ASGD Theorem 4.1, Corollary 4.2	$\frac{\Delta L \sigma^2}{n\varepsilon^2}\tau_A$	✔	✔	✔

(†) Maranjyan & Richtárik (2026) rely on a constant $L' \geq L$ to bound data heterogeneity. In certain settings, $L' = \mathcal{O}(L)$, and Ringleader ASGD achieves optimal time complexity.
(‡) Koloskova et al. (2022); Mishchenko et al. (2022a) rely on a stricter assumption on the data heterogeneity, $\|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \zeta^2$. In Concurrent ASGD, the constant ζ^2 affects the leading term of the time complexity. In Delay-Adaptive ASGD, the gradient norm can be bounded only up to this constant, therefore not achieving arbitrary ε-stationarity, and thus no asymptotic optimality.

Optimal Methods in the Fixed-Computation Model The fixed-computation model offers a benchmark for evaluating the efficiency of parallel optimization algorithms. Tyurin & Richtárik (2023) derive a lower bound for the wall-clock time complexity of first-order stochastic gradient methods when worker computation times τ_i are constant yet heterogeneous. While they propose synchronous methods that achieve this bound in both homogeneous and heterogeneous settings, their approach inherently sacrifices the continuous throughput of asynchronous systems to maintain synchronization. Notably, in the homogeneous data setting, Vanilla ASGD can match this lower bound in regimes with high stochastic gradient variance σ^2 relative to the target precision ε.

Maranjyan et al. (2025d) improve upon these results for the homogeneous setting by introducing a variant of Asynchronous SGD that employs delay-adaptive learning rates and a hard-thresholding rule. By discarding gradients with staleness exceeding a specific threshold, they demonstrate that staleness bias can be actively managed to achieve optimal convergence regardless of the gradient variance. However, such a thresholding mechanism is fundamentally incompatible with the heterogeneous data setting, where it could systematically exclude updates from the slowest workers, thereby introducing the very objective inconsistency our work seeks to resolve.

Maranjyan & Richtárik (2026) propose an asynchronous algorithm, Ringleader ASGD, designed specifically for the heterogeneous regime. While their method approaches the optimal time complexity, its leading term is still constrained by a constant related to local objective similarity, similar to the limitations observed in the work of Koloskova et al. (2022) (cf. Table 1). In contrast, Rescaled ASGD targets the equal-weighted average through stepsize rescaling rather than delay manipulation, gradient tables, or worker selection. This allows us to exploit the higher update frequency of faster workers, a benefit also noted in the empirical performance of Maranjyan & Richtárik (2026). By ensuring that every worker's

contribution is appropriately weighted in the model space, we maintain the exploration advantages of frequent asynchronous updates.

C. Proofs

C.1. Preliminaries

Notation In our model, randomness enters through the stochastic gradients $\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m, \xi_{k,i_k}^m)$ alone. For ease of notation, we drop the dependence on the samples $\xi_{k,i_k}^m \sim \mathcal{D}_{i_k}$ and write $\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)$ instead. We denote by $\mathcal{F}^m$ the sigma field generated by the stochastic gradients delivered up to the beginning of cycle m. Likewise, $\mathcal{F}_k^m$ denotes the sigma field generated by the stochastic gradients delivered up to the beginning of iteration k in cycle m, and define $\mathcal{F}_K^m \coloneqq \mathcal{F}^{m+1}$. The iterates $\mathbf{x}_k^m$ are $\mathcal{F}_k^m$-measurable, as are $\mathbf{y}_k^m$, while the stochastic gradients $\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)$ are $\mathcal{F}_{k+1}^m$-measurable as these are resolved only at the end of iteration k in cycle m.

$\mathbb{E}\left[\bar{\mathbf{x}}\right]$ denotes the unconditional expectation of a random vector $\bar{\mathbf{x}}$, $\mathbb{E}_m\left[\bar{\mathbf{x}}\right]$ the conditional expectation conditional on all information available at the beginning of cycle m, i.e., conditioned on the sigma field $\mathcal{F}^m$, $\mathbb{E}_{m,k}\left[\bar{\mathbf{x}}\right]$ the conditional expectation conditional on all information available at the beginning of iteration k within cycle m, i.e., conditioned on the sigma field $\mathcal{F}_k^m$.

Similarly, $\mathbb{V}_m\left(\bar{\mathbf{x}}\right) \coloneqq \mathbb{E}_m[\|\bar{\mathbf{x}} - \mathbb{E}_m\left[\bar{\mathbf{x}}\right]\|^2] = \mathbb{E}_m[\|\bar{\mathbf{x}}\|^2] - \|\mathbb{E}_m[\bar{\mathbf{x}}]\|^2$ denotes the conditional variance of a random vector $\bar{\mathbf{x}}$ conditioned on all information available at the beginning of cycle m, and $\mathbb{C}_m\left(\bar{\mathbf{x}}, \bar{\mathbf{y}}\right) \coloneqq \mathbb{E}_m\left[\langle \bar{\mathbf{x}} - \mathbb{E}_m\left[\bar{\mathbf{x}}\right], \bar{\mathbf{y}} - \mathbb{E}_m\left[\bar{\mathbf{y}}\right]\rangle\right] = \mathbb{E}_m\left[\langle \bar{\mathbf{x}}, \bar{\mathbf{y}}\rangle\right] - \langle \mathbb{E}_m\left[\bar{\mathbf{x}}\right], \mathbb{E}_m\left[\bar{\mathbf{y}}\right]\rangle$ denotes the conditional covariance of two random vectors $\bar{\mathbf{x}}, \bar{\mathbf{y}}$.

We define $L_{\max} = \max_{i=1,\dots,n} L_i, \gamma_{\max} \coloneqq \max_{i=1,\dots,n} \gamma_i, \tau_{\max} \coloneqq \max_{i=1,\dots,n} \tau_i$. Moreover, $\tau_A \coloneqq \frac{1}{n}\sum_{i=1}^n \tau_i$ and $\tau_H \coloneqq \frac{n}{\sum_{i=1}^n \tau_i^{-1}}$ denote the arithmetic and harmonic mean of computation times, respectively.

Frequently used Assumptions, Definitions, and Standard Results Below, we state a series of standard results, definitions, and the assumptions laid out in Section 2 in simplified form for reference. These will be used extensively in the following proofs.

