BRINGING ORDER TO ASYNCHRONOUS SGD: TOWARDS OPTIMALITY UNDER DATA-DEPENDENT DELAYS WITH MOMENTUM

Anonymous authors
Paper under double-blind review

ABSTRACT

Asynchronous SGD enables scalable distributed training but suffers from stale gradients. When delays depend on the data, slower samples may be underrepresented, biasing training toward faster samples. We introduce *ordered momentum*, a unified framework that attains the best-known rates for smooth convex and non-convex objectives under both data-independent and data-dependent delays. Notably, we establish (i) the first convergence guarantee for smooth convex objectives with data-dependent delays and, among analyses of data-dependent delays, the first to (ii) benefit from parallelization and (iii) match the tight data-independent rate, with a leading stochastic term independent of the number of workers. Finally, we derive robust learning rates that simplify hyperparameter tuning across convex and non-convex settings.

1 INTRODUCTION

The growing scale of modern machine-learning models and datasets has made distributed training indispensable (Verbraeken et al., 2020; Zhao et al., 2023). Among the available approaches, *asynchronous parallelization* is particularly attractive: workers compute gradients independently and update a shared model without waiting for global synchronization (Mishchenko et al., 2022). By eliminating synchronization barriers, asynchronous SGD reduces idle time due to stragglers, accommodates heterogeneous hardware, and can substantially improve wall-clock throughput compared to its synchronous counterpart (Dekel et al., 2012). This efficiency, however, comes at the cost of *staleness*: while a worker computes a gradient, other workers may update the shared model; consequently, by the time the gradient is applied, it may correspond to an earlier (and potentially substantially different) model iterate.

A key assumption underlying tight convergence guarantees for asynchronous optimization is that delays are independent of the data (Cohen et al., 2021; Attia et al., 2024; Aviv et al., 2021; Mishchenko et al., 2022; Koloskova et al., 2022; Shi et al., 2024). In modern training pipelines, however, computation time often depends directly on the input. For example, language-modeling sequences are commonly grouped by length to improve hardware utilization (Guo et al., 2025; Pouransari et al., 2024). Because longer sequences require more computation, their gradients arrive systematically later and are therefore more stale. The same phenomenon arises in video understanding, where longer clips require more computation. Delays may therefore be coupled with the data distribution, causing faster examples to influence the model earlier and more frequently than slower ones. This implicit sampling bias can destabilize training, slow convergence, and degrade generalization.

This distinction determines how staleness can be handled. When delays are independent of the data, stale gradients can be attenuated without systematically altering the effective data distribution. This observation motivates optimal methods that either adapt the learning rate to the observed delay (Mishchenko et al., 2022; Koloskova et al., 2022; Shi et al., 2024; Venigalla et al., 2020; Barkai et al., 2019; Wu et al., 2022) or discard updates beyond a prescribed delay threshold (Cohen et al., 2021; Attia et al., 2024; Maranjyan et al., 2025). When delays depend on the data, however, staleness carries information about the underlying sample: gradients from computationally expensive examples tend to be more stale. The same mechanisms then systematically downweight or discard these examples, implicitly reweighting the data distribution toward faster samples. Avoiding this bias requires a

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.

different approach. Indeed, the best-known guarantees for data-dependent delays generally rely on a fixed, delay-independent learning rate (Mishchenko et al., 2022). Consequently, attaining the best-known rates in the two delay regimes typically requires different algorithms.

In this work, we introduce the *ordered momentum* framework, a unified approach that achieves the best-known convergence rates for convex and non-convex objectives under both data-independent and data-dependent delays. For smooth convex objectives, we (i) establish the first convergence guarantee under data-dependent delays and, among existing analyses of data-dependent delays, are the first to (ii) exhibit a benefit from parallelization and (iii) match the tight data-independent rate, with a leading stochastic term independent of the number of workers M.

To see this gap, the best-known convex rate under data-dependent delays is $\mathcal{O}\left(\frac{DG\sqrt{M}}{\sqrt{T}}\right)$ (Mishchenko et al., 2022), where D is the domain diameter and G bounds the stochastic gradient norm. This guarantee does not exploit smoothness, and its leading term deteriorates as $\sqrt{M}$. A similar dependence appears in the best-known non-convex rate, whose leading term scales as $\mathcal{O}\left((\frac{GL\Delta M}{T})^{2/3}\right)$, where Δ is the initial optimality gap.

