LOCAL MIXVR: BREAKING THE COMMUNICATION-SAMPLE DEPENDENCE IN DISTRIBUTED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Communication overhead is a crucial bottleneck in scalable distributed learning. While existing methods aim to efficiently utilize data points, such as Local SGD, Minibatch SGD, and their accelerated variants, they still exhibit communication-round complexity that scales with the total number of samples N. In this paper, we introduce Local MixVR, a distributed framework that integrates local updates with variance-reduction techniques to mitigate local noise. We show that Local MixVR is the first distributed method to eliminate the dependence of communication complexity on N, achieving a complexity that scales only with the number of workers M. In common regimes where $M \leq \mathcal{O}\left(N^{1/4}\right)$, Local MixVR outperforms the state-of-the-art Minibatch Accelerated SGD baseline, bridging a long-standing gap in distributed optimization and establishing a new paradigm for communication-efficient training.

1 INTRODUCTION

In the era of massive-scale machine learning (Verbraeken et al., 2020; Zhao et al., 2023), parallelizing training across multiple workers has become necessary. However, communication overhead remains a fundamental bottleneck: as the number of workers increases, the cost of synchronizing updates can dominate overall training time. This can be even more severe in poorly connected clusters, where limited bandwidth and high latency between nodes can significantly slow synchronization (Douillard et al., 2023; Jaghouar et al., 2024). As a result, a central goal in distributed learning is to minimize the number of communication rounds while maintaining fast convergence.

A fundamental approach to parallel stochastic optimization is Minibatch SGD (Dekel et al., 2012), where M workers compute stochastic gradients on minibatches of size K and average them at each synchronization round to perform a global update. This is equivalent to using an effective batch size of MK, which reduces the stochastic variance by a factor of MK. Due to its simplicity and effectiveness, Minibatch SGD is widely used in distributed training (Brown et al., 2020). However, increasing the effective batch size is beneficial only up to a point (Kaplan et al., 2020). Once stochastic noise has been sufficiently reduced, further increases in batch size yield diminishing returns. In many modern training regimes, the total batch size MK enabled by many workers already exceeds the level needed for efficient learning. Consequently, adding more workers may not substantially reduce the number of iterations to convergence, while it still increases synchronization and communication costs. This can slow overall training despite the availability of additional compute.

This motivates distributed methods to use the available gradient information more effectively, rather than simply increasing the minibatch size. Instead of using the extra gradients solely to reduce variance, one can also use them to make additional progress in optimization. This is the idea behind Local SGD (Stich, 2019; Mangasarian & Solodov, 1993): each worker performs K local steps, then the server averages the models and sends the averaged model back to all workers. This contrasts with Minibatch SGD, which performs only one global update per communication round. More specifically, over R rounds, Minibatch SGD performs R optimization steps, whereas Local SGD performs KR local updates per worker. Thus, for the same number of processed gradients, Local SGD leverages this information to reduce gradient noise and make more progress along the optimization path.

This approach, however, introduces a new challenge: *worker drift*. Because each worker performs multiple updates with different samples, local models can diverge between synchronization rounds,

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.

Table 1: Convergence rates and $R_{\min}$ for $\sigma = \mathcal{O}(1)$ and $N = MKR$ total samples.

Method	**Rate** $[\mathcal{O}(\cdot)]$	$\boldsymbol{R_{\min}}$ $[\Omega(\cdot)]$
MiniBatch SGD (Dekel et al., 2012)	$\frac{1}{R} + \frac{\sigma}{\sqrt{MKR}}$	$N^{1/2}$
MiniBatch Accelerated SGD (Lan, 2012; Dekel et al., 2012)	$\frac{1}{R^2} + \frac{\sigma}{\sqrt{MKR}}$	$N^{1/4}$
Local SGD (Khaled et al., 2020)	$\frac{1}{KR} + \frac{\sigma^{2/3}}{K^{1/3}R^{2/3}} + \frac{\sigma}{\sqrt{MKR}}$	$MN^{1/2}$
SCAFFOLD (Karimireddy et al., 2020b)	$\frac{1}{R} + \frac{\sigma}{\sqrt{MKR}}$	$N^{1/2}$
Local Momentum (Dahan & Levy, 2024)	$\frac{1}{KR} + \frac{\sigma^{1/2}}{K^{1/4}R} + \frac{1}{K^{1/3}R^{4/3}} + \frac{1}{R^2} + \frac{\sigma}{\sqrt{MKR}}$	$M^{1/3}N^{1/3}$
Accelerated Outer SGD (Khaled et al., 2025)	$\frac{1}{KR^2} + \frac{\sigma^{2/3}}{M^{1/3}K^{1/3}R} + \frac{\sigma^{1/2}}{K^{1/4}R^{5/4}} + \frac{\sigma}{\sqrt{MKR}}$	$M^{1/4}N^{1/4}$
Local SGD lower bound (Glasgow et al., 2022)	$\frac{1}{KR} + \frac{\sigma^{2/3}}{K^{1/3}R^{2/3}} + \frac{\sigma}{\sqrt{MKR}}$	$MN^{1/2}$
Local MixVR (ours) Theorem 2.2	$\frac{1}{KR} + \frac{\sigma}{K^{\mathbf{1/2}}R} + \frac{\sigma}{\sqrt{MKR}}$	**M**

especially when many local steps are taken. Thus, although Local SGD seeks to extract more progress from each sample, its gains are limited by this accumulated drift. Indeed, lower bounds (Glasgow et al., 2022) show that Local SGD cannot require fewer communication rounds than Minibatch SGD to effectively utilize data points; see Table 1.

To assess the required number of communication rounds, one should ask, given a sample budget N, what is the minimum number of communication rounds such that additional optimization steps no longer improve the convergence rate? The key trade-off is between the *optimization error* and the *statistical error*. Since the statistical error is determined by the fixed sample budget N, there is a point beyond which further reducing the optimization error (by performing more optimization steps) does not improve the overall rate. For example, Minibatch SGD has a convergence rate of order

$$\underbrace{\frac{1}{R}}_{\text{optimization error}} + \underbrace{\frac{\sigma}{\sqrt{N}}}_{\text{statistical error}}$$

where $N := MKR$ and σ^2 bounds the stochastic noise. Thus, the statistical error dominates as long as

$$\frac{1}{R} \leq \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$$

Consequently, once the number of communication rounds exceeds the critical threshold $R_{\min} = \Omega\left(\sqrt{N}\right)$, increasing R no longer improves convergence. Conversely, using fewer than $R_{\min}$ rounds worsens the rate, since the optimization error remains larger than the statistical error, which is the lower bound for first-order methods using N samples (Nemirovskij & Yudin, 1983; Agarwal et al., 2009). Thus, $R_{\min}$ is the minimum number of communication rounds required to retain the optimal convergence rate. Although existing local methods aim to reduce this threshold (Table 1), none provably outperform Minibatch Accelerated SGD (ASGD) (Lan, 2012). In fact, existing lower bounds for smooth convex optimization suggest that improving upon Minibatch ASGD is impossible (Woodworth et al., 2021). However, as Woodworth et al. (2021) note, the oracle in their lower-bound construction cannot be written as the gradient of a random smooth function and therefore does not capture the stochastic model in Equation (3). They leave open whether this structure can be exploited to obtain better convergence guarantees or whether the same lower bound can be proved using an oracle that satisfies this constraint. Our work takes a step toward addressing this question by introducing a local method whose upper bounds *surpass Minibatch ASGD in common regimes* under this stochastic model. In particular, the communication complexity of existing methods, including Minibatch ASGD, *grows with the sample size* N. This dependence can be restrictive in common distributed settings, where datasets are often large, and communication is a primary bottleneck.

In this work, we introduce Local MixVR, a distributed framework that integrates three distinct variance-reduction techniques:

- **Local double-momentum:** We adopt a recent double-momentum approach (Dahan & Levy, 2025) to substantially reduce local stochastic variance and keep workers closely aligned.
- **Hybrid local-minibatching:** Near the end of each communication round, we replace the remaining local steps with a single minibatch update of the same size to reduce worker divergence rather than amplify it by progressing locally.
- **Drift correction mechanism:** We introduce a correction mechanism that mitigates bias arising from the transition from local to global parameters at synchronization steps.

Our Contributions:

- **Surpassing the Accelerated Baseline.** We show that in the common regimes where $M \leq \mathcal{O}\left(N^{1/4}\right)$, Local MixVR is the first algorithm to improve upon the state-of-the-art Minibatch ASGD baseline (Table 1). This regime arises naturally in practical training; for example, ImageNet-1K (Deng et al., 2009) has about 1.28×10^6 training images and is often trained with $M = 8$ workers (Goyal et al., 2017), giving $R_{\min}^{\mathrm{ASGD}}/R_{\min}^{\mathrm{MixVR}} \approx N^{1/4}/M \approx 4.2$. Similarly, FineWeb (Penedo et al., 2024) has roughly 15×10^{12} tokens and was trained with $M = 64$ workers, yielding $R_{\min}^{\mathrm{ASGD}}/R_{\min}^{\mathrm{MixVR}} \approx 30.7$. Thus, Local MixVR can use fewer communication rounds than Minibatch ASGD while maintaining optimal performance.
- **Sample-Independent Communication Complexity.** We show that Local MixVR is the first algorithm whose required number of communication rounds is independent of the sample size N and depends only on the number of workers M, thereby yielding a stronger guarantee for settings with massive datasets.

Related Work. Local-update methods, including Local SGD and its variants (Koloskova et al., 2020; Karimireddy et al., 2020a; Khaled et al., 2020; Stich, 2019; Gorbunov et al., 2021; Mishchenko et al., 2022b;a; Patel et al., 2022; Cheng et al., 2023; Dahan & Levy, 2024; Yuan & Ma, 2020; Reddi et al., 2021; Mitra et al., 2021; Zaccone et al., 2023), reduce communication by performing multiple local steps between synchronization rounds. Existing work has primarily focused on matching, and, in some regimes, improving upon, the Minibatch SGD rate (see Table 1), while whether local computation can outperform Minibatch ASGD remains largely open (Nesterov, 2013; Lan, 2012). Our work takes a step toward addressing this question by establishing improvements over Minibatch ASGD in common regimes. A central obstacle to achieving such improvements is worker drift, which existing methods address through several complementary strategies. SCAFFOLD (Karimireddy et al., 2020b) and Mime (Karimireddy et al., 2020a) use synchronized control variates to anchor local trajectories, whereas momentum-based methods (Cheng et al., 2023; Dahan & Levy, 2024; Yuan & Ma, 2020; Zaccone et al., 2023) share momentum information across workers. By incorporating global information from previous synchronization rounds, these mechanisms reduce local stochasticity and keep worker trajectories aligned. A complementary strategy reduces gradient variance through minibatching (Dekel et al., 2012). For example, Post-Local SGD (Lin et al., 2018) first runs Minibatch SGD and then switches to local updates, while Patel et al. (2022) combine minibatches of size $\mathcal{O}(K)$ with $\mathcal{O}(K)$ local steps. More advanced methods use momentum-based variance reduction, such as STORM (Cutkosky & Orabona, 2019), which corrects the momentum estimate to track changes in the stochastic gradient and reduce its variance. Incorporating this mechanism into local updates can improve stability and convergence (Karimireddy et al., 2020a; Patel et al., 2022; Cheng et al., 2023).

Settings. We consider stochastic convex optimization problems of the form

$$f(\mathbf{x}) := \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}\left[f(\mathbf{x};\mathbf{z})\right], \tag{1}$$

where $\mathcal{D}$ is an unknown distribution. We study a distributed setting in which M workers jointly minimize f over R communication rounds, each performing K local steps using independent samples. The algorithm outputs $\mathbf{x}_{\mathrm{out}} \in \mathbb{R}^d$, and its performance is measured by the expected excess loss:

$$\mathbb{E}\left[f(\mathbf{x}_{\mathrm{out}})\right] - \min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}).$$

At each local iteration t, worker i samples $\mathbf{z}_t^{(i)} \sim \mathcal{D}$. The worker then computes a stochastic gradient $\nabla f\left(\mathbf{x}_t^{(i)};\mathbf{z}_t^{(i)}\right)$ that serves as an unbiased estimator, i.e., $\mathbb{E}\left[\nabla f\left(\mathbf{x}_t^{(i)};\mathbf{z}_t^{(i)}\right) \mid \mathbf{x}_t^{(i)}\right] = \nabla f\left(\mathbf{x}_t^{(i)}\right)$.

Algorithm 1 μ^2-SGD

Input: $\bar{\mathbf{x}}_t, \bar{\mathbf{x}}_{t-1}, \mathbf{x}_t, \mathbf{d}_{t-1}$, stepsize η_t, momentum parameters β_t, γ_t
Sample $\mathbf{z}_t \sim \mathcal{D}$, and compute $\mathbf{g}_t \leftarrow \nabla f(\bar{\mathbf{x}}_t; \mathbf{z}_t)$, $\tilde{\mathbf{g}}_{t-1} \leftarrow \nabla f(\bar{\mathbf{x}}_{t-1}; \mathbf{z}_t)$
$\mathbf{d}_t \leftarrow \mathbf{g}_t + (1-\beta_t)\left(\mathbf{d}_{t-1} - \tilde{\mathbf{g}}_{t-1}\right)$
$\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta_t \mathbf{d}_t$
$\bar{\mathbf{x}}_{t+1} \leftarrow \gamma_t \mathbf{x}_{t+1} + (1-\gamma_t)\bar{\mathbf{x}}_t$
return $\bar{\mathbf{x}}_{t+1}, \mathbf{x}_{t+1}, \mathbf{d}_t$

Assumptions. We make the following assumptions for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\mathbf{z} \in \textbf{Support}\{\mathcal{D}\}$:

Convexity: The function $f(\cdot; \mathbf{z})$ is convex, and there exists $\mathbf{x}^\star \in \mathbb{R}^d$ such that $f(\mathbf{x}^\star) \leq f(\mathbf{x})$. We will denote $D_1 := \|\mathbf{x}_1 - \mathbf{x}^\star\|$, the distance between the initial point and the optimal point.

Bounded Variance: There exists $\sigma > 0$ such that $\mathbb{E}\left\|\nabla f(\mathbf{x}; \mathbf{z}) - \nabla f(\mathbf{x})\right\|^2 \leq \sigma^2$ (2)

Smoothness: There exists $L > 0$ such that $\left\|\nabla f(\mathbf{x}; \mathbf{z}) - \nabla f(\mathbf{y}; \mathbf{z})\right\| \leq L\left\|\mathbf{x} - \mathbf{y}\right\|$ (3)

This implies that the expected loss $f(\cdot)$ is L-smooth and that there exists $\sigma_L \in [0, L]$ such that

$$\mathbb{E}\left\|(\nabla f(\mathbf{x}; \mathbf{z}) - \nabla f(\mathbf{x})) - (\nabla f(\mathbf{y}; \mathbf{z}) - \nabla f(\mathbf{y}))\right\|^2 \leq \sigma_L^2 \left\|\mathbf{x} - \mathbf{y}\right\|^2 \tag{4}$$

The proof of Equation (4) is in Section A.

Notation: $\nabla f(\mathbf{x}; \mathbf{z})$ denotes the gradient with respect to $\mathbf{x}$. We use $\|\cdot\|$ to denote the Euclidean norm.

2 Local MixVR

We introduce Local MixVR, a unified framework for mitigating worker drift in distributed learning with infrequent communication. As discussed above, worker drift is a central limitation of local-update methods: when workers perform multiple stochastic steps between synchronization rounds, sampling noise causes their local models to diverge. Controlling this drift typically requires the number of communication rounds to grow with the sample complexity. Local MixVR breaks this dependence by combining complementary variance-reduction mechanisms that control drift at different stages of the algorithm. Their joint effect stabilizes local trajectories and limits the stochastic error accumulated between synchronizations, enabling less frequent communication while preserving the optimal convergence rate. Local MixVR consists of three main components:

I. **Local Double-Momentum.** During each local phase, every worker runs the μ^2-SGD algorithm (Dahan & Levy, 2025) (Algorithm 1) to control drift and keep local trajectories aligned. The algorithm combines two complementary momentum mechanisms:
   - Anytime Averaging (Cutkosky, 2019) maintains an averaged (momentum) sequence $\{\bar{\mathbf{x}}_t^{(i)}\}_{i=1}^M$ for the local iterates $\{\mathbf{x}_t^{(i)}\}_{i=1}^M$. Because the averaged iterates evolve more slowly, they remain close to the most recent synchronized global model parameters, stabilizing local trajectories and limiting worker drift.
   - The STORM estimator (Cutkosky & Orabona, 2019), which uses information from the previous local step to correct the current stochastic gradient estimate, thereby reducing its variance. This produces less noisy update directions, limiting divergence among workers' local trajectories.