Unbiased Gradients:

$$\mathbb{E}_{\xi\sim D_i}\left[\nabla f_i(\mathbf{x}, \xi)\right] = \nabla F_i(\mathbf{x}) \tag{UG}$$

Bounded Gradient Variance:

$$\mathbb{E}_{\xi\sim D_i}\left[\left\|\nabla f_i(\mathbf{x}, \xi) - \nabla F_i(\mathbf{x})\right\|^2\right] \leq \sigma^2 \tag{BV}$$

Smoothness:

$$\left\|\nabla F_i(\mathbf{x}) - \nabla F_i(\mathbf{y})\right\| \leq L_i \left\|\mathbf{x} - \mathbf{y}\right\|, \left\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\right\| \leq L \left\|\mathbf{x} - \mathbf{y}\right\| \tag{LS}$$

Lower Bound:

$$F(\mathbf{x}^0) - F(\mathbf{x}^M) \geq F(\mathbf{x}^0) - F^* = \Delta \tag{LB}$$

Bounded Data Heterogeneity:

$$\left\|\nabla F_i(\mathbf{x})\right\|^2 \leq \zeta^2 + \rho^2 \left\|\nabla F(\mathbf{x})\right\|^2 \tag{BH}$$

Cycle Stepsize:

$$\alpha \coloneqq \textstyle\sum_{k=0}^{K-1} \gamma_{i_k} \tag{α}$$

Sum of Squared Stepsizes:

$$A \coloneqq \textstyle\sum_{k=0}^{K-1} \gamma_{i_k}^2 \tag{A}$$

Cycle Noise:

$$\boldsymbol{\nu}_m \coloneqq \textstyle\sum_{k=0}^{K-1} \gamma_{i_k} \left(\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m) - \nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)\right) \tag{N}$$

Cycle Bias:

$$\mathbf{b}_m \coloneqq \textstyle\sum_{k=0}^{K-1} \gamma_{i_k} \left(\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m) - \nabla F_{i_k}(\mathbf{x}^m)\right) \tag{B}$$

Triangle Inequality:

$$\|\sum_{i=1}^n \mathbf{v}_i\| \leq \sum_{i=1}^n \|\mathbf{v}_i\| \tag{T}$$

Cauchy-Schwarz Inequality:

$$\left(\sum_{i=1}^d u_i v_i\right)^2 \leq \left(\sum_{i=1}^d u_i^2\right)\left(\sum_{i=1}^d v_i^2\right) \tag{CS}$$

Tower Property:

$$\mathbb{E}_m\left[\mathbb{E}_{m,k}\left[\bar{\mathbf{x}}\right]\right] = \mathbb{E}_m\left[\bar{\mathbf{x}}\right] \tag{TP}$$

Descent Lemma:

$$F(\mathbf{y}) \leq F(\mathbf{x}) + \langle \nabla F(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \tfrac{L}{2}\|\mathbf{y}-\mathbf{x}\|^2 \tag{DL}$$

Jensen's Inequality:

$$\|\mathbb{E}_m\left[\bar{\mathbf{x}}\right]\|^2 \leq \mathbb{E}_m\left[\|\bar{\mathbf{x}}\|^2\right] \tag{JN}$$

Young's Inequality:

$$\|\mathbf{u}\|\,\|\mathbf{v}\| \leq \tfrac{s}{2}\|\mathbf{u}\|^2 + \tfrac{1}{2s}\|\mathbf{v}\|^2, \quad \forall s > 0 \tag{YN}$$

Squared Sum Inequality:

$$\|\sum_{i=1}^n \mathbf{v}_i\|^2 \leq n\sum_{i=1}^n \|\mathbf{v}_i\|^2 \tag{SS}$$

Proof Outline Theorem C.9 states our main result in a more general form, showing that Rescaled ASGD can target any convex combination (4) of the local objective functions, and establishes a bound on the expected average squared gradient norm of the cycle iterates. To that end, Lemma C.8 establishes a bound in terms of the squared norm of the cycle bias utilizing a standard descent lemma. A bound on the norm of the expected cumulative cycle bias is derived in Lemmas C.2 to C.6. Similarly, Lemma C.1 establishes a bound on the squared norm of the cycle noise.

Theorems 4.1 and 4.3 restate Theorem C.9 with the appropriately chosen stepsizes. Corollaries 4.2 and 4.4 then follow immediately.

C.2. Auxiliary Results

Throughout this subsection, we denote by

$$F(x) \coloneqq \sum_{i=1}^n w_i F_i(\mathbf{x}) \tag{4}$$

an arbitrary convex combination of the local objective functions F_i. The weights $w_i \geq 0$ sum to unity and are understood to be fixed constants. Assumptions 2.3 and 2.5 are understood to refer to the objective function defined in (4).

Note that we recover the equal-weighted average (1) by setting $w_i = 1/n$. The more general case presented in this subsection allows us to derive the results from the main text (Theorems 4.1 and 4.3) as special cases of the more general Theorem C.9.

Lemma C.1 (Cycle Noise)**.** *Under assumptions 2.1 and 2.2, the cycle noise*

$$\boldsymbol{\nu}_m \coloneqq \sum_{k=0}^{K-1} \gamma_{i_k}\left(\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m) - \nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)\right)$$

satisfies

$$\mathbb{E}_m\left[\boldsymbol{\nu}_m\right] = 0, \qquad \mathbb{E}_m\left[\|\boldsymbol{\nu}_m\|^2\right] \leq A\sigma^2.$$

Proof. Recall that $\mathbf{y}_{k,i_k}^m$ denotes the model held by worker i_k at the beginning of iteration k of cycle m. Thus,

$$\begin{aligned}\mathbb{E}_m\left[\boldsymbol{\nu}_m\right] &\overset{\text{(N)}}{=} \mathbb{E}_m\left[\sum_{k=0}^{K-1}\gamma_{i_k}\left(\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)-\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)\right)\right]\\ &\overset{\text{(TP)}}{=} \sum_{k=0}^{K-1}\gamma_{i_k}\mathbb{E}_m\left[\mathbb{E}_{m,k}\left[\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)-\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)\right]\right]\\ &\overset{\text{(UG)}}{=} \sum_{k=0}^{K-1}\gamma_{i_k}\mathbb{E}_m\left[0\right]\\ &= 0.\end{aligned}$$

Further,

$$\begin{aligned}\mathbb{E}_m\left[\left\|\boldsymbol{\nu}_m\right\|^2\right] &= \mathbb{V}_m\left(\boldsymbol{\nu}_m\right)+\left\|\mathbb{E}_m\left[\boldsymbol{\nu}_m\right]\right\|^2\\ &= \mathbb{V}_m\left(\sum_{k=0}^{K-1}\gamma_{i_k}\left(\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)-\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)\right)\right)\\ &= \sum_{k=0}^{K-1}\sum_{l=0}^{K-1}\gamma_{i_k}\gamma_{i_l}\mathbb{C}_m\left(\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)-\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m),\nabla f_{i_l}(\mathbf{y}_{l,i_l}^m)-\nabla F_{i_l}(\mathbf{y}_{l,i_l}^m)\right).\end{aligned}$$

Let $\mathbf{e}_k^m \coloneqq \nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)-\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)$. Unbiasedness of the stochastic gradients (Assumption 2.1) ensures $\mathbb{E}_m\left[\mathbf{e}_k^m\right] = \mathbb{E}_m\left[\mathbb{E}_{m,k}\left[\mathbf{e}_k^m\right]\right] = 0$. Now consider a cross-term with $k<l$:

$$\begin{aligned}\mathbb{C}_m\left(\mathbf{e}_k^m,\mathbf{e}_l^m\right) &= \mathbb{E}_m\left[\langle\mathbf{e}_k^m,\mathbf{e}_l^m\rangle\right]-\langle\mathbb{E}_m\left[\mathbf{e}_k^m\right],\mathbb{E}_m\left[\mathbf{e}_l^m\right]\rangle\\ &\overset{\text{(TP)}}{=} \mathbb{E}_m\left[\mathbb{E}_{m,l}\left[\langle\mathbf{e}_k^m,\mathbf{e}_l^m\rangle\right]\right]\\ &= \mathbb{E}_m\left[\langle\mathbf{e}_k^m,\mathbb{E}_{m,l}\left[\mathbf{e}_l^m\right]\rangle\right]\\ &= 0,\end{aligned}$$

where we used that $\mathbf{e}_k^m$ is $\mathcal{F}_l^m$-measurable, i.e., its randomness has been resolved by the beginning of iteration $l>k$.