By contrast, for L-smooth objectives under data-independent delays, the tight rate in the convex setting is $\mathcal{O}\left(\frac{D^2LM}{T}+\frac{D\sigma}{\sqrt{T}}\right)$, and in the non-convex setting it is $\mathcal{O}\left(\frac{L\Delta M}{T}+\frac{\sigma\sqrt{L\Delta}}{\sqrt{T}}\right)$ (Mishchenko et al., 2022), where $\sigma \in [0, G]$ bounds the stochastic gradient variance (see Appendix A). Crucially, in both rates, M appears only in a lower-order term, while the dominant stochastic term is independent of the number of workers. Existing guarantees under data-dependent delays therefore fail to capture this parallelization benefit (see Remark 4.3). Our convex analysis closes this gap by recovering an M-independent dominant term even when delays depend on the data, matching the parallelization benefit of the data-independent regime.

Our Contributions. In this work, we propose a momentum-based framework for asynchronous optimization. Our main contributions are as follows:

- We introduce a unified approach attaining the best-known rates for smooth convex and non-convex objectives under both data-independent and data-dependent delays. For convex objectives, we establish the first data-dependent guarantee matching the tight data-independent rate. Unlike the previous best bound, $\mathcal{O}\left(\frac{DG\sqrt{M}}{\sqrt{T}}\right)$, whose leading term deteriorates with the number of workers M, our smooth convex guarantee has an M-independent leading term, capturing a parallelization benefit for the first time under data-dependent delays.
- Our guarantees use a fixed learning rate that can be tuned once offline. In contrast, common delay-adaptive methods (Mishchenko et al., 2022; Koloskova et al., 2022; Shi et al., 2024; Venigalla et al., 2020; Barkai et al., 2019; Wu et al., 2022) use iteration-dependent schedules of the form $\eta_t = \min\left\{\mathcal{O}\left(\frac{1}{L\tau_t}\right), \mathcal{O}\left(\frac{1}{LM}\right), \mathcal{O}\left(\sqrt{\frac{\Delta}{L\sigma^2 T}}\right)\right\}$, potentially suggesting costly online tuning as the delay τ_t varies. Our framework attains the best-known guarantees while retaining standard offline tuning. Moreover, by leveraging a double-momentum mechanism, our framework further preserves the optimal rate across a broad range of learning rates, reducing tuning sensitivity in highly variable asynchronous environments.

Related Work. Asynchronous SGD has long been studied as a way to scale optimization by eliminating synchronization barriers (Tsitsiklis et al., 2003; Mishchenko et al., 2022). Early analyses primarily considered *fixed delays* (Arjevani et al., 2020; Stich & Karimireddy, 2020; Zheng et al., 2017; Giladi et al., 2019), an abstraction suited to homogeneous systems but unable to capture the variability of heterogeneous clusters and edge devices. Subsequent works introduced *virtual iterates*, initially under bounded or fixed delays (Mania et al., 2017; Stich & Karimireddy, 2020) and later in sharper analyses (Mishchenko et al., 2022; Koloskova et al., 2022; Shi et al., 2024). This approach compares the asynchronous iterates with an idealized delay-free trajectory, enabling tighter convergence guarantees. Despite this progress, no existing analysis exploits smoothness to obtain tight convex guarantees under data-dependent delays.

Momentum methods are central to modern machine learning optimization (Polyak, 1964; Nesterov, 2013; Kingma & Ba, 2015; Loshchilov & Hutter, 2019). Early asynchronous momentum methods

were analyzed primarily under restrictive assumptions, such as bounded (Barkai et al., 2019; Hakimi et al., 2019; Yan et al., 2025) or fixed delays (Giladi et al., 2019), limiting their applicability to highly variable environments. More recently, Shi et al. (2024) introduced *ideal momentum*, which assigns each arriving gradient its corresponding synchronous weight, while grouping updates into batches of size M with uniform within-group weights. However, their data-independent guarantees require delay-adaptive learning rates and potentially costly online tuning. In contrast, we introduce refined notions of *ideal momentum* that yield tight guarantees under both data-independent and data-dependent delays using a fixed learning rate that can be tuned once offline. Moreover, by incorporating double momentum (Dahan & Levy, 2025), we further establish the first optimal guarantee for smooth convex optimization under data-dependent delays, matching the tight rate known for data-independent delays. Finally, we show that this optimal rate holds across a broad range of learning rates, demonstrating robustness even in highly variable asynchronous environments.

2 SETTINGS

We consider minimizing the smooth objective

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

where $\mathcal{D}$ is an unknown distribution and samples are drawn i.i.d. We study iterative first-order methods in a distributed system with M workers and a central parameter server ($\mathcal{PS}$). Given $\mathbf{x}$, each worker computes a stochastic gradient $\mathbf{g} := \nabla f(\mathbf{x};\mathbf{z})$, where $\mathbf{z}\sim\mathcal{D}$, satisfying $\mathbb{E}\left[\mathbf{g}\mid\mathbf{x}\right] = \nabla f(\mathbf{x})$.