II. **Budget Mixing: Local Progress vs. Minibatch Averaging.** Local MixVR uses a mixing parameter $\alpha \in (0, 1)$ to split the total budget of K samples into two complementary roles: making progress in optimization and reducing stochastic noise before synchronization. The parameter α controls this trade-off. Specifically, Local MixVR allocates $K_{\text{loc}} = \lfloor (1-\alpha)K \rfloor$ samples to local optimization steps, and reserves $K_{\text{avg}} = \lceil \alpha K \rceil$ samples for minibatch averaging. Thus, for the same budget of K samples, the local phase moves the optimization trajectory forward, while the averaging phase reduces stochastic noise before it is injected into the synchronized parameters.

III. **Drift Correction Mechanism.** Finally, Local MixVR corrects the gradient estimator at each synchronization to account for the drift accumulated during local updates. At synchronization step t, the workers first average their local models $\{\bar{\mathbf{x}}_t^{(i)}\}_{i=1}^M$ to obtain

$$\bar{\mathbf{x}}_t := \frac{1}{M}\sum_{i=1}^{M} \bar{\mathbf{x}}_t^{(i)}. \tag{5}$$

Each worker i then evaluates the gradient on the same minibatch $\mathcal{B}_t^{(i)}$ at its previous local parameter $\bar{\mathbf{x}}_{t-1}^{(i)}$ and at the synchronized parameter $\bar{\mathbf{x}}_t$. The difference

$$\nabla f\left(\bar{\mathbf{x}}_t; \mathcal{B}_t^{(i)}\right) - \nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}; \mathcal{B}_t^{(i)}\right)$$

estimates the gradient change from the local parameter to the synchronized model. Adding this term shifts the estimator to the synchronized point, correcting the mismatch accumulated during local updates. This realigns the local estimators with the global model and allows workers to begin the next local phase with reduced drift and stochastic error.

I. Local Double-Momentum. The first layer of our framework is a local double-momentum mechanism, denoted by μ^2-SGD . Following the Anytime-SGD framework of Cutkosky (2019), one momentum is applied directly to the model parameters. In particular, stochastic gradients are evaluated not at the current iterate $\mathbf{x}_t$, but at the momentum-averaged parameters $\bar{\mathbf{x}}_t$:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla f(\bar{\mathbf{x}}_t; \mathbf{z}_t)\,, \qquad \bar{\mathbf{x}}_{t+1} = \gamma_t \mathbf{x}_{t+1} + (1-\gamma_t)\bar{\mathbf{x}}_t \tag{6}$$

where $\mathbf{z}_t \sim \mathcal{D}$. With the choice $\gamma_t = \frac{2}{t+2}$, the averaged sequence evolves slowly:

$$\|\bar{\mathbf{x}}_{t+1} - \bar{\mathbf{x}}_t\| = \gamma_t \|\mathbf{x}_{t+1} - \bar{\mathbf{x}}_t\| \le \mathcal{O}\left(\frac{\|\mathbf{x}_{t+1} - \bar{\mathbf{x}}_t\|}{t}\right) \tag{7}$$

In the distributed setting, this property ensures that local model trajectories remain closely aligned across workers, effectively curbing the drift induced by local updates (Dahan & Levy, 2024). The second momentum mechanism is designed to reduce stochastic variance by correcting the bias inherent in standard momentum (Polyak, 1964). This bias arises because standard momentum averages past stochastic gradients, which were evaluated at previous, and therefore stale, iterates. The STORM estimator (Cutkosky & Orabona, 2019) addresses this by introducing a correction term:

$$\mathbf{d}_t = \underbrace{(1-\beta_t)\mathbf{d}_{t-1} + \beta_t \nabla f(\mathbf{x}_t; \mathbf{z}_t)}_{\text{standard momentum}} + \underbrace{(1-\beta_t)\left(\nabla f(\mathbf{x}_t; \mathbf{z}_t) - \nabla f(\mathbf{x}_{t-1}; \mathbf{z}_t)\right)}_{\text{correction term}}$$

Equivalently, this update can be written in the compact form used throughout the paper:

$$\mathbf{d}_t = \nabla f(\mathbf{x}_t; \mathbf{z}_t) + (1-\beta_t)\left(\mathbf{d}_{t-1} - \nabla f(\mathbf{x}_{t-1}; \mathbf{z}_t)\right)$$

The correction term measures how the stochastic gradient changes when moving from $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$, using the same sample $\mathbf{z}_t$. This adjustment compensates for the stale-gradient bias introduced by standard momentum. As a result, $\mathbf{d}_t$ is an (unconditionally) unbiased estimator of the true gradient $\nabla f(\mathbf{x}_t)$, that is, $\mathbb{E}\left[\mathbf{d}_t - \nabla f(\mathbf{x}_t)\right] = 0$, though it is conditionally biased. This unbiased estimator yields a stronger reduction in stochastic error than standard momentum (Cutkosky & Orabona, 2019).

μ^2-SGD . As shown by Dahan & Levy (2025), applying STORM to anytime-averaged iterates $\bar{\mathbf{x}}_t$ can substantially reduce stochastic variance. The resulting μ^2-SGD update is

$$\begin{aligned}
\mathbf{d}_t &= \nabla f(\bar{\mathbf{x}}_t; \mathbf{z}_t) + (1-\beta_t)\left(\mathbf{d}_{t-1} - \nabla f(\bar{\mathbf{x}}_{t-1}; \mathbf{z}_t)\right) \\
\mathbf{x}_{t+1} &= \mathbf{x}_t - \eta_t \mathbf{d}_t, \qquad \bar{\mathbf{x}}_{t+1} = \gamma_t \mathbf{x}_{t+1} + (1-\gamma_t)\bar{\mathbf{x}}_t
\end{aligned}$$

Choosing $\beta_t = 1/t$ allows μ^2-SGD to use the full gradient history, yielding a maximal $1/t$ reduction in stochastic variance (in a similar way to minibatch SGD)

$$\mathbb{E}\left\|\mathbf{d}_t - \nabla f(\bar{\mathbf{x}}_t)\right\|^2 \le \mathcal{O}\left(\frac{\tilde{\sigma}^2}{t}\right), \qquad \text{where } \tilde{\sigma} := \Theta\left(\sigma + \sigma_L D_1\right) \tag{8}$$

Moreover, when averaging over M workers, the error decreases further to $\mathcal{O}\left(\frac{\tilde{\sigma}^2}{Mt}\right)$.

Algorithm 2 Accumulation (minibatching) procedure and drift correction for worker i

Input: $\bar{\mathbf{x}}_t, \bar{\mathbf{x}}_{t-1}^{(i)}, \mathbf{d}_{t-1}^{(i)}, \beta_t, K_{\text{avg}}$
$\mathbf{g}_t^{(i)} \leftarrow 0, \quad \tilde{\mathbf{g}}_{t-1}^{(i)} \leftarrow 0$
for $k = 1, \dots, K_{\text{avg}}$ **do** ▷ accumulate gradients
  Sample $\mathbf{z}_t^{(i,k)} \sim \mathcal{D}$
  Compute $\mathbf{g}_t^{(i)} \leftarrow \mathbf{g}_t^{(i)} + \frac{1}{K_{\text{avg}}} \nabla f\left(\bar{\mathbf{x}}_t; \mathbf{z}_{t,k}^{(i)}\right)$
  Compute $\tilde{\mathbf{g}}_{t-1}^{(i)} \leftarrow \tilde{\mathbf{g}}_{t-1}^{(i)} + \frac{1}{K_{\text{avg}}} \nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}; \mathbf{z}_{t,k}^{(i)}\right)$
end for
$\mathbf{d}_t^{(i)} \leftarrow \mathbf{g}_t^{(i)} + (1-\beta_t)\left(\mathbf{d}_{t-1}^{(i)} - \tilde{\mathbf{g}}_{t-1}^{(i)}\right)$ ▷ drift correction
return $\mathbf{d}_t^{(i)}$

Local μ^2-SGD . In Local MixVR, each worker i applies μ^2-SGD locally and computes its own μ^2-SGD estimator during the local phase:

$$\mathbf{d}_t^{(i)} = \nabla f\left(\bar{\mathbf{x}}_t^{(i)}; \mathbf{z}_t^{(i)}\right) + (1-\beta_t)\left(\mathbf{d}_{t-1}^{(i)} - \nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}; \mathbf{z}_t^{(i)}\right)\right)$$
$$\mathbf{x}_{t+1}^{(i)} = \mathbf{x}_t^{(i)} - \eta_t \mathbf{d}_t^{(i)}, \qquad \bar{\mathbf{x}}_{t+1}^{(i)} = \gamma_t \mathbf{x}_{t+1}^{(i)} + (1-\gamma_t)\bar{\mathbf{x}}_t^{(i)}$$

The variance reduction achieved by μ^2-SGD (Equation (8)) can make the local updates more stable and better aligned across workers, thereby helping to mitigate worker drift. However, moving from the minibatch setting of Dahan & Levy (2025) to a local-update regime introduces local bias. This bias must therefore be controlled, as we show next with Local MixVR.

II. Minibatch Averaging (Accumulation Phase). After the local-update phase, Local MixVR synchronizes the model parameters as in Equation (5). The STORM estimators, however, cannot be averaged directly. Unlike the individual stochastic gradients used by Local SGD, each STORM estimator recursively incorporates a worker's previous stochastic gradients, which were evaluated at different local iterates, and therefore carries errors accumulated through local drift. Directly averaging these estimators would propagate this error to the global estimator. To prevent this, each synchronization step t is followed by a gradient-accumulation phase (Algorithm 2), which refines the estimators using gradients evaluated at the common synchronized model, reducing the variance carried over from the local phase.

For each worker i, let $\mathcal{B}_t^{(i)} = \left\{\mathbf{z}_{t,1}^{(i)}, \dots, \mathbf{z}_{t,K_{\text{avg}}}^{(i)}\right\}$ be a minibatch of size K_{avg}. During the accumulation phase, worker i evaluates two gradients using the same minibatch:

$$\nabla f\left(\bar{\mathbf{x}}_t; \mathcal{B}_t^{(i)}\right) := \frac{1}{K_{\text{avg}}} \sum_{k=1}^{K_{\text{avg}}} \nabla f\left(\bar{\mathbf{x}}_t; \mathbf{z}_{t,k}^{(i)}\right), \qquad \nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}; \mathcal{B}_t^{(i)}\right) := \frac{1}{K_{\text{avg}}} \sum_{k=1}^{K_{\text{avg}}} \nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}; \mathbf{z}_{t,k}^{(i)}\right)$$

Using the same minibatch at both points yields a coupled gradient difference. This coupling is important because the variance of the difference is controlled by the distance between the two iterates. Moreover, averaging over K_{avg} samples reduces this variance by a factor of K_{avg}. Hence, under the random-function smoothness setting (Equations (3) and (4)), we have

$$\mathbb{E}\left\|\left(\nabla f\left(\bar{\mathbf{x}}_t; \mathcal{B}_t^{(i)}\right) - \nabla f(\bar{\mathbf{x}}_t)\right) - \left(\nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}; \mathcal{B}_t^{(i)}\right) - \nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right)\right)\right\|^2 \le \frac{\sigma_L^2}{K_{\text{avg}}} \mathbb{E}\left\|\bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{t-1}^{(i)}\right\|^2 \tag{9}$$

We average over the workers and decompose this distance into two terms:

$$\frac{1}{M}\sum_{i=1}^{M}\left\|\bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{t-1}^{(i)}\right\|^2 = \underbrace{\left\|\bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{t-1}\right\|^2}_{\text{global step}} + \underbrace{\frac{1}{M}\sum_{i=1}^{M}\left\|\bar{\mathbf{x}}_{t-1} - \bar{\mathbf{x}}_{t-1}^{(i)}\right\|^2}_{\text{worker drift}}$$

The first term captures the global model's most recent update, while the second measures the discrepancy between each local model and the synchronized model at time t. As in Equation (7), the

anytime mechanism reduces the distance between successive local iterates:

$$\left\|\bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{t-1}\right\|^2 \leq \gamma_{t-1}^2 \left\|\mathbf{x}_t - \bar{\mathbf{x}}_{t-1}\right\|^2 \leq \mathcal{O}\left(\frac{\tilde{D}^2}{t^2}\right) \tag{10}$$

Here, $\tilde{D}^2$ is defined in Section B and depends on the initial distance D_1 and the stochastic error across workers. By Lemma D.2, it bounds the distance of any local iterate $\bar{\mathbf{x}}_t$ or $\mathbf{x}_t$ from $\mathbf{x}^\star$, and therefore also bounds distances between two generated iterates (see, e.g., Equation (21)).

Let t_0 denote the last synchronization time before t. Since the local and global momentum sequences are initialized at the same point $\bar{\mathbf{x}}_{t_0+1}$, this term cancels when subtracting their unrolled recursions. Hence, for $\gamma_t = \frac{2}{t+2}$, $\bar{\mathbf{x}}_t^{(i)} - \bar{\mathbf{x}}_t = \frac{2}{t(t+1)} \sum_{k=t_0+2}^{t} k \left(\mathbf{x}_k^{(i)} - \mathbf{x}_k\right)$. Applying Lemma D.3 yields

$$\frac{1}{M}\sum_{i=1}^{M} \left\|\bar{\mathbf{x}}_t^{(i)} - \bar{\mathbf{x}}_t\right\|^2 \leq \mathcal{O}\left(\frac{K_{\text{loc}}^2 \tilde{D}^2}{t^2}\right)$$

Hence, before accumulation, the worker drift exceeds the local-iterate term by a factor of order K_{loc}^2. The accumulation phase reduces this contribution by the additional factor K_{avg} in Equation (9). Thus, with $K_{\text{avg}} = \lceil \alpha K \rceil$, $K_{\text{loc}} = \lfloor (1-\alpha)K \rfloor$, and $\alpha \in (0,1)$, the accumulated worker-drift scales as

$$\mathcal{O}\left(\frac{K_{\text{loc}}^2}{K_{\text{avg}}} \frac{\tilde{D}^2}{t^2}\right) = \mathcal{O}\left(\frac{K\tilde{D}^2}{t^2}\right) \tag{11}$$

for constant α. Within each synchronization round, the accumulated worker-drift error is of order $\mathcal{O}\left(K\tilde{D}^2/t^2\right)$. The local-iterate error also has the same scale: it is incurred over K_{loc} local steps, each contributing $\approx \mathcal{O}\left(\tilde{D}^2/t^2\right)$, yielding a total of $\approx \mathcal{O}\left(K\tilde{D}^2/t^2\right)$. Thus, after accumulation, both effects enter in the same order per round. In this sense, the accumulation procedure keeps the variance due to worker drift comparable to the variance-reduction effect of the local iterates.