With Assumption 2.2, we now find

$$\begin{aligned}\mathbb{E}_m\left[\left\|\boldsymbol{\nu}_m\right\|^2\right] &= \mathbb{V}_m\left(\boldsymbol{\nu}_m\right)\\ &= \sum_{k=0}^{K-1}\gamma_{i_k}^2\mathbb{C}_m\left(\mathbf{e}_k^m,\mathbf{e}_k^m\right)\\ &= \sum_{k=0}^{K-1}\gamma_{i_k}^2\mathbb{E}_m\left[\left\|\mathbf{e}_k^m\right\|^2\right]\\ &\overset{\text{(BV)}}{\leq} \sum_{k=0}^{K-1}\gamma_{i_k}^2\sigma^2\\ &= A\sigma^2.\end{aligned}$$

□

Lemma C.2. *Under Assumption 2.4, the cycle bias*

$$\mathbf{b}_m \coloneqq \sum_{k=0}^{K-1}\gamma_{i_k}\left(\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)-\nabla F_{i_k}(\mathbf{x}^m)\right)$$

satisfies

$$\|\mathbf{b}_m\|^2 \leq AL_{\max}^2 \sum_{k=0}^{K-1} \left\|\mathbf{y}_{k,i_k}^m - \mathbf{x}^m\right\|^2. \tag{5}$$

Proof. Using our definitions of the cycle bias $\mathbf{b}_m$ and cycle stepsize A, we find

$$\begin{aligned}
\|\mathbf{b}_m\|^2 &\overset{(B)}{=} \left\|\sum_{k=0}^{K-1} \gamma_{i_k} \left(\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m) - \nabla F_{i_k}(\mathbf{x}^m)\right)\right\|^2 \\
&\overset{(T)}{\leq} \left(\sum_{k=0}^{K-1} \gamma_{i_k} \left\|\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m) - \nabla F_{i_k}(\mathbf{x}^m)\right\|\right)^2 \\
&\overset{(CS)}{\leq} \left(\sum_{k=0}^{K-1} \gamma_{i_k}^2\right)\left(\sum_{k=0}^{K-1} \left\|\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m) - \nabla F_{i_k}(\mathbf{x}^m)\right\|^2\right) \\
&\overset{(A)}{=} A \sum_{k=0}^{K-1} \left\|\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m) - \nabla F_{i_k}(\mathbf{x}^m)\right\|^2 \\
&\overset{(LS)}{\leq} A \sum_{k=0}^{K-1} L_{i_k}^2 \left\|\mathbf{y}_{k,i_k}^m - \mathbf{x}^m\right\|^2 \\
&\leq AL_{\max}^2 \sum_{k=0}^{K-1} \left\|\mathbf{y}_{k,i_k}^m - \mathbf{x}^m\right\|^2,
\end{aligned}$$

where we used the smoothness of the local functions (Assumption 2.4) in the penultimate line. □

Lemma C.3. *Let $k' \in \{0, \ldots, K-1\}, m' \in \{m-1, m\}$ denote the iteration and cycle index corresponding to the last model seen by worker i_k at the beginning of iteration k in cycle m, i.e., $\mathbf{y}_{k,i_k}^m = \mathbf{x}_{k'}^{m'}$. If we denote the set of iterations between $\mathbf{y}_{k,i_k}^m$ and $\mathbf{x}^m$ by*

$$\mathcal{I}_k^m \coloneqq \begin{cases} \{k', \ldots, K-1\} \times \{m-1\}, & m' = m-1 \\ \{0, \ldots, k'-1\} \times \{m\}, & m' = m \end{cases},$$

then

$$\left\|\mathbf{y}_{k,i_k}^m - \mathbf{x}^m\right\|^2 \leq K \sum_{(l,c) \in \mathcal{I}_k^m} \gamma_{i_l}^2 \left\|\nabla f_{i_l}(\mathbf{y}_{l,i_l}^c)\right\|^2. \tag{6}$$

Proof. Note that $|\mathcal{I}_k^m| \leq K$ as the cyclic update schedule ensures that each worker receives a model update after at most K iterations. The claim then follows from the squared-sum inequality:

$$\begin{aligned}
\left\|\mathbf{y}_{k,i_k}^m - \mathbf{x}^m\right\|^2 &= \left\|\mathbf{x}_{k'}^{m'} - \mathbf{x}^m\right\|^2 \\
&= \left\|\sum_{(l,c) \in \mathcal{I}_k^m} \gamma_{i_l} \nabla f_{i_l}(\mathbf{y}_{l,i_l}^c)\right\|^2 \\
&\overset{(SS)}{\leq} |\mathcal{I}_k^m| \sum_{(l,c) \in \mathcal{I}_k^m} \gamma_{i_l}^2 \left\|\nabla f_{i_l}(\mathbf{y}_{l,i_l}^c)\right\|^2 \\
&\leq K \sum_{(l,c) \in \mathcal{I}_k^m} \gamma_{i_l}^2 \left\|\nabla f_{i_l}(\mathbf{y}_{l,i_l}^c)\right\|^2.
\end{aligned}$$

□

Lemma C.4. *Under assumptions 2.1 to 2.3 and 2.5,*

$$\mathbb{E}\left[\left\|\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)\right\|^2\right] \leq \sigma^2 + \zeta^2 + 2\rho^2\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right] + 2\rho^2 L^2\mathbb{E}\left[\left\|\mathbf{y}_{k,i_k}^m - \mathbf{x}^m\right\|^2\right]. \tag{7}$$

Proof. By assumptions 2.1 and 2.2,

$$\begin{aligned}\mathbb{E}_{m,k}\left[\left\|\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)\right\|^2\right] &\overset{\text{(UG)}}{=} \mathbb{E}_{m,k}\left[\left\|\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m) - \nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)\right\|^2\right] + \mathbb{E}_{m,k}\left[\left\|\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)\right\|^2\right] \\ &\overset{\text{(BV)}}{\leq} \sigma^2 + \left\|\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)\right\|^2.\end{aligned}$$

For the latter term, using assumptions 2.3 and 2.5, we derive the bound

$$\begin{aligned}\left\|\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)\right\|^2 &\overset{\text{(BH)}}{\leq} \zeta^2 + \rho^2\left\|\nabla F(\mathbf{y}_{k,i_k}^m)\right\|^2 \\ &\overset{\text{(SS)}}{\leq} \zeta^2 + 2\rho^2\left\|\nabla F(\mathbf{x}^m)\right\|^2 + 2\rho^2\left\|\nabla F(\mathbf{y}_{k,i_k}^m) - \nabla F(\mathbf{x}^m)\right\|^2 \\ &\overset{\text{(LS)}}{\leq} \zeta^2 + 2\rho^2\left\|\nabla F(\mathbf{x}^m)\right\|^2 + 2\rho^2 L^2\left\|\mathbf{y}_{k,i_k}^m - \mathbf{x}^m\right\|^2.\end{aligned}$$