We focus on the *asynchronous* setting, where the $\mathcal{PS}$ updates the global model immediately upon receiving a worker update. Consequently, gradients may be computed at stale iterates and applied out of order. We denote by $\tau_t \geq 0$ the delay of the update applied at iteration t.

Asynchronous SGD. In this asynchronous setting, the standard SGD update (Mishchenko et al., 2022; Koloskova et al., 2022) takes the form

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \mathbf{g}_{t-\tau_t} \tag{1}$$

where $\eta_t > 0$ is the learning rate and $\mathbf{g}_{t-\tau_t} := \nabla f(\mathbf{x}_{t-\tau_t};\mathbf{z}_{t-\tau_t})$ denotes a stochastic gradient evaluated at a delayed iterate $\mathbf{x}_{t-\tau_t}$.

Assumptions. Throughout the paper, we consider four regimes that combine nonconvex or convex objectives with data-independent or data-dependent delays. Each regime uses the corresponding assumptions stated below. Stochastic statements are with respect to $\mathbf{z}\sim\mathcal{D}$, and the domain is $\mathbb{R}^d$ in the nonconvex setting and $\mathcal{K}$ in the convex setting.

Assumption 2.1 (Smooth nonconvex objective)**.** The objective $f:\mathbb{R}^d\to\mathbb{R}$ is lower bounded by $f^\star := \inf_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) > -\infty$ and has an L-Lipschitz gradient:

$$\left\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\right\| \leq L\left\|\mathbf{x}-\mathbf{y}\right\|, \qquad \forall \mathbf{x},\mathbf{y}\in\mathbb{R}^d$$

We write $\Delta := f(\mathbf{x}_1) - f^\star$.

Assumption 2.2 (Smooth convex objective)**.** The set $\mathcal{K}\subseteq\mathbb{R}^d$ is nonempty, compact, and convex, with diameter D, and $f:\mathcal{K}\to\mathbb{R}$ is convex. Moreover, every sample loss has an L-Lipschitz gradient:

$$\left\|\nabla f(\mathbf{x};\mathbf{z}) - \nabla f(\mathbf{y};\mathbf{z})\right\| \leq L\left\|\mathbf{x}-\mathbf{y}\right\|, \qquad \forall \mathbf{x},\mathbf{y}\in\mathcal{K},\ \mathbf{z}\in\textbf{Support}\{\mathcal{D}\}$$

This implies (see Appendix A) that f is L-smooth and that, for some $\sigma_L\in[0,L]$:

$$\mathbb{E}\left\|(\nabla f(\mathbf{x};\mathbf{z}) - \nabla f(\mathbf{x})) - (\nabla f(\mathbf{y};\mathbf{z}) - \nabla f(\mathbf{y}))\right\|^2 \leq \sigma_L^2\left\|\mathbf{x}-\mathbf{y}\right\|^2, \qquad \forall \mathbf{x},\mathbf{y}\in\mathcal{K}$$

We denote $\mathbf{x}^\star := \arg\min_{x\in\mathcal{K}} f(\mathbf{x})$, $f^\star := f(\mathbf{x}^\star)$, and $G^\star := \left\|\nabla f(\mathbf{x}^\star)\right\|$.

Assumption 2.3 (Bounded stochastic noise for data-independent delays)**.** The delay is independent of the stochastic samples, and there exist $\sigma\geq 0$, such that $\mathbb{E}\left\|\nabla f(\mathbf{x};\mathbf{z}) - \nabla f(\mathbf{x})\right\|^2 \leq \sigma^2$, for every $\mathbf{x}$ in the relevant domain.

Assumption 2.4 (Bounded stochastic gradients for data-dependent delays)**.** There exists $G>0$, such that $\left\|\nabla f(\mathbf{x};\mathbf{z})\right\| \leq G$, for every $\mathbf{x}$ in the relevant domain and every $\mathbf{z}\in\textbf{Support}\{\mathcal{D}\}$. This implies $\mathbb{E}\left\|\nabla f(\mathbf{x};\mathbf{z}) - \nabla f(\mathbf{x})\right\|^2 \leq \sigma^2$ for some $\sigma\in[0,G]$ (see Appendix A).

Notations. $\left\|\cdot\right\|$ denotes the Euclidean (ℓ_2) norm. For any $N\in\mathbb{N}$, we define $[N] := \{1,\dots,N\}$.