III. Drift Correction Mechanism. Prior to a synchronization step t, each worker's STORM estimator $\mathbf{d}_{t-1}^{(i)}$ is anchored to its local history, with an expectation $\mathbb{E}\left[\mathbf{d}_{t-1}^{(i)} - \nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right)\right] = 0$. After synchronization, however, all workers move to the synchronized parameter $\bar{\mathbf{x}}_t$. Therefore, the estimator $\mathbf{d}_t^{(i)}$ should also be shifted from the stale local target $\nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right)$ to the new global target $\nabla f(\bar{\mathbf{x}}_t)$. In other words, we want the corrected estimator to satisfy $\mathbb{E}\left[\mathbf{d}_t^{(i)} - \nabla f(\bar{\mathbf{x}}_t)\right] = 0$. We achieve this by using the gradients from the accumulation phase as a bridge. Specifically, we use $\nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}; \mathcal{B}_t^{(i)}\right)$ to subtract the error inherited from the local path and $\nabla f\left(\bar{\mathbf{x}}_t; \mathcal{B}_t^{(i)}\right)$ to re-center the estimator toward the new synchronized parameter. This synchronization correction is defined as:

$$\tilde{\mathbf{d}}_t^{(i)} = (1-\beta_t)\mathbf{d}_{t-1}^{(i)} + \beta_t \nabla f\left(\bar{\mathbf{x}}_t; \mathcal{B}_t^{(i)}\right) + (1-\beta_t) \underbrace{\left(\nabla f\left(\bar{\mathbf{x}}_t; \mathcal{B}_t^{(i)}\right) - \nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}; \mathcal{B}_t^{(i)}\right)\right)}_{\text{drift correction}}$$

This correction removes the bias induced by the worker's local drift and shifts the estimator toward the synchronized global parameter. Consequently, $\mathbb{E}\left[\tilde{\mathbf{d}}_t^{(i)} - \nabla f(\bar{\mathbf{x}}_t)\right] = 0$, and after synchronizing the corrected estimators across all M workers, the local corrected momentum $\mathbf{d}_t^{(i)} = \mathbf{d}_t = \frac{1}{M}\sum_{i=1}^{M} \tilde{\mathbf{d}}_t^{(i)}$ preserves the same property $\mathbb{E}\left[\mathbf{d}_t^{(i)} - \nabla f(\bar{\mathbf{x}}_t)\right] = 0$. Moreover, as discussed above, because the correction uses minibatching during the accumulation phase, the drift error remains controlled.

Local MixVR. The full procedure is summarized in Algorithm 3. We now turn to the theoretical analysis of Local MixVR and present its main guarantees.

Lemma 2.1 (Local Stochastic Noise)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex L-smooth function with global minimizer $\mathbf{x}^\star$ and suppose that Equations (2) and (3) hold. Then, applying Algorithm 3 with*

Algorithm 3 Local MixVR

Input: $\bar{\mathbf{x}}_0 \in \mathbb{R}^d$; parameters $\{\eta_t > 0, \beta_t, \gamma_t \in (0,1)\}_t$; workers M; rounds R; steps K; mixing ratio $\alpha \in (0,1)$. Set $K_{\text{loc}} = \lfloor (1-\alpha)K \rfloor$, $K_{\text{avg}} = \lceil \alpha K \rceil$, $\beta_1 = 1$, and $t = 1$. Let $\mathcal{A}$ be the accumulation and drift-correction procedure in Algorithm 2, and μ^2-SGD the update in Algorithm 1.
Initialize: $\bar{\mathbf{x}}_1^{(i)} = \mathbf{x}_1^{(i)} = \bar{\mathbf{x}}_0$ and $\mathbf{d}_0^{(i)} = \mathbf{0}$ for all $i \in [M]$.
for $r = 1, \dots, R$ **do**
  $\bar{\mathbf{x}}_{t-1}^{(i)} \leftarrow \bar{\mathbf{x}}_{t-1}$ for all $i \in [M]$
  for $k = 1, \dots, K_{\text{loc}}$ **do** ▷ $t = (r-1)(K_{\text{loc}}+1) + k$
    for $i = 1, \dots, M$ **in parallel do** ▷ local updates
      $(\bar{\mathbf{x}}_{t+1}^{(i)}, \mathbf{x}_{t+1}^{(i)}, \mathbf{d}_t^{(i)}) \leftarrow \mu^2\text{-SGD}\,(\bar{\mathbf{x}}_t^{(i)}, \bar{\mathbf{x}}_{t-1}^{(i)}, \mathbf{x}_t^{(i)}, \mathbf{d}_{t-1}^{(i)}, \eta_t, \beta_t, \gamma_t), \quad t \leftarrow t+1$
    end for
  end for ▷ $t = r(K_{\text{loc}}+1)$
  $\bar{\mathbf{x}}_t \leftarrow \frac{1}{M}\sum_{i=1}^M \bar{\mathbf{x}}_t^{(i)}, \qquad \mathbf{x}_t \leftarrow \frac{1}{M}\sum_{i=1}^M \mathbf{x}_t^{(i)}$ ▷ parameter all-reduce
  for $i = 1, \dots, M$ **in parallel do**
    $\tilde{\mathbf{d}}_t^{(i)} \leftarrow \mathcal{A}(\bar{\mathbf{x}}_t, \bar{\mathbf{x}}_{t-1}^{(i)}, \mathbf{d}_{t-1}^{(i)}, K_{\text{avg}})$ ▷ accumulate and correct drift
  end for
  $\mathbf{d}_t \leftarrow \frac{1}{M}\sum_{i=1}^M \tilde{\mathbf{d}}_t^{(i)}$ ▷ estimator all-reduce
  for $i = 1, \dots, M$ **in parallel do**
    $\mathbf{d}_t^{(i)} \leftarrow \mathbf{d}_t, \quad \mathbf{x}_{t+1}^{(i)} \leftarrow \mathbf{x}_t - \eta_t \mathbf{d}_t, \quad \bar{\mathbf{x}}_{t+1}^{(i)} \leftarrow \gamma_t \mathbf{x}_{t+1}^{(i)} + (1-\gamma_t)\bar{\mathbf{x}}_t, \quad t \leftarrow t+1$
  end for
end for
return $\bar{\mathbf{x}}_t$ ▷ $t = R(K_{\text{loc}}+1)$

$\alpha = \frac{1}{2}, \beta_t = \frac{1}{t}, \gamma_t = \frac{2}{t+2}, \eta_t = t \cdot \eta$, where assume that the stepsize η at every time $t \geq 1$ satisfies $\eta \leq \frac{1}{8t}\min\left\{\frac{1}{L}, \frac{1}{\sigma_L\sqrt{4K_{\text{loc}}+\frac{5t}{M}}}\right\}$, gives

$$\frac{1}{M}\sum_{i=1}^M t^2 \mathbb{E}\left\|\boldsymbol{\epsilon}_t^{(i)}\right\|^2 \leq \underbrace{4K_{\text{loc}}\tilde{\sigma}^2}_{\text{local error}} + \underbrace{\frac{5t\tilde{\sigma}^2}{M}}_{\text{shared error}}$$

where $\tilde{\sigma}^2 = \sigma^2 + 32\sigma_L^2 D_1^2$.

Remark 2.1. Lemma 2.1 highlights a key temporal divide in the stochastic error of Local MixVR. At synchronization, each worker effectively incorporates information from all workers, so stochasticity before the most recent round is globalized, producing the shared-error term and recovering the optimal $1/M$ minibatch variance reduction. Only the "raw" stochasticity from the current local window of length $\mathcal{O}(K_{\text{loc}})$ avoids this reduction. Thus, as t grows, local noise becomes negligible: nearly all optimization history is globally averaged, while only the most recent $\mathcal{O}(K_{\text{loc}})$ steps remain local.

Theorem 2.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex L-smooth function with global minimizer $\mathbf{x}^\star$ and suppose that Equations (2) and (3) hold. Then, applying Algorithm 3 with $\alpha = \frac{1}{2}, \beta_t = \frac{1}{t}, \gamma_t = \frac{2}{t+2}, \eta_t = t \cdot \eta$, where $\eta = \min\left\{\frac{1}{8LT}, \frac{D_1}{T\sqrt{2\sigma^2+64\sigma_L^2 D_1^2}\sqrt{4K_{\text{loc}}+\frac{5T}{M}}}\right\}$ gives*

$$\Delta_T \leq \mathcal{O}\left(\frac{LD_1^2}{KR} + \frac{\tilde{\sigma}D_1}{K^{1/2}R} + \frac{\tilde{\sigma}D_1}{\sqrt{MKR}}\right)$$

where $T = R(K_{\text{loc}}+1)$ is the final iteration, $\tilde{\sigma} = \Theta(\sigma + \sigma_L D_1)$, and $\Delta_T := \mathbb{E}[f(\bar{\mathbf{x}}_T)] - f(\mathbf{x}^\star)$, with $\bar{\mathbf{x}}_T := \frac{1}{M}\sum_{i=1}^M \bar{\mathbf{x}}_T^{(i)}$.

Remark 2.3. Theorem 2.2 establishes that Local MixVR achieves the optimal statistical rate while reducing the required communication rounds compared to existing methods (Table 1). Specifically, in the regime $M \leq \mathcal{O}\left(N^{1/4}\right)$, our bound improves upon the state-of-the-art Minibatch ASGD baseline. Notably, the communication complexity derived from Theorem 2.2 is entirely independent of the total sample size N, scaling only with the number of workers M. This sample-independent guarantee ensures that Local MixVR remains communication-efficient as datasets grow, making it well-suited to common settings with massive datasets.

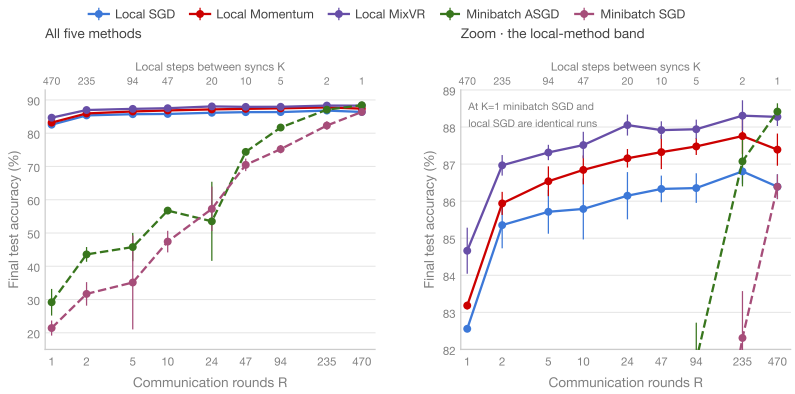

Figure 1: **Test accuracy vs communication rounds R on Fashion-MNIST. $M = 4$ workers.** The top axis reports the local steps $K = \lceil 470/R \rceil$. *left:* all five methods; *right:* the local-method accuracy range.

3 Experiments

We evaluate the effect of the communication budget on image classification using Fashion-MNIST (Xiao et al., 2017) and CIFAR-10 (Krizhevsky et al., 2014). All methods are implemented in PyTorch and are evaluated using the same data budget at each communication setting. The Fashion-MNIST experiments were run using an Apple M4 chip, whereas the CIFAR-10 experiments were run on an NVIDIA A100 GPU. Each point reports the mean final test accuracy over three independent seeds, with error bars denoting one standard deviation. For Fashion-MNIST, we train a LeNet-style convolutional network with $M \in \{4, 8, 16\}$ workers, a per-worker batch size of 64, and a fixed global budget of two epochs. For CIFAR-10, we train a ResNet-18 (He et al., 2016) with GroupNorm, initialized from an ImageNet checkpoint, using $M = 8$ workers, a per-worker batch size of 16, and 30 epochs. Further experimental results and implementation details are provided in Section G.

We compare Local MixVR with Local SGD, Local Momentum, Minibatch SGD, and Minibatch ASGD. On Fashion-MNIST, Local MixVR achieves the highest accuracy among the local-update methods for every tested value of R and all three worker counts (Figures 1, 3 and 4). Although the absolute accuracy of the local methods decreases as the fixed global data budget is distributed across more workers, the benefit of variance reduction becomes more pronounced. At $R = 24$, the margin of Local MixVR over Local SGD increases from 1.90 percentage points for $M = 4$ to 2.93 for $M = 8$ and 4.03 for $M = 16$; the corresponding margins for Local Momentum are 1.01, 1.99, and 2.94 points. The minibatch baselines are less sensitive to the worker count, but Minibatch ASGD becomes more competitive at large communication budgets. The CIFAR-10 results exhibit the same communication-efficient behavior (Figure 2). Local MixVR outperforms all evaluated baselines throughout the reported range and is especially effective when communication is limited, whereas the minibatch methods require more global updates to approach the accuracy of the local methods. Together, these results support the theoretical prediction that Local MixVR can reduce synchronization frequency without sacrificing optimization quality.

4 Conclusions and Future Work

In this work, we show for the first time that the Minibatch ASGD baseline can be surpassed in common regimes, achieving high performance with reduced communication that is independent of the total number of samples N. By appropriately integrating several variance-reduction techniques, our framework addresses complementary sources of error and effectively mitigates bias accumulated from local updates. This combination is conceptually natural, and we believe this design principle will prove useful beyond the specific algorithm studied here. For future work, it would be interesting to derive lower bounds for the distributed setting considered here under the random smooth-function model. Furthermore, it would be valuable to investigate accelerated variants of our framework, as acceleration has been shown to be a practical and effective approach to enhancing distributed learning with local steps (Douillard et al., 2023).

AI USE STATEMENT

In this work, we have not used generative AI tools for any tasks for which disclosure is required under the ICLR 2027 AI Policy for Authors. Additionally, we used generative AI tools to polish the writing and improve the readability of the manuscript. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Alekh Agarwal, Martin J Wainwright, Peter Bartlett, and Pradeep Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. *Advances in Neural Information Processing Systems*, 22, 2009.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Ziheng Cheng, Xinmeng Huang, Pengfei Wu, and Kun Yuan. Momentum benefits non-iid federated learning simply and provably. *arXiv preprint arXiv:2306.16504*, 2023.

Ashok Cutkosky. Anytime online-to-batch, optimism and acceleration. In *International conference on machine learning*, pp. 1446–1454. PMLR, 2019.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

Tehila Dahan and Kfir Y Levy. Slowcalsgd: Slow query points improve local-sgd for stochastic convex optimization. *Advances in Neural Information Processing Systems*, 37:93171–93212, 2024.

Tehila Dahan and Kfir Y Levy. Do stochastic, feel noiseless: Stable stochastic optimization via a double momentum mechanism. In *International Conference on Learning Representations*, volume 2025, pp. 84007–84041, 2025.

Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(1), 2012.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Arthur Douillard, Qixuan Feng, Andrei A Rusu, Rachita Chhaparia, Yani Donchev, Adhiguna Kuncoro, Marc'Aurelio Ranzato, Arthur Szlam, and Jiajun Shen. Diloco: Distributed low-communication training of language models. *arXiv preprint arXiv:2311.08105*, 2023.

Margalit R Glasgow, Honglin Yuan, and Tengyu Ma. Sharp bounds for federated averaging (local sgd) and continuous perspective. In *International Conference on Artificial Intelligence and Statistics*, pp. 9050–9090. PMLR, 2022.

Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. Local sgd: Unified theory and new efficient methods. In *International Conference on Artificial Intelligence and Statistics*, pp. 3556–3564. PMLR, 2021.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

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.

Sami Jaghouar, Jack Min Ong, Manveer Basra, Fares Obeid, Jannik Straube, Michael Keiblinger, Elie Bakouch, Lucas Atkins, Maziyar Panahi, Charles Goddard, et al. Intellect-1 technical report. *arXiv preprint arXiv:2412.01152*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606*, 2020a.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020b.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International conference on artificial intelligence and statistics*, pp. 4519–4529. PMLR, 2020.

Ahmed Khaled, Satyen Kale, Arthur Douillard, Chi Jin, Rob Fergus, and Manzil Zaheer. Understanding outer optimizers in local sgd: Learning rates, momentum, and acceleration. *arXiv preprint arXiv:2509.10439*, 2025.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, pp. 5381–5393. PMLR, 2020.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5), 2014.

Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*, 2018.

Olvi L Mangasarian and Mikhail V Solodov. Backpropagation convergence via deterministic nonmonotone perturbed minimization. *Advances in Neural Information Processing Systems*, 6, 1993.

Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Proximal and federated random reshuffling. In *International Conference on Machine Learning*, pp. 15718–15749. PMLR, 2022a.

Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, and Peter Richtárik. Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! In *International Conference on Machine Learning*, pp. 15750–15769. PMLR, 2022b.