Taking the unconditional expectation then gives

$$\mathbb{E}\left[\left\|\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)\right\|^2\right] \leq \sigma^2 + \zeta^2 + 2\rho^2\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right] + 2\rho^2 L^2\mathbb{E}\left[\left\|\mathbf{y}_{k,i_k}^m - \mathbf{x}^m\right\|^2\right].$$

□

Lemma C.5. *Let*

$$Q_M \coloneqq \sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\gamma_{i_k}^2\mathbb{E}\left[\left\|\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)\right\|^2\right]. \tag{8}$$

For $\gamma_{\max} \leq \frac{1}{2KL\rho}$*, we have*

$$Q_M \leq 2AM\left(\sigma^2 + \zeta^2\right) + 4A\rho^2\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right]. \tag{9}$$

Proof. Multiplying the bound derived in Lemma C.3 by $\gamma_{i_k}^2$ and summing over all cycles $m = 0, \ldots, M-1$ and iterations $k = 0, \ldots, K-1$, we find

$$\begin{aligned}\sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\gamma_{i_k}^2\left\|\mathbf{y}_{k,i_k}^m - \mathbf{x}^m\right\|^2 &\leq K\sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\gamma_{i_k}^2\sum_{(l,c)\in\mathcal{I}_k^m}\gamma_{i_l}^2\left\|\nabla f_{i_l}(\mathbf{y}_{l,i_l}^c)\right\|^2 \\ &= K\sum_{m'=0}^{M-1}\sum_{k'=0}^{K-1}\gamma_{i_{k'}}^2\left\|\nabla f_{i_{k'}}(\mathbf{y}_{k',i_{k'}}^{m'})\right\|^2 \times \left(\sum_{(l,c):(k',m')\in\mathcal{I}_l^c}\gamma_{i_l}^2\right).\end{aligned}$$

In the first line, we sum over the paths $\mathcal{I}_k^m$ connecting $\mathbf{y}_{k,i_k}^m$ and $\mathbf{x}^m$, which we bound by considering the norms of the gradients computed along these paths, $\nabla f_{i_l}(\mathbf{y}_{l,i_l}^c)$. To get to the second line, we instead consider the gradient in an iteration (k', m'), $\nabla f_{i_{k'}}(\mathbf{y}_{k',i_{k'}}^{m'})$, and sum up the contribution of the paths on which it appears. Since the sequence of workers sending updates is cyclic, we can bound this factor uniformly by

$$\sum_{(l,c):(k',m')\in\mathcal{I}_l^c}\gamma_{i_l}^2 \leq \sum_{i=1}^{n}\gamma_i^2 \leq \sum_{k=0}^{K-1}\gamma_{i_k}^2 = A.$$

Hence,

$$\sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\gamma_{i_k}^2\mathbb{E}\left[\left\|\mathbf{y}_{k,i_k}^m-\mathbf{x}^m\right\|^2\right]\leq AK\sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\gamma_{i_k}^2\mathbb{E}\left[\left\|\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)\right\|\right]=AKQ_M. \tag{10}$$

Now, using Lemma C.4, we have

$$\begin{aligned}
Q_M &= \sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\gamma_{i_k}^2\mathbb{E}\left[\left\|\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)\right\|^2\right]\\
&\overset{(7)}{\leq} \sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\gamma_{i_k}^2\left(\sigma^2+\zeta^2+2\rho^2\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right]+2L^2\rho^2\mathbb{E}\left[\left\|\mathbf{y}_{k,i_k}^m-\mathbf{x}^m\right\|^2\right]\right)\\
&= AM\left(\sigma^2+\zeta^2\right)+2A\rho^2\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right]+2L^2\rho^2\sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\gamma_{i_k}^2\mathbb{E}\left[\left\|\mathbf{y}_{k,i_k}^m-\mathbf{x}^m\right\|^2\right]\\
&\overset{(10)}{\leq} AM\left(\sigma^2+\zeta^2\right)+2A\rho^2\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right]+2AKL^2\rho^2Q_M\\
&\leq AM\left(\sigma^2+\zeta^2\right)+2A\rho^2\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right]+2\gamma_{\max}^2K^2L^2\rho^2Q_M,
\end{aligned}$$

where we use $A=\sum_{k=0}^{K-1}\gamma_{i_k}^2\leq\gamma_{\max}\sum_{k=0}^{K-1}\gamma_{i_k}\leq\gamma_{\max}^2K$ to obtain the last inequality.

Gathering Q_M-terms on the left side,

$$\left(1-2\gamma_{\max}^2K^2L^2\rho^2\right)Q_m\leq AM\left(\sigma^2+\zeta^2\right)+2A\rho^2\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right],$$

we see that, if $\gamma_{\max}\leq\frac{1}{2KL\rho}$, then

$$Q_M\leq 2AM\left(\sigma^2+\zeta^2\right)+4A\rho^2\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right].$$

□

Lemma C.6 (Bias)**.** *Under Assumptions 2.1 to 2.5, and if* $\gamma_{\max}\leq\frac{1}{2KL\rho}$*, then*

$$\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\mathbf{b}_m\right\|^2\right]\leq 2A^2K^2L_{\max}^2M\left(\sigma^2+\zeta^2\right)+4A^2K^2L_{\max}^2\rho^2\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right]. \tag{11}$$

Proof. Similar to the proof of Lemma C.5, we first bound

$$\begin{aligned}
\sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\left\|\mathbf{y}_{k,i_k}^m-\mathbf{x}^m\right\|^2 &\leq K\sum_{m=0}^{M-1}\sum_{k=0}^{K-1}1\times\sum_{(l,c)\in\mathcal{I}_k^m}\gamma_{i_l}^2\left\|\nabla f_{i_l}(\mathbf{y}_{l,i_l}^c)\right\|^2\\
&= K\sum_{m'=0}^{M-1}\sum_{k'=0}^{K-1}\gamma_{i_{k'}}^2\left\|\nabla f_{i_{k'}}(\mathbf{y}_{k',i_{k'}}^{m'})\right\|^2\times\underbrace{\left(\sum_{(l,c):(k',m')\in\mathcal{I}_l^c}1\right)}_{\leq K},
\end{aligned}$$

so

$$\sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\mathbf{y}_{k,i_k}^m-\mathbf{x}^m\right\|^2\right]\leq K^2Q_M. \tag{12}$$

Now

$$\sum_{m=0}^{M-1}\mathbb{E}\left[\|\mathbf{b}_m\|^2\right] \overset{(5)}{\leq} AL_{\max}^2\sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\mathbf{y}_{k,i_k}^m-\mathbf{x}^m\right\|^2\right] \overset{(12)}{\leq} AK^2L_{\max}^2Q_M.$$

If $\gamma_{\max}\leq\frac{1}{2KL\rho}$, we can apply Lemma C.5 to obtain the desired bound:

$$\sum_{m=0}^{M-1}\mathbb{E}\left[\|\mathbf{b}_m\|^2\right]\leq 2A^2K^2L_{\max}^2M\left(\sigma^2+\zeta^2\right)+4A^2K^2L_{\max}^2\rho^2\sum_{m=0}^{M-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}^m)\|^2\right].$$

□

Lemma C.7 (Cycle Step Decomposition)**.** *Let the worker-specific stepsizes be chosen such that $\gamma_i\propto w_i\tau_i$. Then,*

$$\mathbf{S}_m\coloneqq\sum_{k=0}^{K-1}\gamma_{i_k}\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m)=\alpha\nabla F(\mathbf{x}^m)+\mathbf{b}_m+\boldsymbol{\nu}_m, \tag{13}$$

where $\alpha\coloneqq\sum_{k=0}^{K-1}\gamma_{i_k}$ is the cycle stepsize.