3 Momentum-Based Asynchronous Framework

Momentum is a cornerstone of modern optimization, accelerating convergence while stabilizing noisy stochastic updates (Polyak, 1964; Nesterov, 2013; Kingma & Ba, 2015; Loshchilov & Hutter, 2019; Jordan et al., 2024). At a high level, it extends vanilla SGD by maintaining a running average of past gradients, combining historical information with the current gradient to determine the update direction. Formally, in the single-worker setting, the classical momentum method of Polyak (1964) updates the momentum buffer and model parameters as follows:

$$\mathbf{m}_t^{\text{syn}} = \beta \mathbf{g}_t + (1-\beta)\mathbf{m}_{t-1}^{\text{syn}}, \quad \mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{m}_t^{\text{syn}} \tag{2}$$

where $\eta > 0$ is the learning rate, $\beta \in (0,1)$ is the momentum parameter, $\mathbf{g}_t := \nabla f(\mathbf{x}_t; \mathbf{z}_t)$ is the stochastic gradient at step t, and $\mathbf{m}_t^{\text{syn}}$ is the momentum in the single-worker synchronous baseline.

Momentum-Based Solution. A natural extension of momentum to the asynchronous setting is to replace the current stochastic gradient with a delayed one. Specifically, let $\tau_t \geq 0$ denote the delay at iteration t, so that $\mathbf{g}_{t-\tau_t} := \nabla f(\mathbf{x}_{t-\tau_t}; \mathbf{z}_{t-\tau_t})$ is evaluated at the stale iterate $\mathbf{x}_{t-\tau_t}$. The resulting asynchronous momentum update is

$$\mathbf{m}_t^{\text{asy}} = \beta \mathbf{g}_{t-\tau_t} + (1-\beta)\mathbf{m}_{t-1}^{\text{asy}} \tag{3}$$

where $\mathbf{m}_t^{\text{asy}}$ denotes the asynchronous momentum buffer. When $\tau_t = 0$ for all t, the update reduces to the synchronous method, and $\mathbf{m}_t^{\text{asy}} = \mathbf{m}_t^{\text{syn}}$. Despite its simplicity, this extension has a fundamental drawback: momentum can assign greater weight to gradients evaluated at older iterates than to those evaluated more recently. To see this, it is useful to unroll the synchronous momentum recursion in Equation (2), making the contribution of each past gradient explicit:

$$\mathbf{m}_t^{\text{syn}} = \beta \sum_{\tau=0}^{t-1} (1-\beta)^\tau \mathbf{g}_{t-\tau} \tag{4}$$

This expansion shows that momentum forms an *exponential moving average* of past gradients. Thus, the gradient from τ iterations ago is weighted by exactly $\beta(1-\beta)^\tau$, where $(1-\beta)^\tau$ reflects its "distance" from the current iterate. In other words, momentum naturally emphasizes recent gradients while retaining a diminishing contribution from older ones.

Our Proposal. The unrolled recursion in Equation (4) provides a simple interpretation of momentum; in the ideal delay-free setting, each stochastic gradient contributes to the momentum buffer according to its exact position in an exponentially decaying sequence: a gradient computed τ iterations in the past is weighted by $\beta(1-\beta)^\tau$. Preserving this ideal ordering suggests that, under delays, each gradient should be incorporated using precisely the weight it would receive in the single-worker trajectory. Motivated by this observation, we modify the asynchronous momentum update to

$$\mathbf{m}_t = \beta(1-\beta)^{\tau_t} \mathbf{g}_{t-\tau_t} + (1-\beta)\mathbf{m}_{t-1} \tag{5}$$

A related "ordered weighting" perspective was recently highlighted by Shi et al. (2024), who group iterations into blocks of size M and assign each block a uniform weight in their *virtual momentum* construction that mimics the ideal momentum. In contrast, we adopt the same underlying principle but enforce it exactly. Specifically, each delayed gradient $\mathbf{g}_{t-\tau_t}$ is assigned the exact coefficient $\beta(1-\beta)^{\tau_t}$ it would have in the ideal, delay-free trajectory. As a result, stale gradients are neither discarded nor treated as fresh; instead, they are consistently integrated into our momentum structure, with any deviation arising solely from missing future gradients that have not yet arrived.