Aritra Mitra, Rayana Jaafar, George J Pappas, and Hamed Hassani. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. *Advances in Neural Information Processing Systems*, 34:14606–14619, 2021.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Kumar Kshitij Patel, Lingxiao Wang, Blake Woodworth, Brian Bullins, and Nathan Srebro. Towards optimal communication complexity in distributed non-convex optimization. In *Advances in Neural Information Processing Systems*, 2022.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

Sebastian U Stich. Local sgd converges fast and communicates little. In *International Conference on Learning Representations*, 2019.

Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *Acm computing surveys (csur)*, 53(2): 1–33, 2020.

Blake E Woodworth, Brian Bullins, Ohad Shamir, and Nathan Srebro. The min-max complexity of distributed stochastic convex optimization with intermittent communication. In *Conference on Learning Theory*, pp. 4386–4437. PMLR, 2021.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Honglin Yuan and Tengyu Ma. Federated accelerated stochastic gradient descent. *Advances in Neural Information Processing Systems*, 33:5332–5344, 2020.

Riccardo Zaccone, Sai Praneeth Karimireddy, Carlo Masone, and Marco Ciccone. Communication-efficient heterogeneous federated learning with generalized heavy-ball momentum. *arXiv preprint arXiv:2311.18578*, 2023.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

A Implications of the Smoothness Assumption

We first show that Equation (3) implies that Equation (4) holds for some $\sigma_L \in [0, L]$.

$$\begin{aligned}&\mathbb{E}\left\|(\nabla f(\mathbf{x};\mathbf{z}) - \nabla f(\mathbf{x})) - (\nabla f(\mathbf{y};\mathbf{z}) - \nabla f(\mathbf{y}))\right\|^2 \\ =&\mathbb{E}\left\|\nabla f(\mathbf{x};\mathbf{z}) - \nabla f(\mathbf{y};\mathbf{z})\right\|^2 - \left\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\right\|^2 \leq L^2\left\|\mathbf{x}-\mathbf{y}\right\|^2\end{aligned}$$

Here, we used the identity $\mathbb{E}\left[\nabla f(\mathbf{x};\mathbf{z}) - \nabla f(\mathbf{y};\mathbf{z})\right] = \nabla f(\mathbf{x}) - \nabla f(\mathbf{y})$, together with the identity $\mathbb{E}\left\|X - \mathbb{E}\left[X\right]\right\|^2 = \mathbb{E}\left\|X\right\|^2 - \left\|\mathbb{E}\left[X\right]\right\|^2$, and finally Equation (3). Therefore, we conclude that $\sigma_L \in [0, L]$.

Lemma A.1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be an L-smooth function with global minimizer $\mathbf{x}^\star$. Then, for any $\mathbf{x} \in \mathbb{R}^d$,*

$$\left\|\nabla f(\mathbf{x})\right\|^2 \leq 2L\left(f(\mathbf{x}) - f(\mathbf{x}^\star)\right)$$

Proof of Lemma A.1. By L-smoothness, for every $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

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

Choosing $\mathbf{y} = \mathbf{x} - \frac{1}{L}\nabla f(\mathbf{x})$ gives

$$f(\mathbf{y}) \leq f(\mathbf{x}) - \frac{1}{L}\left\|\nabla f(\mathbf{x})\right\|^2 + \frac{1}{2L}\left\|\nabla f(\mathbf{x})\right\|^2 = f(\mathbf{x}) - \frac{1}{2L}\left\|\nabla f(\mathbf{x})\right\|^2$$

Using $f(\mathbf{y}) \geq f(\mathbf{x}^\star)$ and rearranging yields the claim. □

B Local MixVR Analysis

We use $\mathcal{S}_t \subseteq \mathbb{N}$ to denote the set of synchronization steps up to iteration t. For each worker i, define the local stochastic error and the local optimality gap by

$$\boldsymbol{\epsilon}_t^{(i)} := \begin{cases}\mathbf{d}_t^{(i)} - \nabla f\left(\bar{\mathbf{x}}_t^{(i)}\right), & t \notin \mathcal{S}_T \\ \mathbf{d}_t^{(i)} - \nabla f(\bar{\mathbf{x}}_t), & t \in \mathcal{S}_T\end{cases}, \qquad \Delta_t^{(i)} := \begin{cases}f\left(\bar{\mathbf{x}}_t^{(i)}\right) - f(\mathbf{x}^\star), & t \notin \mathcal{S}_T \\ f(\bar{\mathbf{x}}_t) - f(\mathbf{x}^\star), & t \in \mathcal{S}_T\end{cases}$$

For a more refined analysis, define $\tilde{D}_t^2 := 2D_1^2 + \frac{4\eta^2 t}{M}\sum_{i=1}^M\sum_{k=1}^t k^2\mathbb{E}\left\|\boldsymbol{\epsilon}_k^{(i)}\right\|^2$. Note that $\tilde{D}_t^2$ is non-decreasing in t. We will denote $\tilde{D}^2 := \tilde{D}_T^2$, as it is a bound on all $\tilde{D}_t^2$.

C Auxiliary Inequalities

In this section, we collect several auxiliary inequalities that will be used later in the analysis.

Lemma C.1 ((Dahan & Levy, 2025)). *Let $\{A_t\}_{t\in[T]}$ be a sequence of nonnegative elements and $B \in \mathbb{R}$, and assume that for any $t \in [T]$:*

$$A_t \leq B + \frac{1}{2T}\sum_{\tau=1}^T A_\tau$$

Then the following bound holds:

$$A_t \leq 2B$$

Proof of Lemma C.1. Summing over $t = 1, \ldots, T$, we obtain $A_{1:T} \leq TB + \frac{1}{2}A_{1:T}$, and hence $A_{1:T} \leq 2TB$. Substituting this estimate back into the original bound yields:

$$A_t \leq B + \frac{1}{2T}A_{1:T} \leq 2B$$

□

Lemma C.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex, and define $\bar{\mathbf{x}}_t := \frac{1}{M}\sum_{i=1}^M \bar{\mathbf{x}}_t^{(i)}$, $\Delta_t := f(\bar{\mathbf{x}}_t) - f(\mathbf{x}^\star)$, and $\Delta_t^{(i)} := f\left(\bar{\mathbf{x}}_t^{(i)}\right) - f(\mathbf{x}^\star)$. Then $\Delta_t \le \frac{1}{M}\sum_{i=1}^M \Delta_t^{(i)}$.*

Proof of Lemma C.2. Since f is convex, Jensen's inequality implies:

$$f(\bar{\mathbf{x}}_t) = f\left(\frac{1}{M}\sum_{i=1}^M \bar{\mathbf{x}}_t^{(i)}\right) \le \frac{1}{M}\sum_{i=1}^M f\left(\bar{\mathbf{x}}_t^{(i)}\right)$$

Subtracting $f(\mathbf{x}^\star)$ from both sides yields $\Delta_t \le \frac{1}{M}\sum_{i=1}^M \Delta_t^{(i)}$. □

For synchronization steps, $\Delta_t^{(i)} = \Delta_t$, so the lemma is trivial.

Lemma C.3 (Similar to Theorem 1 in Cutkosky (2019))**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function with global minimizer $\mathbf{x}^\star$. Let $\{\mathbf{x}_t\}_{t\ge1} \subset \mathbb{R}^d$ and $\{\bar{\mathbf{x}}_t\}_{t\ge1} \subset \mathbb{R}^d$ be as in Algorithm 3 with $\gamma_t = \frac{2}{t+2}$. Then the following holds for any $t \ge 1$:*

$$\begin{aligned}
&\frac{t(t+1)}{2M}\sum_{i=1}^M \Delta_t^{(i)} \\
\le&\frac{1}{M}\sum_{i=1}^M\left(\sum_{\tau\in[t]\setminus\mathcal{S}_t} \tau\left\langle\nabla f\left(\bar{\mathbf{x}}_\tau^{(i)}\right), \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\rangle + \sum_{\tau\in\mathcal{S}_t}\tau\left\langle\nabla f(\bar{\mathbf{x}}_\tau), \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\rangle\right)
\end{aligned}$$

Proof of Lemma C.3. Following Cutkosky (2019), define $\alpha_t := t$ and $\alpha_{1:t} := \sum_{\tau=1}^t \alpha_\tau$. Then:

$$\frac{\alpha_{t+1}}{\alpha_{1:t+1}} = \frac{t+1}{\sum_{\tau=1}^{t+1}\tau} = \frac{2}{t+2} = \gamma_t$$

Accordingly, in what follows, we use the equivalent representation of γ_t in terms of $\{\alpha_t\}_t$.

For a **non-synchronization step** after the first one, the momentum update step can be written as:

$$\bar{\mathbf{x}}_t^{(i)} = \frac{\alpha_t}{\alpha_{1:t}}\mathbf{x}_t^{(i)} + \left(1 - \frac{\alpha_t}{\alpha_{1:t}}\right)\bar{\mathbf{x}}_{t-1}^{(i)}$$

Similarly, for a **first step after synchronization**:

$$\bar{\mathbf{x}}_t^{(i)} = \frac{\alpha_t}{\alpha_{1:t}}\mathbf{x}_t^{(i)} + \left(1 - \frac{\alpha_t}{\alpha_{1:t}}\right)\bar{\mathbf{x}}_{t-1}$$

At a **synchronization step**:

$$\bar{\mathbf{x}}_t = \frac{\alpha_t}{\alpha_{1:t}}\mathbf{x}_t + \left(1 - \frac{\alpha_t}{\alpha_{1:t}}\right)\bar{\mathbf{x}}_{t-1}$$

From convexity, we know that $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, f(\mathbf{x}) - f(\mathbf{y}) \le \langle\nabla f(\mathbf{x}), \mathbf{x} - \mathbf{y}\rangle$. In a non-synchronization step, the update rule is $\alpha_{1:t}\bar{\mathbf{x}}_t^{(i)} = \alpha_{1:t-1}\bar{\mathbf{x}}_{t-1}^{(i)} + \alpha_t\mathbf{x}_t^{(i)}$. We get:

$$\begin{aligned}
&\alpha_{1:t}\Delta_t^{(i)} := \alpha_{1:t}\left(f\left(\bar{\mathbf{x}}_t^{(i)}\right) - f(\mathbf{x}^\star)\right) \\
=&\alpha_{1:t}f\left(\bar{\mathbf{x}}_t^{(i)}\right) - \alpha_{1:t-1}f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right) + \alpha_{1:t-1}f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right) - \alpha_{1:t}f(\mathbf{x}^\star) \\
=&\alpha_{1:t-1}\left(f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right) - f(\mathbf{x}^\star)\right) + \alpha_{1:t-1}\left(f\left(\bar{\mathbf{x}}_t^{(i)}\right) - f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right)\right) + \alpha_t\left(f\left(\bar{\mathbf{x}}_t^{(i)}\right) - f(\mathbf{x}^\star)\right) \\
\le&\alpha_{1:t-1}\left(f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right) - f(\mathbf{x}^\star)\right) + \alpha_{1:t-1}\left\langle\nabla f\left(\bar{\mathbf{x}}_t^{(i)}\right), \bar{\mathbf{x}}_t^{(i)} - \bar{\mathbf{x}}_{t-1}^{(i)}\right\rangle + \alpha_t\left\langle\nabla f\left(\bar{\mathbf{x}}_t^{(i)}\right), \bar{\mathbf{x}}_t^{(i)} - \mathbf{x}^\star\right\rangle \\
=&\alpha_{1:t-1}\left(f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right) - f(\mathbf{x}^\star)\right) + \left\langle\nabla f\left(\bar{\mathbf{x}}_t^{(i)}\right), \alpha_{1:t}\bar{\mathbf{x}}_t^{(i)} - \alpha_{1:t-1}\bar{\mathbf{x}}_{t-1}^{(i)} - \alpha_t\mathbf{x}^\star\right\rangle \\
=&\alpha_{1:t-1}\left(f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right) - f(\mathbf{x}^\star)\right) + \alpha_t\left\langle\nabla f\left(\bar{\mathbf{x}}_t^{(i)}\right), \mathbf{x}_t^{(i)} - \mathbf{x}^\star\right\rangle \\
=&\alpha_{1:t-1}\Delta_{t-1}^{(i)} + \alpha_t\left\langle\nabla f\left(\bar{\mathbf{x}}_t^{(i)}\right), \mathbf{x}_t^{(i)} - \mathbf{x}^\star\right\rangle
\end{aligned}$$

Similarly, in a synchronization step, the update rule is $\alpha_{1:t}\bar{\mathbf{x}}_t = \alpha_{1:t-1}\bar{\mathbf{x}}_{t-1} + \alpha_t \mathbf{x}_t$. We get:

$$\begin{aligned}
\alpha_{1:t}\Delta_t^{(i)} :=& \alpha_{1:t}\left(f(\bar{\mathbf{x}}_t) - f(\mathbf{x}^\star)\right) \\
\leq& \alpha_{1:t-1}\left(f(\bar{\mathbf{x}}_{t-1}) - f(\mathbf{x}^\star)\right) + \alpha_t \left\langle \nabla f(\bar{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star \right\rangle \\
\leq& \frac{1}{M}\sum_{i=1}^M \alpha_{1:t-1}\left(f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right) - f(\mathbf{x}^\star)\right) + \alpha_t \left\langle \nabla f(\bar{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star \right\rangle \\
=& \frac{1}{M}\sum_{i=1}^M \alpha_{1:t-1}\Delta_{t-1}^{(i)} + \frac{1}{M}\sum_{i=1}^M \alpha_t \left\langle \nabla f(\bar{\mathbf{x}}_t), \mathbf{x}_t^{(i)} - \mathbf{x}^\star \right\rangle
\end{aligned}$$

where the inequality uses Lemma C.2.

In a step after synchronization, the update rule is $\alpha_{1:t}\bar{\mathbf{x}}_t^{(i)} = \alpha_{1:t-1}\bar{\mathbf{x}}_{t-1} + \alpha_t \mathbf{x}_t^{(i)}$. Therefore:

$$\alpha_{1:t}\Delta_t^{(i)} \leq \alpha_{1:t-1}\Delta_{t-1}^{(i)} + \alpha_t \left\langle \nabla f\left(\bar{\mathbf{x}}_t^{(i)}\right), \mathbf{x}_t^{(i)} - \mathbf{x}^\star \right\rangle$$

When averaging over $i \in [M]$, both options yield the same result for non-synchronization steps:

$$\frac{1}{M}\sum_{i=1}^M \alpha_{1:t}\Delta_t^{(i)} \leq \frac{1}{M}\sum_{i=1}^M \alpha_{1:t-1}\Delta_{t-1}^{(i)} + \frac{1}{M}\sum_{i=1}^M \alpha_t \left\langle \nabla f\left(\bar{\mathbf{x}}_t^{(i)}\right), \mathbf{x}_t^{(i)} - \mathbf{x}^\star \right\rangle$$

For the synchronization steps, we have:

$$\frac{1}{M}\sum_{i=1}^M \alpha_{1:t}\Delta_t^{(i)} \leq \frac{1}{M}\sum_{i=1}^M \alpha_{1:t-1}\Delta_{t-1}^{(i)} + \frac{1}{M}\sum_{i=1}^M \alpha_t \left\langle \nabla f(\bar{\mathbf{x}}_t), \mathbf{x}_t^{(i)} - \mathbf{x}^\star \right\rangle$$

Unrolling, we get:

$$\begin{aligned}
\frac{1}{M}\sum_{i=1}^M \alpha_{1:t}\Delta_t^{(i)} \leq& \frac{1}{M}\sum_{i=1}^M \sum_{\tau \in [t]\setminus \mathcal{S}_t} \alpha_\tau \left\langle \nabla f\left(\bar{\mathbf{x}}_\tau^{(i)}\right), \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star \right\rangle \\
&+ \frac{1}{M}\sum_{i=1}^M \sum_{\tau \in \mathcal{S}_t} \alpha_\tau \left\langle \nabla f(\bar{\mathbf{x}}_\tau), \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star \right\rangle
\end{aligned}$$

Finally, since $\alpha_t = t$ and $\alpha_{1:t} = \sum_{\tau=1}^t \tau = \frac{t(t+1)}{2}$, we input these values. This proves the result. □

D Convergence Bounds

Lemma D.1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex, L-smooth function with global minimizer $\mathbf{x}^\star$, and let $\{\mathbf{x}_t^{(i)}\}_{t\geq 1, i\in[M]}$ be the iterates generated by Algorithm 3 with $\gamma_t = \frac{2}{t+2}, \eta_t = t\cdot\eta$. Then, for any $T > 0$:*

$$\begin{aligned}
&\frac{1}{2\eta M}\sum_{i=1}^M \left\|\mathbf{x}_{T+1}^{(i)} - \mathbf{x}^\star\right\|^2 + \frac{1}{M}\sum_{i=1}^M\sum_{t=1}^T t\left\langle \mathbf{d}_t^{(i)}, \mathbf{x}_t^{(i)} - \mathbf{x}^\star \right\rangle \\
\leq& \frac{D_1^2}{2\eta} + \frac{2L\eta}{M}\sum_{i=1}^M\sum_{t=1}^T t^2\Delta_t^{(i)} + \frac{\eta}{M}\sum_{i=1}^M\sum_{t=1}^T t^2\left\|\boldsymbol{\epsilon}_t^{(i)}\right\|^2
\end{aligned}$$

where $D_1 := \|\bar{\mathbf{x}}_1 - \mathbf{x}^\star\|$.