Proof. Let $\gamma_i=cw_i\tau_i$ for some $c>0$. Then

$$\begin{aligned}\alpha&=\sum_{k=0}^{K-1}\gamma_{i_k}\\&=\sum_{i=1}^{n}\gamma_iK_i\\&=\sum_{i=1}^{n}cw_i\tau_i\cdot\frac{\tau_{\max}}{\tau_i}\\&=c\tau_{\max},\end{aligned}$$

and consequently

$$\begin{aligned}\sum_{k=0}^{K-1}\gamma_{i_k}\nabla F_{i_k}(\mathbf{x}^m)&=\sum_{i=1}^{n}\gamma_iK_i\nabla F_i(\mathbf{x}^m)\\&=\sum_{i=1}^{n}cw_i\tau_i\cdot\frac{\tau_{\max}}{\tau_i}\cdot\nabla F_i(\mathbf{x}^m)\\&=c\tau_{\max}\sum_{i=1}^{n}w_i\nabla F_i(\mathbf{x}^m)\\&=\alpha\nabla F(\mathbf{x}^m).\end{aligned} \tag{14}$$

For the cycle step, we now have

$$\begin{aligned}\mathbf{S_m} &= \sum_{k=0}^{K-1} \gamma_{i_k} \nabla f_{i_k}(\mathbf{y}_{k,i_k}^m) \\ &= \sum_{k=0}^{K-1} \gamma_{i_k} \nabla F_{i_k}(\mathbf{x}^m) \\ &\quad + \sum_{k=0}^{K-1} \gamma_{i_k} \left(\nabla F_{i_k}(\mathbf{y}_{k,i_k}^m) - \nabla F_{i_k}(\mathbf{x}^m)\right) \\ &\quad + \sum_{k=0}^{K-1} \gamma_{i_k} \left(\nabla f_{i_k}(\mathbf{y}_{k,i_k}^m) - \nabla F_{i_k}(\mathbf{y}_{k,i_k}^m)\right) \\ &\overset{(14),(B),(N)}{=} \alpha \nabla F(\mathbf{x}^m) + \mathbf{b}_m + \boldsymbol{\nu}_m.\end{aligned}$$

□

Lemma C.8. *Let the target objective be $F(\mathbf{x}) = \sum_{i=1}^n w_i F_i(\mathbf{x})$, and suppose Assumptions 2.1 to 2.3 and 3.1 hold. Assume the worker-specific stepsizes are chosen such that $\gamma_i \propto w_i \tau_i$. If the cycle stepsize $\alpha := \sum_{k=0}^{K-1} \gamma_{i_k}$ satisfies $0 < \alpha \leq \frac{1}{6L}$, then, after $M \geq 1$ cycles, the cycle iterates $\mathbf{x}^m$ generated by Algorithm 2 satisfy*

$$\frac{1}{M}\sum_{m=0}^{M-1} \mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right] \leq \frac{2\Delta}{\alpha M} + \frac{3AL\sigma^2}{\alpha} + \left(\frac{2}{\alpha^2} + \frac{3L}{\alpha}\right)\frac{1}{M}\sum_{m=0}^{M-1} \mathbb{E}\left[\left\|\mathbf{b}_m\right\|^2\right]. \tag{15}$$

Proof. In Lemma C.7, we defined the cycle step $\mathbf{S}_m$, so that for the cycle iterates of Algorithm 2 $\mathbf{S}_m = \mathbf{x}^m - \mathbf{x}^{m+1}$ holds.

By L-smoothness of the objective function F (Assumption 2.3), we obtain the standard bound (Nesterov, 2018)

$$\begin{aligned}F(\mathbf{x}^{m+1}) &\leq F(\mathbf{x}^m) + \left\langle \nabla F(\mathbf{x}^m), \mathbf{x}^{m+1} - \mathbf{x}_m \right\rangle + \frac{L}{2}\left\|\mathbf{x}^{m+1} - \mathbf{x}^m\right\|^2 \\ &= F(\mathbf{x}^m) - \left\langle \nabla F(\mathbf{x}^m), \mathbf{S_m} \right\rangle + \frac{L}{2}\left\|\mathbf{S_m}\right\|^2.\end{aligned}$$

Taking expectations conditional on information available at the beginning of cycle m, we find

$$\begin{aligned}\mathbb{E}_m\left[F(\mathbf{x}^{m+1})\right] &\leq F(\mathbf{x}^m) - \left\langle \nabla F(\mathbf{x}^m), \mathbb{E}_m\left[\mathbf{S_m}\right] \right\rangle + \frac{L}{2}\mathbb{E}_m\left[\left\|\mathbf{S_m}\right\|^2\right] \\ &\overset{(13)}{=} F(\mathbf{x}^m) - \left\langle \nabla F(\mathbf{x}^m), \alpha \nabla F(\mathbf{x}^m) + \mathbb{E}_m\left[\mathbf{b}_m\right] + \mathbb{E}_m\left[\boldsymbol{\nu}_m\right] \right\rangle + \frac{L}{2}\mathbb{E}_m\left[\left\|\mathbf{S_m}\right\|^2\right] \\ &= F(\mathbf{x}^m) - \alpha\left\|\nabla F(\mathbf{x}^m)\right\|^2 - \left\langle \nabla F(\mathbf{x}^m), \mathbb{E}_m\left[\mathbf{b}_m\right] \right\rangle + \frac{L}{2}\mathbb{E}_m\left[\left\|\mathbf{S_m}\right\|^2\right],\end{aligned} \tag{16}$$

where we used Lemma C.1 to drop $\mathbb{E}_m\left[\boldsymbol{\nu}_m\right] = 0$.

We bound the inner product term

$$\begin{aligned}-\left\langle \nabla F(\mathbf{x}^m), \mathbb{E}_m\left[\mathbf{b}_m\right] \right\rangle &\overset{(CS)}{\leq} \left\|\nabla F(\mathbf{x}^m)\right\| \left\|\mathbb{E}_m\left[\mathbf{b}_m\right]\right\| \\ &\overset{(YN)}{\leq} \frac{\alpha}{4}\left\|\nabla F(\mathbf{x}^m)\right\|^2 + \frac{1}{\alpha}\left\|\mathbb{E}_m\left[\mathbf{b}_m\right]\right\|^2 \\ &\overset{(JN)}{\leq} \frac{\alpha}{4}\left\|\nabla F(\mathbf{x}^m)\right\|^2 + \frac{1}{\alpha}\mathbb{E}_m\left[\left\|\mathbf{b}_m\right\|^2\right].\end{aligned} \tag{17}$$

Now,

$$\begin{aligned}\left\|\mathbf{S_m}\right\|^2 &\overset{(13)}{=} \left\|\alpha \nabla F(\mathbf{x}^m) + \mathbf{b}_m + \boldsymbol{\nu}_m\right\|^2 \\ &\overset{(SS)}{\leq} 3\alpha^2\left\|\nabla F(\mathbf{x}^m)\right\|^2 + 3\left\|\mathbf{b}_m\right\|^2 + 3\left\|\boldsymbol{\nu}_m\right\|^2,\end{aligned}$$

and thus

$$\mathbb{E}_m\left[\|\mathbf{S_m}\|^2\right] \leq 3\alpha^2\|\nabla F(\mathbf{x}^m)\|^2 + 3\mathbb{E}_m\left[\|\mathbf{b}_m\|^2\right] + 3A\sigma^2, \tag{18}$$

utilizing the second part of Lemma C.1.