Formally, for iteration $t \geq 1$ and index $1 \leq k \leq t$, define the *weighted gradient* as

$$\mathbf{g}_{t,k} := \beta(1-\beta)^{t-k} \mathbf{g}_k$$

where for $k = 1$, we only use the contribution of the first worker, while the other workers replace the first gradient they receive with zero. Thus, the resulting momentum with ordered weights (Equation (5) and Algorithm 1) can be represented as

$$\mathbf{m}_t = \sum_{k \in [t] \setminus \mathcal{T}(t)} \mathbf{g}_{t,k} \tag{6}$$

Algorithm 1 Asynchronous Momentum with Ordered Weights

Input: #Iterations T, initial point $\mathbf{x}_1 \in \mathbb{R}^d$, learning rate $\eta > 0$, momentum parameter $\beta \in (0, 1)$.
Set $\mathbf{m}_0 = \mathbf{0}^d$.
for $t = 1$ to T **do**
  Receive $\mathbf{g}_{t-\tau_t} := \nabla f(\mathbf{x}_{t-\tau_t}; \mathbf{z}_{t-\tau_t})$ from a worker, and replace $\mathbf{g}_{t-\tau_t} := \mathbf{0}^d$ on its first response for workers other than the first, i.e., when $t - \tau_t = 1$ and $t > 1$.
  Update: $\mathbf{m}_t = \beta(1-\beta)^{\tau_t}\mathbf{g}_{t-\tau_t} + (1-\beta)\mathbf{m}_{t-1}$
  Update: $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\mathbf{m}_t$
  Send $\mathbf{x}_{t+1}$ back to the worker.
end for
Output: $\bar{\mathbf{x}}_T$ (chosen uniformly at random from the set $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$).

Here, $\mathcal{T}(t)$ denotes the set of iterations whose gradients are still pending at time t, and therefore $|\mathcal{T}(t)| \leq M - 1$. Thus, Equation (6) contains all gradients associated with iterations up to t that have already arrived, while omitting those indexed for the iterations that have not yet arrived and are therefore still missing. The resulting momentum buffer preserves the weighting of the ideal ordered momentum, differing only by at most $M - 1$ pending gradients. Based on this representation, we define the ideal momentum $\hat{\mathbf{m}}_t$ at iteration t as

$$\hat{\mathbf{m}}_t := \sum_{k=1}^{t} \mathbf{g}_{t,k} \tag{7}$$

Where we add the gradients to fill in the missing iterations. We shall *use these gradients only for the sake of analysis*. Specifically, we shall use these gradients to define an ideal (impractical) version of the momentum, which serves as an analytical tool for comparing with and analyzing the practical momentum used in Algorithm 1. Finally, the stochastic error of delayed momentum, $\boldsymbol{\epsilon}_t := \mathbf{m}_t - \nabla f(\mathbf{x}_t)$, can be decomposed as

$$\boldsymbol{\epsilon}_t = \hat{\boldsymbol{\epsilon}}_t + \mathbf{b}_t$$

where $\hat{\boldsymbol{\epsilon}}_t = \hat{\mathbf{m}}_t - \nabla f(\mathbf{x}_t)$ is the error of the ideal momentum and $\mathbf{b}_t$ is the difference induced by delay. This difference has the explicit form

$$\mathbf{b}_t := \mathbf{m}_t - \hat{\mathbf{m}}_t = -\sum_{k \in \mathcal{T}(t)} \mathbf{g}_{t,k} \tag{8}$$

which scales at most linearly with the number of workers, i.e., $\|\mathbf{b}_t\| \leq \mathcal{O}(M)$. Consequently, the stochastic error of the delayed momentum, $\boldsymbol{\epsilon}_t$, is controlled by two terms: (i) an $\mathcal{O}(M)$ bias from the pending gradients and (ii) the same stochastic error that appears in the ideal delay-free setting. The key intuition is that our asynchronous momentum aims to be *as close as possible to ideal, delay-free momentum*, so its bias depends only on the at most $M - 1$ missing updates, rather than on the magnitude of their delays.

Ordered Momentum for Data-Independent Delays.

Theorem 3.1 (Non-convex Smooth Objectives on Data-Independent Delays)**.** *Let f satisfy assumptions 2.1 and 2.3. Consider the iterates $\{\mathbf{x}_t\}_{t=1}^T$ generated by Algorithm 1 with $\beta = \min\left\{\frac{1}{8(M-1)}, \frac{2\sqrt{3L\Delta}}{\sigma\sqrt{T}}\right\}$, and $\eta = \frac{\beta}{4L}$. Then:*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{L\Delta M}{T} + \frac{\sigma\sqrt{L\Delta}}{\sqrt{T}}\right)$$

Ordered Momentum for Data-Dependent Delays.