Proof of Lemma D.1. Fix an arbitrary worker $i \in [M]$. By the update rule for $t > 0$ that is not a synchronization step:

$$\mathbf{x}_{t+1}^{(i)} = \mathbf{x}_t^{(i)} - \eta t \mathbf{d}_t^{(i)}$$

Thus:

$$\left\|\mathbf{x}_{t+1}^{(i)}-\mathbf{x}^{\star}\right\|^2=\left\|\mathbf{x}_t^{(i)}-\eta t\mathbf{d}_t^{(i)}-\mathbf{x}^{\star}\right\|^2$$
$$=\left\|\mathbf{x}_t^{(i)}-\mathbf{x}^{\star}\right\|^2-2\eta t\left\langle\mathbf{d}_t^{(i)},\mathbf{x}_t^{(i)}-\mathbf{x}^{\star}\right\rangle+\eta^2t^2\left\|\mathbf{d}_t^{(i)}\right\|^2$$

Rearranging it yields:

$$2\eta t\left\langle\mathbf{d}_t^{(i)},\mathbf{x}_t^{(i)}-\mathbf{x}^{\star}\right\rangle=\left\|\mathbf{x}_t^{(i)}-\mathbf{x}^{\star}\right\|^2-\left\|\mathbf{x}_{t+1}^{(i)}-\mathbf{x}^{\star}\right\|^2+\eta^2t^2\left\|\mathbf{d}_t^{(i)}\right\|^2 \tag{12}$$

For a synchronization step $t>0$, the update rule is defined as:

$$\mathbf{x}_{t+1}^{(i)}=\mathbf{x}_t-\eta t\mathbf{d}_t^{(i)}$$

Therefore, similarly to the analysis for Equation (12), we have for a synchronization step $t>0$:

$$2\eta t\left\langle\mathbf{d}_t^{(i)},\mathbf{x}_t-\mathbf{x}^{\star}\right\rangle=\left\|\mathbf{x}_t-\mathbf{x}^{\star}\right\|^2-\left\|\mathbf{x}_{t+1}^{(i)}-\mathbf{x}^{\star}\right\|^2+\eta^2t^2\left\|\mathbf{d}_t^{(i)}\right\|^2 \tag{13}$$

Consider round r. Let s_r and e_r denote its first and last steps, respectively. At steps $\{s_r\}_{r=1}^R$, the model is synchronized, meaning that $\mathbf{x}_{s_r}^{(i)}=\mathbf{x}_{s_r}$ for all $i\in[M]$, while $\{e_r\}_{r=1}^R$ are the synchronization steps that follow Equation (13), where $\mathbf{d}_{e_r}^{(i)}=\mathbf{d}_{e_r}$.

We now sum Equation (12) over the iterations belonging to round r, over $t=s_r,\dots,e_r-1$:

$$2\eta\sum_{t=s_r}^{e_r-1}t\left\langle\mathbf{d}_t^{(i)},\mathbf{x}_t^{(i)}-\mathbf{x}^{\star}\right\rangle=\sum_{t=s_r}^{e_r-1}\left(\left\|\mathbf{x}_t^{(i)}-\mathbf{x}^{\star}\right\|^2-\left\|\mathbf{x}_{t+1}^{(i)}-\mathbf{x}^{\star}\right\|^2\right)+\eta^2\sum_{t=s_r}^{e_r-1}t^2\left\|\mathbf{d}_t^{(i)}\right\|^2$$
$$=\left\|\mathbf{x}_{s_r}^{(i)}-\mathbf{x}^{\star}\right\|^2-\left\|\mathbf{x}_{e_r}^{(i)}-\mathbf{x}^{\star}\right\|^2+\eta^2\sum_{t=s_r}^{e_r-1}t^2\left\|\mathbf{d}_t^{(i)}\right\|^2 \tag{14}$$

Adding the synchronization step of Equation (13) at $t=e_r$ to Equation (14):

$$2\eta\left(\sum_{t=s_r}^{e_r-1}t\left\langle\mathbf{d}_t^{(i)},\mathbf{x}_t^{(i)}-\mathbf{x}^{\star}\right\rangle+e_r\left\langle\mathbf{d}_{e_r}^{(i)},\mathbf{x}_{e_r}-\mathbf{x}^{\star}\right\rangle\right)$$
$$\leq\left\|\mathbf{x}_{s_r}^{(i)}-\mathbf{x}^{\star}\right\|^2-\left\|\mathbf{x}_{e_r+1}^{(i)}-\mathbf{x}^{\star}\right\|^2+\left\|\mathbf{x}_{e_r}-\mathbf{x}^{\star}\right\|^2-\left\|\mathbf{x}_{e_r}^{(i)}-\mathbf{x}^{\star}\right\|^2+\eta^2\sum_{t=s_r}^{e_r}t^2\left\|\mathbf{d}_t^{(i)}\right\|^2$$
$$=\left\|\mathbf{x}_{s_r}-\mathbf{x}^{\star}\right\|^2-\left\|\mathbf{x}_{s_{r+1}}-\mathbf{x}^{\star}\right\|^2+\left\|\mathbf{x}_{e_r}-\mathbf{x}^{\star}\right\|^2-\left\|\mathbf{x}_{e_r}^{(i)}-\mathbf{x}^{\star}\right\|^2+\eta^2\sum_{t=s_r}^{e_r}t^2\left\|\mathbf{d}_t^{(i)}\right\|^2$$

where in the final equality we used $e_r+1=s_{r+1}$ and $\mathbf{x}_{s_r}^{(i)}=\mathbf{x}_{s_r}$.

We average over all M workers and obtain:

$$\frac{2\eta}{M}\sum_{i=1}^{M}\sum_{t=s_r}^{e_r}t\left\langle\mathbf{d}_t^{(i)},\mathbf{x}_t^{(i)}-\mathbf{x}^{\star}\right\rangle\leq\left\|\mathbf{x}_{s_r}-\mathbf{x}^{\star}\right\|^2-\left\|\mathbf{x}_{s_{r+1}}-\mathbf{x}^{\star}\right\|^2+\frac{\eta^2}{M}\sum_{i=1}^{M}\sum_{t=s_r}^{e_r}t^2\left\|\mathbf{d}_t^{(i)}\right\|^2$$

where we used $\mathbf{x}_{e_r}=\frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_{e_r}^{(i)}$ and $\mathbf{d}_{e_r}^{(i)}=\mathbf{d}_{e_r}$, which yield $\frac{1}{M}\sum_{i=1}^{M}\left\langle\mathbf{d}_{e_r}^{(i)},\mathbf{x}_{e_r}-\mathbf{x}^{\star}\right\rangle=\frac{1}{M}\sum_{i=1}^{M}\left\langle\mathbf{d}_{e_r}^{(i)},\mathbf{x}_{e_r}^{(i)}-\mathbf{x}^{\star}\right\rangle$ and $\left\|\mathbf{x}_{e_r}-\mathbf{x}^{\star}\right\|^2\leq\frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{x}_{e_r}^{(i)}-\mathbf{x}^{\star}\right\|^2$.

Next, summing over $r=1,\dots,R_T$, where e_{R_T} is the last synchronization iteration in $[T]$, gives:

$$\frac{2\eta}{M}\sum_{i=1}^{M}\sum_{r=1}^{R_T}\sum_{t=s_r}^{e_r}t\left\langle\mathbf{d}_t^{(i)},\mathbf{x}_t^{(i)}-\mathbf{x}^{\star}\right\rangle$$
$$\leq\sum_{r=1}^{R_T}\left(\left\|\mathbf{x}_{s_r}-\mathbf{x}^{\star}\right\|^2-\left\|\mathbf{x}_{s_{r+1}}-\mathbf{x}^{\star}\right\|^2\right)+\frac{\eta^2}{M}\sum_{i=1}^{M}\sum_{r=1}^{R_T}\sum_{t=s_r}^{e_r}t^2\left\|\mathbf{d}_t^{(i)}\right\|^2$$
$$=\left\|\mathbf{x}_{s_1}-\mathbf{x}^{\star}\right\|^2-\left\|\mathbf{x}_{s_{R_T+1}}-\mathbf{x}^{\star}\right\|^2+\frac{\eta^2}{M}\sum_{i=1}^{M}\sum_{r=1}^{R_T}\sum_{t=s_r}^{e_r}t^2\left\|\mathbf{d}_t^{(i)}\right\|^2$$

The intervals $\{s_r, \ldots, e_r\}_{r=1}^{R_T}$ partition $\{1, \ldots, T-\tau\}$, and thus $\sum_{r=1}^{R_T} \sum_{t=s_r}^{e_r}(\cdot) = \sum_{t=1}^{T-\tau}(\cdot)$, where $\tau \geq 0$ is the number of extra iterations after the last synchronization.

Substituting $s_1 = 1$ and $s_{R_T+1} = e_{R_T} + 1 = T - \tau + 1$ gives:

$$\frac{2\eta}{M}\sum_{i=1}^{M}\sum_{t=1}^{T-\tau} t\left\langle \mathbf{d}_t^{(i)}, \mathbf{x}_t^{(i)} - \mathbf{x}^\star \right\rangle \leq \|\mathbf{x}_1 - \mathbf{x}^\star\|^2 - \|\mathbf{x}_{T-\tau+1} - \mathbf{x}^\star\|^2 + \frac{\eta^2}{M}\sum_{i=1}^{M}\sum_{t=1}^{T-\tau} t^2 \left\|\mathbf{d}_t^{(i)}\right\|^2$$

Adding the average over $i \in [M]$ of Equation (12) for $t = T - \tau + 1, \ldots, T$ gives:

$$\frac{2\eta}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} t\left\langle \mathbf{d}_t^{(i)}, \mathbf{x}_t^{(i)} - \mathbf{x}^\star \right\rangle \leq \|\mathbf{x}_1 - \mathbf{x}^\star\|^2 - \frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{x}_{T+1}^{(i)} - \mathbf{x}^\star\right\|^2 + \frac{\eta^2}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} t^2 \left\|\mathbf{d}_t^{(i)}\right\|^2$$

Using $\|\mathbf{x}_1 - \mathbf{x}^\star\| = \|\bar{\mathbf{x}}_1 - \mathbf{x}^\star\| = D_1$, and the bound $\|\mathbf{a}+\mathbf{b}\|^2 \leq 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$ with the identity:

$$\mathbf{d}_t^{(i)} - \boldsymbol{\epsilon}_t^{(i)} = \begin{cases} \nabla f\left(\bar{\mathbf{x}}_t^{(i)}\right), & \text{if } t \notin \mathcal{S}_T \\ \nabla f(\bar{\mathbf{x}}_t), & \text{if } t \in \mathcal{S}_T \end{cases}$$

yields:

$$\begin{aligned}
&\frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{x}_{T+1}^{(i)} - \mathbf{x}^\star\right\|^2 + \frac{2\eta}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} t\left\langle \mathbf{d}_t^{(i)}, \mathbf{x}_t^{(i)} - \mathbf{x}^\star \right\rangle \\
\leq& D_1^2 + \frac{2\eta^2}{M}\sum_{i=1}^{M}\left(\sum_{t\in[T]\setminus\mathcal{S}_T} t^2\left\|\nabla f\left(\bar{\mathbf{x}}_t^{(i)}\right)\right\|^2 + \sum_{t\in\mathcal{S}_T} t^2\left\|\nabla f(\bar{\mathbf{x}}_t)\right\|^2\right) + \frac{2\eta^2}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} t^2\left\|\boldsymbol{\epsilon}_t^{(i)}\right\|^2
\end{aligned}$$

Dividing by 2η and using Lemma A.1, we get:

$$\begin{aligned}
&\frac{1}{2\eta M}\sum_{i=1}^{M}\left\|\mathbf{x}_{T+1}^{(i)} - \mathbf{x}^\star\right\|^2 + \frac{1}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} t\left\langle \mathbf{d}_t^{(i)}, \mathbf{x}_t^{(i)} - \mathbf{x}^\star \right\rangle \\
\leq& \frac{D_1^2}{2\eta} + \frac{2L\eta}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} t^2\Delta_t^{(i)} + \frac{\eta}{M}\sum_{i=1}^{M}\sum_{t=1}^{T} t^2\left\|\boldsymbol{\epsilon}_t^{(i)}\right\|^2
\end{aligned}$$

□

Lemma D.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex, L-smooth function with global minimizer $\mathbf{x}^\star$, and let $\left\{\mathbf{x}_t^{(i)}\right\}_{t\geq 1, i\in[M]}$ be the iterates generated by Algorithm 3 with $\gamma_t = \frac{2}{t+2}, \eta_t = t\cdot\eta$, where $\eta \leq \frac{1}{8LT}$. Then, for any $0 < t \leq T$:*

$$\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 \leq \frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{x}_{t+1}^{(i)} - \mathbf{x}^\star\right\|^2 \leq \tilde{D}_t^2$$

$$\|\bar{\mathbf{x}}_{t+1} - \mathbf{x}^\star\|^2 \leq \frac{1}{M}\sum_{i=1}^{M}\left\|\bar{\mathbf{x}}_{t+1}^{(i)} - \mathbf{x}^\star\right\|^2 \leq \tilde{D}_t^2$$

where $D_1 := \|\bar{\mathbf{x}}_1 - \mathbf{x}^\star\|$ and $\tilde{D}_t^2 := 2D_1^2 + \frac{4\eta^2 t}{M}\sum_{i=1}^{M}\sum_{\tau=1}^{t}\tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$.