Using (17) and (18) to further bound (16), we obtain

$$\begin{aligned}
\mathbb{E}_m\left[F(\mathbf{x}^{m+1})\right] &\overset{(16)}{\leq} F(\mathbf{x}^m) - \alpha\|\nabla F(\mathbf{x}^m)\|^2 - \langle\nabla F(\mathbf{x}^m), \mathbb{E}_m[\mathbf{b}_m]\rangle + \frac{L}{2}\mathbb{E}_m\left[\|\mathbf{S_m}\|^2\right] \\
&\overset{(17),(18)}{\leq} F(\mathbf{x}^m) - \alpha\|\nabla F(\mathbf{x}^m)\|^2 \\
&\quad + \frac{\alpha}{4}\|\nabla F(\mathbf{x}^m)\|^2 + \frac{1}{\alpha}\|\mathbb{E}_m[\mathbf{b}_m]\|^2 \\
&\quad + \frac{L}{2}\left(3\alpha^2\|\nabla F(\mathbf{x}^m)\|^2 + 3\mathbb{E}_m\left[\|\mathbf{b}_m\|^2\right] + 3A\sigma^2\right) \\
&\overset{(JN)}{\leq} F(\mathbf{x}^m) - \left(\frac{3}{4}\alpha - \frac{3L}{2}\alpha^2\right)\|\nabla F(\mathbf{x}^m)\|^2 \\
&\quad + \frac{3AL\sigma^2}{2} + \left(\frac{1}{\alpha} + \frac{3L}{2}\right)\mathbb{E}_m\left[\|\mathbf{b}_m\|^2\right].
\end{aligned}$$

For $\alpha \leq \frac{1}{6L}$, we have $\left(\frac{3}{4}\alpha - \frac{3L}{2}\alpha^2\right) \geq \frac{\alpha}{2}$ and thus

$$\mathbb{E}_m\left[F(\mathbf{x}^{m+1})\right] \leq F(\mathbf{x}^m) - \frac{\alpha}{2}\|\nabla F(\mathbf{x}^m)\|^2 + \frac{3AL\sigma^2}{2} + \left(\frac{1}{\alpha} + \frac{3L}{2}\right)\mathbb{E}_m\left[\|\mathbf{b}_m\|^2\right].$$

Rearranging gives

$$\|\nabla F(\mathbf{x}^m)\|^2 \leq \frac{2\left(F(\mathbf{x}^m) - \mathbb{E}_m\left[F(\mathbf{x}^{m+1})\right]\right)}{\alpha} + \frac{3AL\sigma^2}{\alpha} + \left(\frac{2}{\alpha^2} + \frac{3L}{\alpha}\right)\mathbb{E}_m\left[\|\mathbf{b}_m\|^2\right].$$

Taking unconditional expectation, averaging, and applying the bound $F(\mathbf{x}^0) - \mathbb{E}\left[F(\mathbf{x}^M)\right] \geq F(\mathbf{x}^0) - F^* = \Delta$ (Assumption 2.3) gives

$$\begin{aligned}
\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}^m)\|^2\right] &\leq \frac{2\sum_{m=0}^{M-1}\mathbb{E}\left[F(\mathbf{x}^m) - F(\mathbf{x}^{m+1})\right]}{\alpha M} \\
&\quad + \frac{3AL\sigma^2}{\alpha} + \left(\frac{2}{\alpha^2} + \frac{3L}{\alpha}\right)\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\mathbb{E}_m\left[\|\mathbf{b}_m\|^2\right]\right] \\
&\leq \frac{2\Delta}{\alpha M} + \frac{3AL\sigma^2}{\alpha} + \left(\frac{2}{\alpha^2} + \frac{3L}{\alpha}\right)\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\|\mathbf{b}_m\|^2\right].
\end{aligned}$$

□

Theorem C.9 (Convergence to a Weighted Average)**.** *Let the target objective be $F(\mathbf{x}) = \sum_{i=1}^n w_i F_i(\mathbf{x})$, and suppose Assumptions 2.1 to 2.5 and 3.1 hold. Assume the worker-specific stepsizes are chosen such that $\gamma_i \propto w_i\tau_i$. If the maximum stepsize satisfies $0 < \gamma_{\max} \leq \frac{1}{5KL_{\max\rho}}$ and the cycle stepsize $\alpha := \sum_{k=0}^{K-1}\gamma_{i_k}$ satisfies $0 < \alpha \leq \frac{1}{6L}$, then, after $M \geq 1$ cycles, the cycle iterates $\mathbf{x}^m$ generated by Algorithm 2 satisfy*

$$\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}^m)\|^2\right] \leq c_0\cdot\frac{\Delta}{\alpha M} + c_1\cdot\frac{A}{\alpha}L\sigma^2 + c_2\cdot\left(\frac{A}{\alpha}\right)^2 K^2L_{\max}^2\left(\sigma^2+\zeta^2\right),$$

for some constants $c_0, c_1, c_2 > 0$, where $A := \sum_{k=0}^{K-1}\gamma_{i_k}^2$.

Proof. Plug the bound on the cycle bias from Lemma C.6 into Lemma C.8 to obtain

$$\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right] \leq \frac{2\Delta}{\alpha M} + \frac{3AL\sigma^2}{\alpha} + \left(\frac{2}{\alpha^2}+\frac{3L}{\alpha}\right)2A^2K^2L_{\max}^2\left(\sigma^2+\zeta^2\right) + \left(\frac{2}{\alpha^2}+\frac{3L}{\alpha}\right)4A^2K^2L_{\max}^2\rho^2\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right].$$

Now, if $\alpha \leq \frac{1}{6L}$, then also $\left(\frac{2}{\alpha^2}+\frac{3L}{\alpha}\right) \leq \frac{2.5}{\alpha^2}$. Grouping terms involving the gradient norm on the left gives

$$\left(1-\frac{10A^2K^2L_{\max}^2\rho^2}{\alpha^2}\right)\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right] \leq \frac{2\Delta}{\alpha M}+\frac{3AL\sigma^2}{\alpha}+\frac{5A^2K^2L_{\max}^2\left(\sigma^2+\zeta^2\right)}{\alpha^2}$$

We require the first factor on the left-hand side to exceed $1/2$, or, equivalently, $\frac{10A^2K^2L_{\max}^2\rho^2}{\alpha^2} \leq \frac{1}{2}$. Noting that $A = \sum_{k=0}^{K-1}\gamma_{i_k}^2 \leq \gamma_{\max}\sum_{k=0}^{K-1}\gamma_{i_k} = \gamma_{\max}\alpha$, this requirement is met if $10\gamma_{\max}^2K^2L_{\max}^2\rho^2 \leq \frac{1}{2}$, or, if $\gamma_{\max} \leq \frac{1}{5KL_{\max}\rho} \leq \frac{1}{\sqrt{20}KL_{\max}\rho}$.