Theorem 3.2 (Non-convex Lipschitz-Smooth Objectives on Data-Dependent Delays)**.** *Let f satisfy assumptions 2.1 and 2.4. Consider the iterates $\{\mathbf{x}_t\}_{t=1}^T$ generated by Algorithm 1 with $\beta = \min\left\{\frac{1}{2}, \frac{\sqrt{5L\Delta}}{\sigma\sqrt{T}}, \left(\frac{5L\Delta}{2G^2(M-1)^2T}\right)^{1/3}\right\}$, and $\eta = \frac{\beta}{2\sqrt{2}L}$. Then:*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \mathcal{O}\left(\frac{L\Delta}{T} + \frac{\sigma\sqrt{L\Delta}}{\sqrt{T}} + \left(\frac{GL\Delta M}{T}\right)^{2/3}\right)$$

Remark 3.3 (A unified approach to both delay regimes). Existing optimal methods typically treat data-independent and data-dependent delays differently. For example, in asynchronous SGD, under data-dependent delays, the optimal guarantee can be achieved with a fixed learning rate (Mishchenko et al., 2022; Koloskova et al., 2022), whereas attaining the optimal rate under data-independent delays typically requires adapting the learning rate to the realized delay (Mishchenko et al., 2022; Koloskova et al., 2022). In contrast, our method matches the best-known guarantees for general smooth non-convex objectives in both regimes within the same algorithmic framework, using a fixed learning rate.

Remark 3.4 (A fixed, delay-independent learning rate). Several optimal methods for data-independent delays use learning rates that adapt to the realized delay (Mishchenko et al., 2022; Koloskova et al., 2022; Shi et al., 2024), for example:

$$\eta_t = \min\left\{\Theta\left(\frac{1}{L\tau_t}\right), \Theta\left(\frac{1}{LM}\right), \Theta\left(\sqrt{\frac{\Delta}{L\sigma^2 T}}\right)\right\} \tag{9}$$

Implementing such schedules can be difficult in practice: the quantities entering the minimum, such as L, Δ, and σ^2, are generally unknown, while the realized delay τ_t may vary substantially across iterations. Consequently, a learning rate tuned for one delay may be unsuitable for another, potentially requiring repeated online retuning rather than a single offline search. In contrast, our method uses a fixed learning rate that is independent of the realized delays. It can therefore be tuned once offline, for example via a standard grid search, and kept unchanged throughout training.

4 Towards Optimality via Double Momentum

Despite substantial progress in asynchronous optimization, convergence in the smooth convex setting with data-dependent delays has remained unresolved. We close this gap by establishing the first convergence guarantee for this setting, matching the tight rate known for data-**in**dependent delays (Mishchenko et al., 2022). To obtain this result, we instantiate our framework with the double-momentum mechanism of μ^2-SGD (Dahan & Levy, 2025), which combines two complementary forms of momentum: (i) STORM, a momentum-based variance-reduction correction that controls stochastic gradient noise (Cutkosky & Orabona, 2019), and (ii) parameter-level momentum, also known as Anytime-SGD (Cutkosky, 2019). Finally, through our ordered-momentum framework, we show that the resulting convergence guarantees remain robust across a broad range of learning rates, even in highly unstable asynchronous environments with data-dependent delays.

Notations. For any sequence $\{a_n\}_n$, we use the shorthand $a_{i:j} := \sum_{n=i}^{j} a_n$, and for every $\mathbf{x} \in \mathbb{R}^d$, we denote the orthogonal projection of $\mathbf{x}$ onto a set $\mathcal{K}$ by $\Pi_{\mathcal{K}}(\mathbf{x}) := \arg\min_{\mathbf{y}\in\mathcal{K}} \|\mathbf{y} - \mathbf{x}\|^2$.

4.1 First Momentum: Stochastic Recursive Momentum

The first momentum component in μ^2-SGD is STORM (Cutkosky & Orabona, 2019). STORM extends classical momentum (Polyak, 1964) by incorporating a sample-reuse correction term into the momentum recursion:

$$\mathbf{d}_t^{\text{syn}} = \underbrace{(1-\beta)\mathbf{d}_{t-1}^{\text{syn}} + \beta\mathbf{g}_t}_{\text{classical momentum}} + \underbrace{(1-\beta)\left(\mathbf{g}_t - \tilde{\mathbf{g}}_{t-1}\right)}_{\text{correction term}}$$

where $\mathbf{g}_t := \nabla f(\mathbf{x}_t; \mathbf{z}_t)$ and $\tilde{\mathbf{g}}_{t-1} := \nabla f(\mathbf{x}_{t-1}; \mathbf{z}_t)$ use the same sample $\mathbf{z}_t$. Applying our asynchronous framework to this recursion yields the **ordered asynchronous STORM** update:

$$\mathbf{d}_t = (1-\beta)\mathbf{d}_{t-1} + (1-\beta)^{\tau_t}(\mathbf{g}_{t-\tau_t} - (1-\beta)\tilde{\mathbf{g}}_{t-\tau_t-1}) \tag{10}$$

This recovers the same momentum term as in Equation (5), along with the STORM correction term, which is likewise ordered according to the correct time index.