In addition,

$$\frac{1}{M}\sum_{i=1}^{M}\Delta_t^{(i)} \leq \frac{2D_1^2}{\eta t^2} + \frac{4\eta}{tM}\sum_{i=1}^{M}\sum_{\tau=1}^{t}\tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$$

Proof of Lemma D.2. The first inequality in each line follows from $\mathbf{x}_t = \frac{1}{M}\sum_{i=1}^M \mathbf{x}_t^{(i)}$ and $\bar{\mathbf{x}}_t = \frac{1}{M}\sum_{i=1}^M \bar{\mathbf{x}}_t^{(i)}$. We will focus on the second part. From Lemma D.1, we get:

$$\frac{1}{2\eta M}\sum_{i=1}^M \left\|\mathbf{x}_{t+1}^{(i)} - \mathbf{x}^\star\right\|^2 \le \frac{D_1^2}{2\eta} - \frac{1}{M}\sum_{i=1}^M\sum_{\tau=1}^t \tau\left\langle \mathbf{d}_\tau^{(i)}, \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\rangle$$
$$+\frac{2L\eta}{M}\sum_{i=1}^M\sum_{\tau=1}^t \tau^2\Delta_\tau^{(i)} + \frac{\eta}{M}\sum_{i=1}^M\sum_{\tau=1}^t \tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$$

Using the definition of $\boldsymbol{\epsilon}_\tau^{(i)}$, we split the negative term into two terms:

$$\frac{1}{M}\sum_{i=1}^M\sum_{\tau=1}^t \tau\left\langle \mathbf{d}_\tau^{(i)}, \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\rangle = \frac{1}{M}\sum_{i=1}^M\sum_{\tau=1}^t \tau\left\langle \boldsymbol{\epsilon}_\tau^{(i)}, \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\rangle$$
$$+\frac{1}{M}\sum_{i=1}^M\left(\sum_{\tau\in[t]\setminus\mathcal{S}_t} \tau\left\langle \nabla f\left(\bar{\mathbf{x}}_\tau^{(i)}\right), \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\rangle + \sum_{\tau\in\mathcal{S}_t}\tau\left\langle \nabla f(\bar{\mathbf{x}}_\tau), \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\rangle\right)$$

The first term can be bounded by Young's inequality $2\langle \mathbf{a}, \mathbf{b}\rangle \le \lambda\|\mathbf{a}\|^2 + \lambda^{-1}\|\mathbf{b}\|^2$ with $\mathbf{a} = \tau\boldsymbol{\epsilon}_\tau^{(i)}, \mathbf{b} = \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star, \lambda = \eta(2T-1)$:

$$-\frac{1}{M}\sum_{i=1}^M\sum_{\tau=1}^t \tau\left\langle \boldsymbol{\epsilon}_\tau^{(i)}, \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\rangle \le \frac{\eta(2T-1)}{2M}\sum_{i=1}^M\sum_{\tau=1}^t \tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$$
$$+\frac{1}{2\eta(2T-1)M}\sum_{i=1}^M\sum_{\tau=1}^t\left\|\mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\|^2$$

The second term can be bounded by Lemma C.3:

$$-\frac{1}{M}\sum_{i=1}^M\left(\sum_{\tau\in[t]\setminus\mathcal{S}_t} \tau\left\langle \nabla f\left(\bar{\mathbf{x}}_\tau^{(i)}\right), \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\rangle + \sum_{\tau\in\mathcal{S}_t}\tau\left\langle \nabla f(\bar{\mathbf{x}}_\tau), \mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\rangle\right)$$
$$\le -\frac{t(t+1)}{2M}\sum_{i=1}^M \Delta_t^{(i)}$$

Finally, we get:

$$\frac{1}{2M}\sum_{i=1}^M t(t+1)\Delta_t^{(i)} + \frac{1}{2\eta M}\sum_{i=1}^M\left\|\mathbf{x}_{t+1}^{(i)} - \mathbf{x}^\star\right\|^2 \le \frac{D_1^2}{2\eta} + \frac{\eta(2T+1)}{2M}\sum_{i=1}^M\sum_{\tau=1}^t \tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$$
$$+\frac{2L\eta}{M}\sum_{i=1}^M\sum_{\tau=1}^t \tau(\tau+1)\Delta_\tau^{(i)} + \frac{1}{2\eta(2T-1)M}\sum_{i=1}^M\sum_{\tau=1}^t\left\|\mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\|^2$$

Adding $\frac{1}{2\eta(2T-1)M}\sum_{i=1}^M\left\|\bar{\mathbf{x}}_{t+1}^{(i)} - \mathbf{x}^\star\right\|^2$ to both sides, using $D_1 := \|\mathbf{x}_1 - \mathbf{x}^\star\| = \left\|\mathbf{x}_1^{(i)} - \mathbf{x}^\star\right\|$ and $2L\eta \le \frac{1}{4T}$, and extending the sums to T, we obtain:

$$\frac{1}{2M}\sum_{i=1}^M t(t+1)\Delta_t^{(i)} + \frac{T}{\eta(2T-1)M}\sum_{i=1}^M\left\|\mathbf{x}_{t+1}^{(i)} - \mathbf{x}^\star\right\|^2$$
$$\le \frac{D_1^2 T}{\eta(2T-1)} + \frac{\eta(2T+1)}{2M}\sum_{i=1}^M\sum_{\tau=1}^t \tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$$
$$+\frac{1}{4TM}\sum_{i=1}^M\sum_{\tau=1}^T \tau(\tau+1)\Delta_\tau^{(i)} + \frac{1}{2\eta(2T-1)M}\sum_{i=1}^M\sum_{\tau=1}^T\left\|\mathbf{x}_{\tau+1}^{(i)} - \mathbf{x}^\star\right\|^2$$

We can now apply Lemma C.1 for $t \le T$:

$$\frac{t(t+1)}{2M}\sum_{i=1}^{M}\Delta_t^{(i)} + \frac{T}{\eta(2T-1)M}\sum_{i=1}^{M}\left\|\mathbf{x}_{t+1}^{(i)} - \mathbf{x}^\star\right\|^2$$
$$\le \frac{2D_1^2 T}{\eta(2T-1)} + \frac{\eta(2T+1)}{M}\sum_{i=1}^{M}\sum_{\tau=1}^{T}\tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$$

Setting $T = t$ and multiplying by $\eta\frac{2t-1}{t}$, we obtain:

$$\frac{\eta(t+1)(2t-1)}{2M}\sum_{i=1}^{M}\Delta_t^{(i)} + \frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{x}_{t+1}^{(i)} - \mathbf{x}^\star\right\|^2$$
$$\le 2D_1^2 + \frac{\eta^2(2t+1)(2t-1)}{tM}\sum_{i=1}^{M}\sum_{\tau=1}^{t}\tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$$

Since $(t+1)(2t-1) = 2t^2 + t - 1 \ge 2t^2$ and $(2t+1)(2t-1) = 4t^2 - 1 \le 4t^2$:

$$\frac{\eta t^2}{M}\sum_{i=1}^{M}\Delta_t^{(i)} + \frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{x}_{t+1}^{(i)} - \mathbf{x}^\star\right\|^2 \le 2D_1^2 + \frac{4\eta^2 t}{M}\sum_{i=1}^{M}\sum_{\tau=1}^{t}\tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$$

Next, we use $\Delta_t^{(i)} \ge 0$:

$$\frac{1}{M}\sum_{i=1}^{M}\left\|\mathbf{x}_{t+1}^{(i)} - \mathbf{x}^\star\right\|^2 \le 2D_1^2 + \frac{4\eta^2 t}{M}\sum_{i=1}^{M}\sum_{\tau=1}^{t}\tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$$

The bound on $\bar{\mathbf{x}}_{t+1}^{(i)}$ follows from the bound on $\mathbf{x}_{t+1}^{(i)}$ and $\mathbf{x}_{t+1}$ because $\bar{\mathbf{x}}_{t+1}^{(i)}$ is a weighted average of such terms.

Using the nonnegativity of the norm and dividing by ηt^2, we also obtain:

$$\frac{1}{M}\sum_{i=1}^{M}\Delta_t^{(i)} \le \frac{2D_1^2}{\eta t^2} + \frac{4\eta}{tM}\sum_{i=1}^{M}\sum_{\tau=1}^{t}\tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$$

□

Lemma D.3. *Let $\{\mathbf{x}_t\}_{t\ge1} \subset \mathbb{R}^d$ and $\{\bar{\mathbf{x}}_t\}_{t\ge1} \subset \mathbb{R}^d$ be such that $\bar{\mathbf{x}}_t$ is the $\{\alpha_t \ge 0\}_{t\ge1}$-weighted average of $\{\mathbf{x}_t\}_{t\ge1}$. Then running Algorithm 3 with $\gamma_t = \frac{2}{t+2}, \eta_t = t\cdot\eta$, where $\eta \le \frac{1}{8LT}$, yields:*

$$\frac{1}{M}\sum_{i=1}^{M}\left\|\bar{\mathbf{x}}_t^{(i)} - \bar{\mathbf{x}}_t\right\|^2 \le \frac{4K_{\mathrm{loc}}^2\tilde{D}_{t-1}^2}{t^2}$$

where $D_1 := \|\bar{\mathbf{x}}_1 - \mathbf{x}^\star\|$ and $\tilde{D}_t^2 := 2D_1^2 + \frac{4\eta^2 t}{M}\sum_{i=1}^{M}\sum_{\tau=1}^{t}\tau^2\left\|\boldsymbol{\epsilon}_\tau^{(i)}\right\|^2$.

Proof of Lemma D.3. Define $\alpha_t := t$ and $\alpha_{1:t} := \sum_{\tau=1}^{t}\alpha_\tau$; then $\gamma_t = \frac{\alpha_{t+1}}{\alpha_{1:t+1}} = \frac{2}{t+2}$. Letting t_0 denote the last synchronization step, we can write $\alpha_{1:t}\bar{\mathbf{x}}_t^{(i)} = \alpha_{1:t_0+1}\bar{\mathbf{x}}_{t_0+1}^{(i)} + \sum_{\tau=t_0+2}^{t}\alpha_\tau\mathbf{x}_\tau^{(i)}$. Similarly, $\alpha_{1:t}\bar{\mathbf{x}}_t = \alpha_{1:t_0+1}\bar{\mathbf{x}}_{t_0+1} + \sum_{\tau=t_0+2}^{t}\alpha_\tau\mathbf{x}_\tau$. Note that $\bar{\mathbf{x}}_{t_0+1}^{(i)} = \bar{\mathbf{x}}_{t_0+1}$. It follows that:

$$\bar{\mathbf{x}}_t^{(i)} - \bar{\mathbf{x}}_t = \frac{1}{\alpha_{1:t}}\sum_{\tau=t_0+2}^{t}\alpha_\tau\left(\mathbf{x}_\tau^{(i)} - \mathbf{x}_\tau\right) = \frac{2}{t(t+1)}\sum_{\tau=t_0+2}^{t}\tau\left(\mathbf{x}_\tau^{(i)} - \mathbf{x}_\tau\right)$$

Bounding:

$$\frac{1}{M}\sum_{i=1}^{M}\left\|\bar{\mathbf{x}}_t^{(i)}-\bar{\mathbf{x}}_t\right\|^2=\frac{4}{t^2(t+1)^2M}\sum_{i=1}^{M}\left\|\sum_{\tau=t_0+2}^{t}\tau\left(\mathbf{x}_\tau^{(i)}-\mathbf{x}_\tau\right)\right\|^2$$
$$\leq\frac{4\left(\sum_{\tau=t_0+2}^{t}\tau\right)}{t^4M}\sum_{i=1}^{M}\sum_{\tau=t_0+2}^{t}\tau\left\|\mathbf{x}_\tau^{(i)}-\mathbf{x}_\tau\right\|^2$$
$$\leq\frac{4\left(\sum_{\tau=t_0+2}^{t}\tau\right)}{t^4M}\sum_{i=1}^{M}\sum_{\tau=t_0+2}^{t}\tau\left\|\mathbf{x}_\tau^{(i)}-\mathbf{x}^\star\right\|^2$$

where we used Jensen's inequality and then $\mathbb{E}\left\|X-\mathbb{E}\left[X\right]\right\|^2\leq\mathbb{E}\left\|X\right\|^2$, with $X=\mathbf{x}_\tau^{(i)}-\mathbf{x}^\star$ and the expectation replaced by averaging over i. We can now use Lemma D.2 to obtain:

$$\frac{1}{M}\sum_{i=1}^{M}\left\|\bar{\mathbf{x}}_t^{(i)}-\bar{\mathbf{x}}_t\right\|^2\leq\left(\frac{2\sum_{\tau=t_0+2}^{t}\tau}{t^2}\right)^2\tilde{D}_{t-1}^2$$

We bound $\sum_{\tau=t_0+2}^{t}\tau\leq(t-t_0-1)t$, and $t-t_0-1\leq K_{\text{loc}}$.

$$\frac{1}{M}\sum_{i=1}^{M}\left\|\bar{\mathbf{x}}_t^{(i)}-\bar{\mathbf{x}}_t\right\|^2\leq\frac{4K_{\text{loc}}^2\tilde{D}_{t-1}^2}{t^2}$$

□

E Error Bound

Lemma E.1 (Stochastic error decomposition)**.** *Assume Algorithm 1 uses $\beta_t=\frac{1}{t}$. Fix any $t>0$, and let t_0 be the last synchronization step before or at time t. For each worker i, define*

$$\mathbf{v}_\tau^{(i)}:=\begin{cases}\frac{1}{K_{\text{avg}}}\sum_{k=1}^{K_{\text{avg}}}\left(\nabla f\left(\bar{\mathbf{x}}_\tau;\mathbf{z}_\tau^{(i,k)}\right)-\nabla f(\bar{\mathbf{x}}_\tau)\right), & \tau\in\mathcal{S}_T\\ \nabla f\left(\bar{\mathbf{x}}_\tau^{(i)};\mathbf{z}_\tau^{(i)}\right)-\nabla f\left(\bar{\mathbf{x}}_\tau^{(i)}\right), & \tau\notin\mathcal{S}_T\end{cases}$$
$$\tilde{\mathbf{v}}_{\tau-1}^{(i)}:=\begin{cases}\frac{1}{K_{\text{avg}}}\sum_{k=1}^{K_{\text{avg}}}\left(\nabla f\left(\bar{\mathbf{x}}_{\tau-1}^{(i)};\mathbf{z}_\tau^{(i,k)}\right)-\nabla f\left(\bar{\mathbf{x}}_{\tau-1}^{(i)}\right)\right), & \tau\in\mathcal{S}_T\\ \nabla f\left(\bar{\mathbf{x}}_{\tau-1};\mathbf{z}_\tau^{(i)}\right)-\nabla f(\bar{\mathbf{x}}_{\tau-1}), & \tau-1\in\mathcal{S}_T\\ \nabla f\left(\bar{\mathbf{x}}_{\tau-1}^{(i)};\mathbf{z}_\tau^{(i)}\right)-\nabla f\left(\bar{\mathbf{x}}_{\tau-1}^{(i)}\right), & \textit{otherwise}\end{cases}$$

Then $\{\tau\mathbf{v}_\tau^{(i)}-(\tau-1)\tilde{\mathbf{v}}_{\tau-1}^{(i)}\}_\tau$ is a martingale difference sequence, and

$$t^2\mathbb{E}\left\|\boldsymbol{\epsilon}_t^{(i)}\right\|^2=\frac{1}{M^2}\sum_{j=1}^{M}\sum_{\tau=1}^{t_0}\mathbb{E}\left\|\tau\mathbf{v}_\tau^{(j)}-(\tau-1)\tilde{\mathbf{v}}_{\tau-1}^{(j)}\right\|^2+\sum_{\tau=t_0+1}^{t}\mathbb{E}\left\|\tau\mathbf{v}_\tau^{(i)}-(\tau-1)\tilde{\mathbf{v}}_{\tau-1}^{(i)}\right\|^2$$

Lemma E.1 establishes that the local stochastic error $\boldsymbol{\epsilon}_t^{(i)}$ integrates noise across **all preceding iterations**, but its variance is not distributed uniformly. Specifically, noise generated prior to the most recent synchronization step t_0 has been averaged across all M workers, effectively scaling its variance by a factor of $\frac{1}{M}$. In contrast, the stochasticity accumulated within the current round spans at most K_{loc} iterations and is restricted to the stochasticity of one gradient.

Proof of Lemma E.1. We examine the recursion of $\boldsymbol{\epsilon}_t^{(i)}$ in all three cases.