Thus, for $\gamma_{\max} \leq \frac{1}{5KL_{\max}\rho}$, we find

$$\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right] \leq 4\frac{\Delta}{\alpha M}+6\frac{A}{\alpha}L\sigma^2+10\frac{A^2}{\alpha^2}K^2L_{\max}^2\left(\sigma^2+\zeta^2\right). \tag{19}$$

□

C.3. Proof of Theorem 4.1 and Corollary 4.2

Theorem 4.1 (Convergence to the Equal-Weighted Average)**.** *Let the worker-specific stepsizes be chosen according to (2) and suppose Assumptions 2.1 to 2.5 and 3.1 hold. If the stepsize parameter satisfies* $0 < \gamma \leq \min\left\{\frac{1}{6L\tau_H}, \frac{1}{5L_{\max}\rho\tau_{\max}}\right\}$*, then, after* $M \geq 1$ *cycles, the cycle iterates* $\mathbf{x}^m$ *generated by Algorithm 2 satisfy*

$$\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right] \leq c_0\cdot\frac{\Delta}{\gamma\tau_H M}+c_1\cdot\frac{\gamma\tau_A L\sigma^2}{K} + c_2\cdot\gamma^2\tau_A^2L_{\max}^2\left(\sigma^2+\zeta^2\right)$$

for some absolute constants $c_0, c_1, c_2 > 0$*.*

Proof. This is a special case of Theorem C.9 with stepsizes chosen according to (2).

The cycle stepsize now becomes

$$\alpha = \sum_{k=0}^{K-1}\gamma_{i_k} = \sum_{i=1}^{n}\gamma_i\cdot K_i = \sum_{i=1}^{n}\gamma\tau_i\frac{\tau_H}{n\tau_{\max}}\cdot\frac{\tau_{\max}}{\tau_i} = \gamma\tau_H,$$

the sum of squared stepsizes

$$A = \sum_{k=0}^{K-1}\gamma_{i_k}^2 = \sum_{i=1}^{n}\gamma_i^2\cdot K_i = \sum_{i=1}^{n}\gamma^2\tau_i^2\frac{\tau_H^2}{n^2\tau_{\max}^2}\cdot\frac{\tau_{\max}}{\tau_i} = \gamma^2\frac{\tau_H^2\tau_A}{n\tau_{\max}} = \gamma^2\frac{\tau_H\tau_A}{K},$$

and the maximum stepsize

$$\gamma_{\max} = \gamma\tau_{\max}\frac{\tau_H}{n\tau_{\max}} = \gamma\frac{\tau_H}{n} = \gamma\frac{\tau_{\max}}{K},$$

where we used the identity $K/n = \tau_{\max}/\tau_H$ (Assumption 3.1).

Therefore, $\frac{A}{\alpha} = \frac{\gamma\tau_A}{K}$. Plugging this into (19) gives

$$\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right] \leq 4\frac{\Delta}{\alpha M} + 6\frac{\gamma\tau_A L\sigma^2}{K} + 10\gamma^2\tau_A^2 L_{\max}^2\left(\sigma^2+\zeta^2\right).$$

The bound on the cycle stepsize translates to

$$\alpha \leq \frac{1}{6L} \Leftrightarrow \gamma \leq \frac{1}{6L\tau_H},$$

and that on the maximum stepsize to

$$\gamma_{\max} \leq \frac{1}{5KL_{\max}\rho} \Leftrightarrow \gamma \leq \frac{1}{5L_{\max}\rho\tau_{\max}}.$$

□

Corollary 4.2 (Time Complexity for the Equal-Weighted Average)**.** *In the setting of Theorem 4.1, the worst-case wall-clock time complexity to find an ε-stationary point of F is*

$$\mathcal{O}\left(\frac{\Delta L\sigma^2}{n\varepsilon^2}\tau_A + \frac{\Delta L_{\max}\sqrt{\sigma^2+\zeta^2}}{\varepsilon^{1.5}}\frac{\tau_{\max}}{\tau_H}\tau_A + \frac{\Delta L_{\max}\rho}{\varepsilon}\frac{\tau_{\max}}{\tau_H}\tau_{\max} + \frac{\Delta L}{\varepsilon}\tau_{\max}\right).$$

Proof. Under the assumptions of Theorem 4.1, we have

$$\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}^m)\right\|^2\right] \leq \underbrace{c_0\cdot\frac{\Delta}{\gamma\tau_H M}}_{T_0} + \underbrace{c_1\cdot\frac{\gamma\tau_A L\sigma^2}{K}}_{T_1} + \underbrace{c_2\cdot\gamma^2\tau_A^2 L_{\max}^2\left(\sigma^2+\zeta^2\right)}_{T_2}.$$

To achieve $T_1 \leq \frac{\varepsilon}{4}$, we require $\gamma \leq \frac{1}{4c_1}\frac{\varepsilon K}{L\sigma^2\tau_A}$. To achieve $T_2 \leq \frac{\varepsilon}{4}$, we likewise require $\gamma \leq \frac{1}{\sqrt{4c_2}}\frac{\sqrt{\varepsilon}}{L_{\max}\tau_A\sqrt{\sigma^2+\zeta^2}}$.

The largest γ satisfying these constraints, as well as those in Theorem 4.1, is

$$\hat{\gamma} \coloneqq \min\left\{\frac{1}{4c_1}\frac{\varepsilon K}{L\sigma^2\tau_A}, \frac{1}{\sqrt{4c_2}}\frac{\sqrt{\varepsilon}}{L_{\max}\tau_A\sqrt{\sigma^2+\zeta^2}}, \frac{1}{5L_{\max}\rho\tau_{\max}}, \frac{1}{6L\tau_H}\right\}.$$

To achieve $T_0 \leq \frac{\varepsilon}{2}$ and bound the right-hand side of Theorem 4.1 by ε, the cycle count must satisfy $M \geq \frac{2c_0\Delta}{\varepsilon\hat{\gamma}\tau_H}$. Ignoring constant factors, this cycle bound can be expressed as

$$M \gtrsim \frac{\Delta L\sigma^2}{\varepsilon^2 K\tau_H}\tau_A + \frac{\Delta L_{\max}\sqrt{\sigma^2+\zeta^2}}{\varepsilon^{1.5}\tau_H}\tau_A + \frac{\Delta L_{\max}\rho}{\varepsilon\tau_H}\tau_{\max} + \frac{\Delta L}{\varepsilon}.$$