Role of the correction. The STORM correction leverages objective smoothness to correlate gradient estimates at consecutive iterates. In particular, the stochastic error in the gradient difference is controlled by $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|$, thereby reducing the estimator variance (Cutkosky & Orabona, 2019). Classical momentum stabilizes the updates but generally introduces a persistent bias:

$\mathbb{E}\left[\mathbf{m}_t^{\text{syn}} - \nabla f(\mathbf{x}_t)\right] \neq 0$. In contrast, the STORM estimator is unbiased in total expectation, $\mathbb{E}\left[\mathbf{d}_t^{\text{syn}} - \nabla f(\mathbf{x}_t)\right] = 0$ (but not conditionally unbiased). Equivalently, its update can be written as

$$\mathbf{d}_t^{\text{syn}} = \mathbf{g}_t + \underbrace{(1-\beta)\left(\mathbf{d}_{t-1}^{\text{syn}} - \tilde{\mathbf{g}}_{t-1}\right)}_{\text{unbiased term}}$$

4.2 Second Momentum: Anytime-SGD

The Anytime-GD mechanism, introduced by Cutkosky (2019), applies gradient updates to an auxiliary sequence while evaluating gradients at an averaged query point. Specifically, it maintains gradient-descent iterates $\{\mathbf{w}_t\}_t$ and query points $\{\mathbf{x}_t\}_t$, where $\mathbf{x}_t$ is a weighted average of $\{\mathbf{w}_\tau\}_{\tau=1}^t$ with positive weights $\{\alpha_\tau\}_{\tau=1}^t$. The updates are

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{K}}\left(\mathbf{w}_t - \eta\alpha_t\mathbf{g}_t\right), \qquad \mathbf{x}_{t+1} = \mathbf{x}_t + \frac{\alpha_{t+1}}{\alpha_{1:t+1}}\left(\mathbf{w}_{t+1} - \mathbf{x}_t\right)$$

where $\mathbf{g}_t := \nabla f(\mathbf{x}_t; \mathbf{z}_t)$ is a stochastic gradient evaluated at the query point $\mathbf{x}_t$. A key property of Anytime-GD is that the averaging update ensures:

$$\|\mathbf{x}_{t+1} - \mathbf{x}_t\| \leq \frac{\alpha_{t+1}}{\alpha_{1:t+1}} D$$

Thus, parameter-level momentum brings successive query points closer together, providing additional stability, and when combined with STORM, this contraction further reduces stochastic error and improves robustness, as shown by Dahan & Levy (2025).

In asynchronous settings, Aviv et al. (2021) showed that this mechanism adapts to delays by using

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{K}}\left(\mathbf{w}_t - \eta\alpha_t\mathbf{g}_{t-\tau_t}\right), \qquad \mathbf{x}_{t+1} = \mathbf{x}_t + \frac{\alpha_{t+1}}{\alpha_{1:t+1}}\left(\mathbf{w}_{t+1} - \mathbf{x}_t\right)$$

with $\alpha_t = t$. However, their guarantees require delays to be independent of the data. Instead, by replacing the SGD estimator with STORM in our asynchronous momentum framework, we show that optimal guarantees hold even when the data distribution and delay are dependent.

4.3 Combining Anytime-SGD and STORM

We combine Anytime-SGD with STORM by replacing the stochastic gradient in the Anytime update with the STORM estimator:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{K}}\left(\mathbf{w}_t - \eta\alpha_t\mathbf{d}_t^{\text{syn}}\right), \qquad \mathbf{x}_{t+1} = \mathbf{x}_t + \frac{\alpha_{t+1}}{\alpha_{1:t+1}}\left(\mathbf{w}_{t+1} - \mathbf{x}_t\right)$$

We use increasing weights $\{\alpha_t\}_{t\geq 1}$, with $\alpha_0 = 0$, and choose the time-varying momentum parameter

$$\beta_t = 1 - \frac{\alpha_{t-1}}{\alpha_t}$$

Under this choice, the *ideal* STORM estimator admits the following compact representation:

$$\alpha_t\mathbf{d}_t^{\text{syn}} = \alpha_{t-1}\mathbf{d}_{t-1}^{\text{syn}} + \alpha_t\mathbf{g}_t - \alpha_{t-1}\tilde{\mathbf{g}}_{t-1}$$