Synchronization step t_0. Let t_0 denote the most recent synchronization step before iteration t. At synchronization, all workers share the same estimator, and therefore:

$$\mathbf{d}_{t_0}^{(i)} = \mathbf{d}_{t_0} = (1-\beta_{t_0})\mathbf{d}_{t_0-1} + \frac{1}{K_{\text{avg}}M}\sum_{j=1}^{M}\sum_{k=1}^{K_{\text{avg}}}\left(\nabla f\left(\bar{\mathbf{x}}_{t_0};\mathbf{z}_{t_0}^{(j,k)}\right) - (1-\beta_{t_0})\nabla f\left(\bar{\mathbf{x}}_{t_0-1}^{(j)};\mathbf{z}_{t_0}^{(j,k)}\right)\right)$$

where $\mathbf{d}_{t_0} := \frac{1}{M}\sum_{j=1}^{M}\tilde{\mathbf{d}}_{t_0}^{(j)}$ and $\mathbf{d}_{t_0-1} := \frac{1}{M}\sum_{j=1}^{M}\mathbf{d}_{t_0-1}^{(j)}$. Choosing $\beta_t = \frac{1}{t}$ yields:

$$t_0\mathbf{d}_{t_0}^{(i)} = \frac{1}{M}\sum_{j=1}^{M}(t_0-1)\mathbf{d}_{t_0-1}^{(j)} + \frac{1}{K_{\text{avg}}M}\sum_{j=1}^{M}\sum_{k=1}^{K_{\text{avg}}}\left(t_0\nabla f\left(\bar{\mathbf{x}}_{t_0};\mathbf{z}_{t_0}^{(j,k)}\right) - (t_0-1)\nabla f\left(\bar{\mathbf{x}}_{t_0-1}^{(j)};\mathbf{z}_{t_0}^{(j,k)}\right)\right)$$

Subtracting $t_0\nabla f(\bar{\mathbf{x}}_{t_0})$ from both sides, and adding and subtracting $\frac{1}{M}\sum_{j=1}^{M}(t_0-1)\nabla f\left(\bar{\mathbf{x}}_{t_0-1}^{(j)}\right)$, we obtain:

$$t_0\boldsymbol{\epsilon}_{t_0}^{(i)} = \frac{1}{M}\sum_{j=1}^{M}(t_0-1)\boldsymbol{\epsilon}_{t_0-1}^{(j)} + \frac{1}{M}\sum_{j=1}^{M}\left(t_0\mathbf{v}_{t_0}^{(j)} - (t_0-1)\tilde{\mathbf{v}}_{t_0-1}^{(j)}\right) \tag{15}$$

where

$$\mathbf{v}_{t_0}^{(i)} := \frac{1}{K_{\text{avg}}}\sum_{k=1}^{K_{\text{avg}}}\nabla f\left(\bar{\mathbf{x}}_{t_0};\mathbf{z}_{t_0}^{(i,k)}\right) - \nabla f(\bar{\mathbf{x}}_{t_0})$$

$$\tilde{\mathbf{v}}_{t_0-1}^{(i)} := \frac{1}{K_{\text{avg}}}\sum_{k=1}^{K_{\text{avg}}}\nabla f\left(\bar{\mathbf{x}}_{t_0-1}^{(i)};\mathbf{z}_{t_0}^{(i,k)}\right) - \nabla f\left(\bar{\mathbf{x}}_{t_0-1}^{(i)}\right)$$

First step after synchronization: t_0+1. Applying the same argument for the next step we get:

$$(t_0+1)\mathbf{d}_{t_0+1}^{(i)} = t_0\mathbf{d}_{t_0}^{(i)} + (t_0+1)\nabla f\left(\bar{\mathbf{x}}_{t_0+1}^{(i)};\mathbf{z}_{t_0+1}^{(i)}\right) - t_0\nabla f\left(\bar{\mathbf{x}}_{t_0};\mathbf{z}_{t_0+1}^{(i)}\right)$$

Subtracting $(t_0+1)\nabla f\left(\bar{\mathbf{x}}_{t_0+1}^{(i)}\right)$ and adding and subtracting $t_0\nabla f(\bar{\mathbf{x}}_{t_0})$, we obtain:

$$(t_0+1)\boldsymbol{\epsilon}_{t_0+1}^{(i)} = t_0\boldsymbol{\epsilon}_{t_0}^{(i)} + \left((t_0+1)\mathbf{v}_{t_0+1}^{(i)} - t_0\tilde{\mathbf{v}}_{t_0}^{(i)}\right) \tag{16}$$

where

$$\mathbf{v}_{t_0+1}^{(i)} := \nabla f\left(\bar{\mathbf{x}}_{t_0+1}^{(i)};\mathbf{z}_{t_0+1}^{(i)}\right) - \nabla f\left(\bar{\mathbf{x}}_{t_0+1}^{(i)}\right), \quad \tilde{\mathbf{v}}_{t_0}^{(i)} := \nabla f\left(\bar{\mathbf{x}}_{t_0};\mathbf{z}_{t_0+1}^{(i)}\right) - \nabla f(\bar{\mathbf{x}}_{t_0})$$

Non-synchronization steps after the first post-synchronization iteration. Now consider a non-synchronization step t after the first post-synchronization iteration:

$$t\boldsymbol{\epsilon}_t^{(i)} = (t-1)\boldsymbol{\epsilon}_{t-1}^{(i)} + \left(t\mathbf{v}_t^{(i)} - (t-1)\tilde{\mathbf{v}}_{t-1}^{(i)}\right) \tag{17}$$

Here:

$$\mathbf{v}_t^{(i)} := \nabla f\left(\bar{\mathbf{x}}_t^{(i)};\mathbf{z}_t^{(i)}\right) - \nabla f\left(\bar{\mathbf{x}}_t^{(i)}\right), \quad \tilde{\mathbf{v}}_{t-1}^{(i)} := \nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)};\mathbf{z}_t^{(i)}\right) - \nabla f\left(\bar{\mathbf{x}}_{t-1}^{(i)}\right)$$

Unrolling Equation (17) up to t_0+2 and adding Equation (16) yields:

$$t\boldsymbol{\epsilon}_t^{(i)} = t_0\boldsymbol{\epsilon}_{t_0}^{(i)} + \sum_{\tau=t_0+1}^{t}\left(\tau\mathbf{v}_\tau^{(i)} - (\tau-1)\tilde{\mathbf{v}}_{\tau-1}^{(i)}\right) \tag{18}$$

This result is true for all non-synchronization steps.

Using Equation (15) at the next synchronization step t_1, together with Equation (18) at $t = t_1 - 1$:

$$t_1 \boldsymbol{\epsilon}_{t_1}^{(i)} = \frac{1}{M} \sum_{j=1}^{M} t_0 \boldsymbol{\epsilon}_{t_0}^{(j)} + \frac{1}{M} \sum_{j=1}^{M} \sum_{\tau=t_0+1}^{t_1} \left(\tau \mathbf{v}_\tau^{(j)} - (\tau - 1) \tilde{\mathbf{v}}_{\tau-1}^{(j)} \right)$$

Because t_0 is a synchronization step, $\boldsymbol{\epsilon}_{t_0}^{(j)} = \boldsymbol{\epsilon}_{t_0}^{(i)}$; thus,

$$t_1 \boldsymbol{\epsilon}_{t_1}^{(i)} = t_0 \boldsymbol{\epsilon}_{t_0}^{(i)} + \frac{1}{M} \sum_{j=1}^{M} \sum_{\tau=t_0+1}^{t_1} \left(\tau \mathbf{v}_\tau^{(j)} - (\tau - 1) \tilde{\mathbf{v}}_{\tau-1}^{(j)} \right)$$

Unrolling this result over all synchronization steps and substituting it into Equation (18), we obtain:

$$t \boldsymbol{\epsilon}_t^{(i)} = \frac{1}{M} \sum_{j=1}^{M} \sum_{\tau=1}^{t_0} \left(\tau \mathbf{v}_\tau^{(j)} - (\tau - 1) \tilde{\mathbf{v}}_{\tau-1}^{(j)} \right) + \sum_{\tau=t_0+1}^{t} \left(\tau \mathbf{v}_\tau^{(i)} - (\tau - 1) \tilde{\mathbf{v}}_{\tau-1}^{(i)} \right)$$

We can see that this is a martingale difference sequence, and thus we get:

$$t^2 \mathbb{E} \left\| \boldsymbol{\epsilon}_t^{(i)} \right\|^2 = \frac{1}{M^2} \sum_{j=1}^{M} \sum_{\tau=1}^{t_0} \mathbb{E} \left\| \tau \mathbf{v}_\tau^{(j)} - (\tau - 1) \tilde{\mathbf{v}}_{\tau-1}^{(j)} \right\|^2 + \sum_{\tau=t_0+1}^{t} \mathbb{E} \left\| \tau \mathbf{v}_\tau^{(i)} - (\tau - 1) \tilde{\mathbf{v}}_{\tau-1}^{(i)} \right\|^2$$

□

Proof of Lemma 2.1. We begin with Lemma E.1.

Bounding the stochastic terms. It remains to bound $\mathbb{E} \left\| \tau \mathbf{v}_\tau^{(i)} - (\tau - 1) \tilde{\mathbf{v}}_{\tau-1}^{(i)} \right\|^2$.

We consider three cases separately: (i) the first local step after synchronization, (ii) a later local step within the same round, and (iii) a synchronization step. In all cases, the same elementary decomposition applies:

$$\mathbb{E} \left\| \tau \mathbf{v}_\tau^{(i)} - (\tau - 1) \tilde{\mathbf{v}}_{\tau-1}^{(i)} \right\|^2 \leq 2 \underbrace{\mathbb{E} \left\| \mathbf{v}_\tau^{(i)} \right\|^2}_{\text{gradient variance}} + 2(\tau - 1)^2 \underbrace{\mathbb{E} \left\| \mathbf{v}_\tau^{(i)} - \tilde{\mathbf{v}}_{\tau-1}^{(i)} \right\|^2}_{\text{smoothness variance}}$$

The first term is the intrinsic stochastic variance of the gradient estimator, whereas the second term measures the change in the estimator between the two points.

Step 1: bounding the gradient variance term. The term $\mathbb{E} \left\| \mathbf{v}_\tau^{(i)} \right\|^2$ is directly bounded by the assumption in Equation (2). Summing over iterations gives

$$\frac{1}{M^2} \sum_{j=1}^{M} \sum_{\tau=1}^{t_0} \mathbb{E} \left\| \mathbf{v}_\tau^{(j)} \right\|^2 + \sum_{\tau=t_0+1}^{t} \mathbb{E} \left\| \mathbf{v}_\tau^{(i)} \right\|^2 \leq (t - t_0)\sigma^2 + \frac{(r-1)\sigma^2}{K_{\text{avg}} M} + \frac{(r-1) K_{\text{loc}} \sigma^2}{M}$$

Here, the first term corresponds to the current round, whose noise has not yet been averaged. The second term arises from synchronization steps, where averaging over both workers and the K_{avg} samples yields an additional reduction in variance. The third term arises from averaging all local updates prior to the last synchronization and across workers. Using $t_0 = (r-1)(K_{\text{loc}} + 1)$ and $K_{\text{avg}} \geq 1$, we obtain

$$\frac{1}{M^2} \sum_{j=1}^{M} \sum_{\tau=1}^{t_0} \mathbb{E} \left\| \mathbf{v}_\tau^{(j)} \right\|^2 + \sum_{\tau=t_0+1}^{t} \mathbb{E} \left\| \mathbf{v}_\tau^{(i)} \right\|^2 \leq (t - t_0)\sigma^2 + \frac{t_0 \sigma^2}{M} \tag{19}$$

Thus, only a limited amount of variance from the most recent steps $t - t_0 \leq K_{\text{loc}}$ remains local and unaveraged, while updates from earlier iterations have already been synchronized across workers and reduce their stochastic error by a factor of M.

Step 2: bounding the smoothness variance term. We now turn to $\mathbb{E}\left\|\mathbf{v}_\tau^{(i)} - \tilde{\mathbf{v}}_{\tau-1}^{(i)}\right\|^2$. By the assumption in Equation (4), this term is bounded by the distance between the two iterates at which the stochastic gradients are evaluated. More precisely,

$$\mathbb{E}\left\|\mathbf{v}_\tau^{(i)} - \tilde{\mathbf{v}}_{\tau-1}^{(i)}\right\|^2 \leq \begin{cases} \frac{\sigma_L^2}{K_{\text{avg}}}\mathbb{E}\left\|\bar{\mathbf{x}}_\tau - \bar{\mathbf{x}}_{\tau-1}^{(i)}\right\|^2, & \tau \in \mathcal{S}_T \\ \sigma_L^2\mathbb{E}\left\|\bar{\mathbf{x}}_\tau^{(i)} - \bar{\mathbf{x}}_{\tau-1}\right\|^2, & \tau - 1 \in \mathcal{S}_T \\ \sigma_L^2\mathbb{E}\left\|\bar{\mathbf{x}}_\tau^{(i)} - \bar{\mathbf{x}}_{\tau-1}^{(i)}\right\|^2, & \text{otherwise} \end{cases} \tag{20}$$

So in every case, the problem reduces to controlling the distance between iterates at that stage.

For non-synchronization steps $\tau, \tau - 1 \notin \mathcal{S}_T$, the anytime momentum update implies

$$\bar{\mathbf{x}}_\tau^{(i)} - \bar{\mathbf{x}}_{\tau-1}^{(i)} = \gamma_{\tau-1}\left(\mathbf{x}_\tau^{(i)} - \bar{\mathbf{x}}_{\tau-1}^{(i)}\right)$$

Using Lemma D.2, setting $\gamma_t = \frac{2}{t+2}$, and defining $\tilde{D}_\tau^2 := 2D_1^2 + \frac{4\eta^2\tau}{M}\sum_{i=1}^M\sum_{k=1}^\tau k^2\left\|\boldsymbol{\epsilon}_k^{(i)}\right\|^2$, we obtain

$$\begin{aligned} &\frac{1}{M}\sum_{i=1}^M\mathbb{E}\left\|\bar{\mathbf{x}}_\tau^{(i)} - \bar{\mathbf{x}}_{\tau-1}^{(i)}\right\|^2 = \frac{1}{M}\sum_{i=1}^M\gamma_{\tau-1}^2\mathbb{E}\left\|\mathbf{x}_\tau^{(i)} - \bar{\mathbf{x}}_{\tau-1}^{(i)}\right\|^2 \\ \leq &\frac{1}{M}\sum_{i=1}^M 2\gamma_{\tau-1}^2\left(\mathbb{E}\left\|\mathbf{x}_\tau^{(i)} - \mathbf{x}^\star\right\|^2 + \mathbb{E}\left\|\bar{\mathbf{x}}_{\tau-1}^{(i)} - \mathbf{x}^\star\right\|^2\right) \leq \frac{16\tilde{D}_{\tau-1}^2}{\tau^2} \end{aligned} \tag{21}$$

Similarly, for $\tau - 1 \in \mathcal{S}_T$ we have $\frac{1}{M}\sum_{i=1}^M\mathbb{E}\left\|\bar{\mathbf{x}}_\tau^{(i)} - \bar{\mathbf{x}}_{\tau-1}\right\|^2 \leq \frac{16\tilde{D}_{\tau-1}^2}{\tau^2}$.

The synchronization drift. The synchronization case is slightly different, as the iterates transition from the local point $\bar{\mathbf{x}}_{t_0-1}^{(i)}$ to the averaged global point $\bar{\mathbf{x}}_{t_0}$. Using $\frac{1}{M}\sum_{i=1}^M\bar{\mathbf{x}}_{t_0-1}^{(i)} = \bar{\mathbf{x}}_{t_0-1}$, we decompose:

$$\frac{1}{M}\sum_{i=1}^M\left\|\bar{\mathbf{x}}_{t_0} - \bar{\mathbf{x}}_{t_0-1}^{(i)}\right\|^2 = \left\|\bar{\mathbf{x}}_{t_0} - \bar{\mathbf{x}}_{t_0-1}\right\|^2 + \frac{1}{M}\sum_{i=1}^M\left\|\bar{\mathbf{x}}_{t_0-1}^{(i)} - \bar{\mathbf{x}}_{t_0-1}\right\|^2$$

The first term is handled exactly as in Equation (21).