To obtain the wall-clock time complexity, we multiply the required number of cycles by the wall-clock duration of a single cycle. Under Assumption 3.1, a cycle is completed after $\tau_{\max} = \frac{K\tau_H}{n}$ time units. Thus, after

$$\mathcal{O}\left(\frac{\Delta L\sigma^2}{n\varepsilon^2}\tau_A + \frac{\Delta L_{\max}\sqrt{\sigma^2+\zeta^2}}{\varepsilon^{1.5}}\frac{\tau_{\max}}{\tau_H}\tau_A + \frac{\Delta L_{\max}\rho}{\varepsilon}\frac{\tau_{\max}}{\tau_H}\tau_{\max} + \frac{\Delta L}{\varepsilon}\tau_{\max}\right)$$

time units, we achieve the target accuracy. □

C.4. Proof of Theorem 4.3 and Corollary 4.4

Theorem 4.3 (Convergence to the Frequency-Weighted Average)**.** *Let the workers' stepsizes be equal, $\gamma_i = \gamma/K$, and suppose Assumptions 2.1 to 2.5 and 3.1 hold for $\tilde{F}$.*[2] *If the stepsize parameter satisfies $0 < \gamma \leq \min\left\{\frac{1}{6\tilde{L}}, \frac{1}{5L_{\max}\tilde{\rho}}\right\}$, then, after $M \geq 1$ cycles, the cycle iterates $\mathbf{x}^m$ generated by Algorithm 2 satisfy*

$$\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla\tilde{F}(\mathbf{x}^m)\right\|^2\right] \leq c_0\cdot\frac{\tilde{\Delta}}{\gamma M} + c_1\cdot\frac{\gamma\tilde{L}\sigma^2}{K} + c_2\cdot\gamma^2 L_{\max}^2\left(\sigma^2+\tilde{\zeta}^2\right)$$

for some absolute constants $c_0, c_1, c_2 > 0$.

Proof. This is a special case of Theorem C.9 with equal stepsizes $\gamma_i = \gamma/K$.

The cycle stepsize now becomes

$$\alpha = \sum_{k=0}^{K-1}\gamma_{i_k} = K\cdot\frac{\gamma}{K} = \gamma,$$

the sum of squared stepsizes

$$A = \sum_{k=0}^{K-1}\gamma_{i_k}^2 = K\cdot\left(\frac{\gamma}{K}\right)^2 = \frac{\gamma^2}{K},$$

and the maximum stepsize

$$\gamma_{\max} = \frac{\gamma}{K}.$$

Therefore, $\frac{A}{\alpha} = \frac{\gamma}{K}$. Plugging this into (19) gives

$$\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla\tilde{F}(\mathbf{x}^m)\right\|^2\right] \leq 4\frac{\tilde{\Delta}}{\gamma M} + 6\frac{\gamma\tilde{L}\sigma^2}{K} + 10\gamma^2 L_{\max}^2\left(\sigma^2+\tilde{\zeta}^2\right).$$

The bound on the cycle stepsize translates to

$$\alpha \leq \frac{1}{6\tilde{L}} \Leftrightarrow \gamma \leq \frac{1}{6\tilde{L}},$$

and that on the maximum stepsize

$$\gamma_{\max} \leq \frac{1}{5KL_{\max}\tilde{\rho}} \Leftrightarrow \gamma \leq \frac{1}{5L_{\max}\tilde{\rho}}.$$

□

Corollary 4.4 (Time Complexity for the Frequency-Weighted Average)**.** *In the setting of Theorem 4.3, the worst-case wall-clock time complexity to find an ε-stationary point of $\tilde{F}$ is*

$$\mathcal{O}\left(\frac{\tilde{\Delta}\tilde{L}\sigma^2}{n\varepsilon^2}\tau_H + \frac{\tilde{\Delta}L_{\max}\sqrt{\sigma^2+\tilde{\zeta}^2}}{\varepsilon^{1.5}}\tau_{\max} + \frac{\tilde{\Delta}L_{\max}\tilde{\rho}}{\varepsilon}\tau_{\max} + \frac{\tilde{\Delta}\tilde{L}}{\varepsilon}\tau_{\max}\right).$$

[2] That is, we replace $F, \Delta, L, \zeta, \rho$ by $\tilde{F}, \tilde{\Delta}, \tilde{L}, \tilde{\zeta}, \tilde{\rho}$ in Assumptions 2.3 and 2.5.

Proof. Under the assumptions of Theorem 4.3, we have

$$\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}\left[\left\|\nabla\tilde{F}(\mathbf{x}^m)\right\|^2\right] \leq \underbrace{c_0\cdot\frac{\tilde{\Delta}}{\gamma M}}_{T_0} + \underbrace{c_1\cdot\frac{\gamma^2\tilde{L}\sigma^2}{K}}_{T_1} + \underbrace{c_2\cdot\gamma^2 L_{\max}^2\left(\sigma^2+\tilde{\zeta}^2\right)}_{T_2}.$$

To achieve $T_1 \leq \frac{\varepsilon}{4}$, we require $\gamma \leq \frac{1}{4c_1}\frac{\varepsilon K}{\tilde{L}\sigma^2}$. To achieve $T_2 \leq \frac{\varepsilon}{4}$, we likewise require $\gamma \leq \frac{1}{\sqrt{4c_2}}\frac{\sqrt{\varepsilon}}{L_{\max}\sqrt{\sigma^2+\tilde{\zeta}^2}}$.

The largest γ satisfying these constraints, as well as those in Theorem 4.3, is

$$\hat{\gamma} \coloneqq \min\left\{\frac{1}{4c_1}\frac{K\varepsilon}{\tilde{L}\sigma^2}, \frac{1}{\sqrt{4c_2}}\frac{\sqrt{\varepsilon}}{L_{\max}\sqrt{\sigma^2+\tilde{\zeta}^2}}, \frac{1}{5L_{\max}\tilde{\rho}}, \frac{1}{6\tilde{L}}\right\}.$$

To achieve $T_0 \leq \frac{\varepsilon}{2}$ and bound the right-hand side of Theorem 4.3 by ε, the cycle count must satisfy $M \geq \frac{2c_0\tilde{\Delta}}{\varepsilon\hat{\gamma}}$. Ignoring constant factors, this cycle bound can be expressed as

$$M \gtrsim \frac{\tilde{\Delta}\tilde{L}\sigma^2}{\varepsilon^2 K} + \frac{\tilde{\Delta}L_{\max}\sqrt{\sigma^2+\tilde{\zeta}^2}}{\varepsilon^{1.5}} + \frac{\tilde{\Delta}L_{\max}\tilde{\rho}}{\varepsilon} + \frac{\tilde{\Delta}\tilde{L}}{\varepsilon}.$$

To obtain the wall-clock time complexity, we multiply the required number of cycles by the wall-clock duration of a single cycle. Under Assumption 3.1, a cycle is completed after $\tau_{\max} = \frac{K\tau_H}{n}$ time units. Thus, after

$$\mathcal{O}\left(\frac{\tilde{\Delta}\tilde{L}\sigma^2}{n\varepsilon^2}\tau_H + \frac{\tilde{\Delta}L_{\max}\sqrt{\sigma^2+\tilde{\zeta}^2}}{\varepsilon^{1.5}}\tau_{\max} + \frac{\tilde{\Delta}L_{\max}\tilde{\rho}}{\varepsilon}\tau_{\max} + \frac{\tilde{\Delta}\tilde{L}}{\varepsilon}\tau_{\max}\right)$$

time units, we achieve the target accuracy. □