For the second term, we employ Lemma D.3, which gives

$$\frac{1}{M}\sum_{i=1}^M\left\|\bar{\mathbf{x}}_{t_0-1}^{(i)} - \bar{\mathbf{x}}_{t_0-1}\right\|^2 \leq \frac{4K_{\text{loc}}^2\tilde{D}_{t_0-2}^2}{(t_0-1)^2}$$

Compared with Equation (21), this introduces an additional factor K_{loc}^2, reflecting the fact that disagreement can accumulate over an entire local round before being reset by synchronization. Multiplying each of these equations by $(\tau-1)^2$ gives:

$$\begin{aligned} &\frac{1}{M^2}\sum_{j=1}^M\sum_{\tau=1}^{t_0}(\tau-1)^2\mathbb{E}\left\|\mathbf{v}_\tau^{(j)} - \tilde{\mathbf{v}}_{\tau-1}^{(j)}\right\|^2 + \sum_{\tau=t_0+1}^{t}(\tau-1)^2\mathbb{E}\left\|\mathbf{v}_\tau^{(i)} - \tilde{\mathbf{v}}_{\tau-1}^{(i)}\right\|^2 \\ \leq &16(t-t_0)\sigma_L^2\tilde{D}_t^2 + \frac{(r-1)\sigma_L^2\tilde{D}_t^2(4K_{\text{loc}}^2+16)}{K_{\text{avg}}M} + \frac{16(r-1)K_{\text{loc}}\sigma_L^2\tilde{D}_t^2}{M} \\ \leq &16\sigma_L^2\tilde{D}_t^2\left((t-t_0) + \frac{5(r-1)(K_{\text{loc}}+1)}{4M}\right) = 16\sigma_L^2\tilde{D}_t^2\left((t-t_0) + \frac{5t_0}{4M}\right) \end{aligned} \tag{22}$$

where in the last inequality we used $\alpha = \frac{1}{2}$ to bound $\frac{K_{\text{loc}}^2+4}{4K_{\text{avg}}} \leq \frac{K_{\text{loc}}+5}{4}$. We also used the fact that $\tilde{D}_\tau$ is nondecreasing in τ to bound all such terms by $\tilde{D}_t$.

Combining Equations (19) and (22), we obtain the following bound for all steps:

$$\begin{aligned}\frac{1}{M}\sum_{i=1}^{M} t^2\mathbb{E}\left\|\boldsymbol{\epsilon}_t^{(i)}\right\|^2 \leq& 2\sigma^2\left((t-t_0)+\frac{t_0}{M}\right)+32\sigma_L^2\tilde{D}_t^2\left((t-t_0)+\frac{5t_0}{4M}\right)\\ \leq& \left(\sigma^2+16\sigma_L^2\tilde{D}_t^2\right)\left(2(t-t_0)+\frac{5t_0}{2M}\right)\end{aligned}$$

Substituting the definition of $\tilde{D}_t$ and using $2(t-t_0)+\frac{5t_0}{2M} \leq 2K_{\text{loc}}+\frac{5t}{2M}$, we obtain:

$$\begin{aligned}\frac{1}{M}\sum_{i=1}^{M} t^2\mathbb{E}\left\|\boldsymbol{\epsilon}_t^{(i)}\right\|^2 \leq& \left(\sigma^2+32\sigma_L^2D_1^2+\frac{64\sigma_L^2\eta^2 t}{M}\sum_{i=1}^{M}\sum_{k=1}^{t}k^2\mathbb{E}\left\|\boldsymbol{\epsilon}_k^{(i)}\right\|^2\right)\left(2K_{\text{loc}}+\frac{5t}{2M}\right)\\ &\left(\sigma^2+32\sigma_L^2D_1^2+\frac{64\sigma_L^2\eta^2 T}{M}\sum_{i=1}^{M}\sum_{k=1}^{T}k^2\mathbb{E}\left\|\boldsymbol{\epsilon}_k^{(i)}\right\|^2\right)\left(2K_{\text{loc}}+\frac{5T}{2M}\right)\end{aligned}$$

Where we increased t into T. Choosing $\eta \leq \frac{1}{8\sigma_L T\sqrt{4K_{\text{loc}}+\frac{5T}{M}}}$, we can apply Lemma C.1 to obtain:

$$\frac{1}{M}\sum_{i=1}^{M} t^2\mathbb{E}\left\|\boldsymbol{\epsilon}_t^{(i)}\right\|^2 \leq \left(\sigma^2+32\sigma_L^2D_1^2\right)\left(4K_{\text{loc}}+\frac{5T}{M}\right)$$

Since this is true for all $t \leq T$, we substitute t back instead of T for both bounds of $\boldsymbol{\epsilon}_t, \eta$. □

F Loss Bound

Proof of Theorem 2.2. Starting from Lemma D.2, we substitute the result of Lemma 2.1 to obtain:

$$\Delta_T \leq \frac{2D_1^2}{\eta T^2}+4\eta\left(\sigma^2+32\sigma_L^2D_1^2\right)\left(4K_{\text{loc}}+\frac{5T}{M}\right) \tag{23}$$

We pick $\eta = \min\left\{\frac{1}{8LT}, \frac{D_1}{T\sqrt{2\sigma^2+64\sigma_L^2D_1^2}\sqrt{4K_{\text{loc}}+\frac{5T}{M}}}\right\}$. Note that this choice satisfies the conditions of Lemma 2.1, because $T \geq t$ and $\sqrt{2\sigma^2+64\sigma_L^2D_1^2} \geq 8\sigma_L D_1$. Using this η, we get:

$$\begin{aligned}\Delta_T \leq& \frac{16LD_1^2}{T}+\frac{4D_1}{T}\sqrt{2\sigma^2+64\sigma_L^2D_1^2}\sqrt{4K_{\text{loc}}+\frac{5T}{M}}\\ \leq& \frac{16LD_1^2}{T}+4D_1\left(\sqrt{2}\sigma+8\sigma_L D_1\right)\left(\frac{2\sqrt{K_{\text{loc}}}}{T}+\sqrt{\frac{5}{TM}}\right)\end{aligned}$$

where we used $\sqrt{a+b} \leq \sqrt{a}+\sqrt{b}$. Using $K_{\text{loc}} \leq \frac{K}{2}$ and $T = R(K_{\text{loc}}+1) \geq \frac{KR}{2}$, we get:

$$\Delta_T \leq \frac{32LD_1^2}{KR}+4D_1\left(\sqrt{2}\sigma+8\sigma_L D_1\right)\left(\frac{2\sqrt{2}}{\sqrt{KR}}+\frac{\sqrt{10}}{\sqrt{MKR}}\right)$$

Bounding $\sqrt{2} \leq \frac{3}{2}$ and $\sqrt{10} \leq \frac{19}{6}$, we obtain:

$$\Delta_T \leq \frac{32LD_1^2}{KR}+D_1\left(3\sigma+16\sigma_L D_1\right)\left(\frac{6}{\sqrt{KR}}+\frac{19}{3\sqrt{MKR}}\right)$$

□

G Experimental Details

We evaluate Local MixVR against four baselines on Fashion-MNIST and CIFAR-10. In every experiment, the total sample budget is held fixed while the communication budget is varied over more than two orders of magnitude. Consequently, the horizontal axis isolates communication frequency from total computation.

G.1 Datasets and Architectures

Fashion-MNIST. Fashion-MNIST (Xiao et al., 2017) contains 60,000 training and 10,000 test grayscale $1 \times 28 \times 28$ images from 10 clothing categories. We train a LeNet-style network with $M \in \{4, 8, 16\}$ workers, a per-worker batch size of 64, and 2 epochs. The network has two 5×5 convolutional layers with 20 and 50 output channels, each followed by ReLU and 2×2 max pooling, and two fully connected layers of sizes $800 \to 50$ and $50 \to 10$.

CIFAR-10. CIFAR-10 (Krizhevsky et al., 2014) contains 50,000 training and 10,000 test color $3 \times 32 \times 32$ images from 10 classes. We train a ResNet-18 (He et al., 2016) with $M = 8$ workers, a per-worker batch size of 16, and 30 epochs. Interior convolutions are initialized from the torchvision ImageNet checkpoint. The 3×3 CIFAR stem and classifier have incompatible shapes and are initialized randomly.

G.2 Methods Compared

- **Local SGD.** Workers use plain SGD locally, and the server averages their parameters at every synchronization. The local optimizer state is reinitialized after each round.
- **Local Momentum.** Workers use SGD with momentum 0.9 and dampening 0.9, corresponding to the exponential-moving-average form of the momentum recursion. The momentum coefficient 0.9 is a common parameter choice and corresponds to $\beta_t = 0.1$ in our notation, since the previous direction is weighted by $1 - \beta_t$. Momentum buffers are averaged across workers at synchronization and carried into the next interval.
- **Local MixVR.** We set $\beta_t = 0.1$ and $\gamma_t = 0.9$. Each round divides its budget of K samples into $K_{\text{avg}} = \lceil \alpha K \rceil$ accumulation samples and $K - K_{\text{avg}}$ local steps; synchronization of the method state is intrinsic to Local MixVR.
- **Minibatch SGD.** The method performs one optimizer update per round using the same samples as the corresponding local interval, with gradients accumulated in bounded micro-batches.
- **Minibatch ASGD.** This is Minibatch SGD with a Nesterov server optimizer and server momentum 0.9, a common parameter choice that likewise corresponds to $\beta_t = 0.1$ in our notation.

G.3 Tuning and Evaluation

Each method is tuned independently at every communication budget. Every reported point is the configuration with the highest mean final test accuracy for that (method, R) cell. All configurations are evaluated using three seeds, and we report the mean and population standard deviation of final test accuracy. At each communication budget, we sweep the learning rate over $\eta \in \{0.01, 0.03, 0.06, 0.1, 0.3\}$ for every method. For Local MixVR, we additionally sweep the mixing parameter over $\alpha \in \{0.01, 0.05, 0.1\}$. This parameter controls the division of each round's K samples between the accumulation phase, which uses $K_{\text{avg}} = \lceil \alpha K \rceil$ samples, and the local-update phase, which uses the remaining $K - K_{\text{avg}}$ samples.

G.4 Additional Results: CIFAR-10

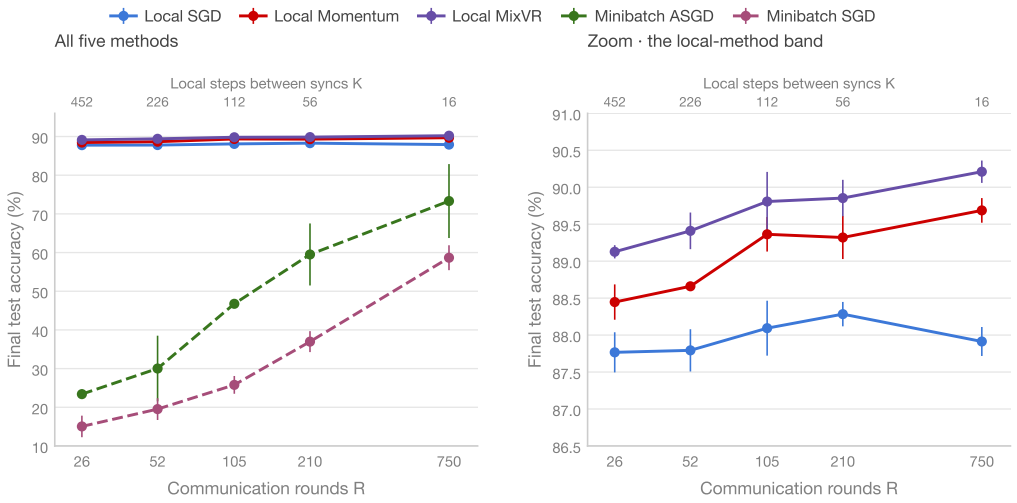

Figure 2: **Test accuracy vs communication rounds R on CIFAR-10. $M = 8$ workers.** The top axis reports the local steps $K = \lceil 11730/R \rceil$. *left:* all five methods; *right:* the local-method accuracy range. Local MixVR achieves the highest accuracy among the local methods at every communication budget, outperforming Local Momentum by 0.45–0.75 percentage points, while Minibatch SGD improves sharply as the number of communication rounds increases.

G.5 Additional Results: Fashion-MNIST

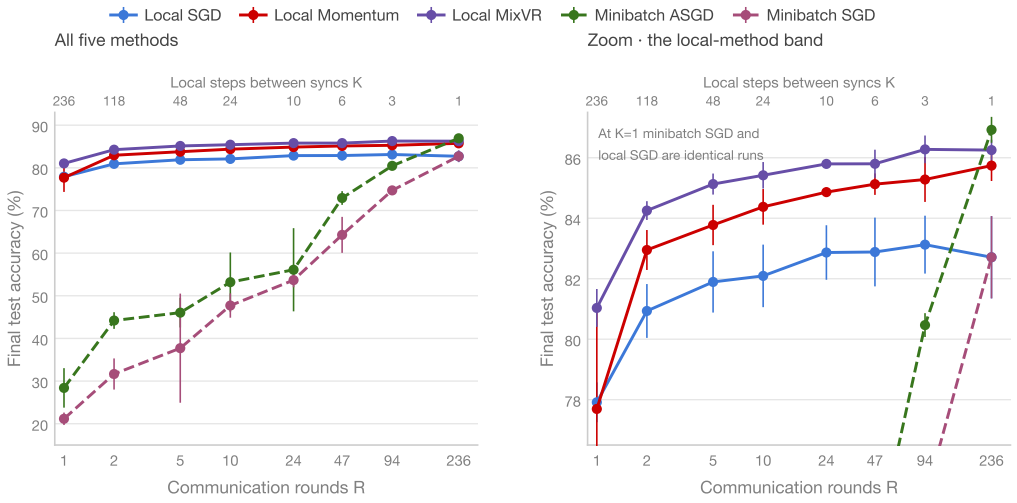

Figure 3: **Test accuracy vs communication rounds R on Fashion-MNIST. $M = 8$ workers.** The top axis reports the local steps $K = \lceil 236/R \rceil$. *left:* all five methods; *right:* the local-method accuracy range. Local MixVR outperforms Local Momentum and Local SGD throughout the sweep; at $R = 24$, it exceeds Local SGD by 2.93 percentage points.

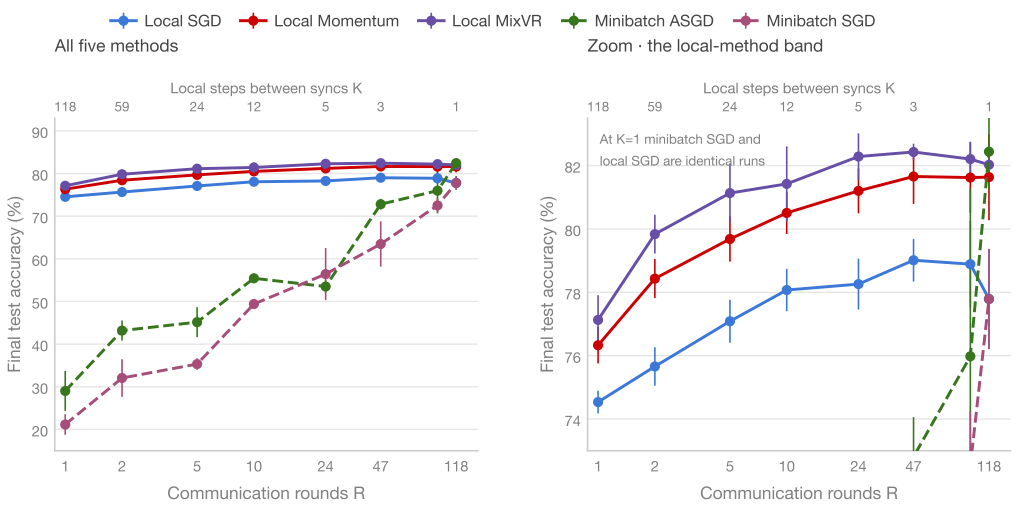

Figure 4: **Test accuracy vs communication rounds R on Fashion-MNIST. $M = 16$ workers.** The top axis reports the local steps $K = \lceil 118/R \rceil$. *left:* all five methods; *right:* the local-method accuracy range. Local MixVR remains the strongest method throughout the sweep; at $R = 24$, it exceeds Local SGD by 4.03 percentage points, while Local Momentum exceeds Local SGD by 2.94 points.