We can unroll this equality and get:

$$\mathbf{d}_t^{\text{syn}} = \sum_{k=1}^{t}\left(\frac{\alpha_k}{\alpha_t}\mathbf{g}_k - \frac{\alpha_{k-1}}{\alpha_t}\tilde{\mathbf{g}}_{k-1}\right)$$

Since $\frac{\alpha_k}{\alpha_t} = \prod_{m=k+1}^{t}(1-\beta_m)$, this form extends Equation (4) to time-varying β_t. For the asynchronous method, define the ordered momentum increment:

$$\mathbf{s}_t := \alpha_t\mathbf{g}_t - \alpha_{t-1}\tilde{\mathbf{g}}_{t-1}$$

With this choice, the *ordered* STORM momentum simplifies to:

$$\alpha_t\mathbf{d}_t = \alpha_{t-1}\mathbf{d}_{t-1} + \mathbf{s}_{t-\tau_t} \tag{11}$$

In general, $\mathbf{d}_t^{\text{syn}}$ is an unbiased estimator of $\nabla f(\mathbf{x}_t)$ (Cutkosky & Orabona, 2019). In the asynchronous setting, $\mathbf{s}_t$ is replaced by a delayed increment $\mathbf{s}_{t-\tau_t}$ (Equation (11)), which introduces bias into the estimator. As discussed earlier, for the choice of $\alpha_t = t$, the magnitude of the bias of $\alpha_t\mathbf{d}_t$ scales as $\mathcal{O}(M)$ (see Section 3), which means that the bias of $\mathbf{d}_t$ is $\mathcal{O}\left(\frac{M}{t}\right)$.

Algorithm 2 Asynchronous μ^2-SGD with Ordered Weights

Input: Initial point $\mathbf{x}_0 \in \mathcal{K}$, step size $\eta > 0$, weights $\{\alpha_t > 0\}_{t=1}^T$, $\alpha_0 = 0$. Set $\mathbf{w}_1 = \mathbf{x}_1 = \mathbf{x}_0$.
for $t = 1$ to T **do**
  Get $\mathbf{g}_{t-\tau_t} := \nabla f(\mathbf{x}_{t-\tau_t}; \mathbf{z}_{t-\tau_t})$ and $\tilde{\mathbf{g}}_{t-\tau_t-1} := \nabla f(\mathbf{x}_{t-\tau_t-1}; \mathbf{z}_{t-\tau_t})$ from worker.
  Compute: $\mathbf{s}_{t-\tau_t} = \alpha_{t-\tau_t}\mathbf{g}_{t-\tau_t} - \alpha_{t-\tau_t-1}\tilde{\mathbf{g}}_{t-\tau_t-1}$
  Update: $\alpha_t \mathbf{d}_t = \alpha_{t-1}\mathbf{d}_{t-1} + \mathbf{s}_{t-\tau_t}$
  Update: $\mathbf{w}_{t+1} = \Pi_{\mathcal{K}}\left(\mathbf{w}_t - \eta\alpha_t \mathbf{d}_t\right)$
  Update: $\mathbf{x}_{t+1} = \mathbf{x}_t + \frac{\alpha_{t+1}}{\alpha_{1:t+1}}\left(\mathbf{w}_{t+1} - \mathbf{x}_t\right)$
  Send $\mathbf{x}_{t+1}$ back to the worker.
end for
Output: $\mathbf{x}_T$

Ordered Double Momentum for Data-Independent Delays.

Theorem 4.1 (Convex Smooth Objectives on Data Independent Delays)**.** *Let f satisfy assumptions 2.2 and 2.3. Consider $\{\mathbf{x}_t\}_{t=1}^T$ to be the iterates generated by the update rule according to Algorithm 2 with $\alpha_t = t$, where η satisfies $4L \le \frac{1}{T\eta} \le \mathcal{O}\left(\frac{G^\star}{D} + \left(\frac{\sigma}{D} + \sigma_L\right)\sqrt{T} + LM\right)$. Then we get an excess loss bound of:*

$$\mathbb{E}f(\mathbf{x}_T) - f^\star \le \mathcal{O}\left(\frac{DG^\star}{T} + \frac{D\sigma + D^2\sigma_L}{\sqrt{T}} + \frac{D^2LM}{T}\right)$$

Ordered Double Momentum for Data-Dependent Delays.

Theorem 4.2 (Convex Lipschitz-Smooth Objectives on Data-Dependent Delays)**.** *Let f satisfy assumptions 2.2 and 2.4. Consider $\{\mathbf{x}_t\}_{t=1}^T$ to be the iterates generated by the update rule according to Algorithm 2 with $\alpha_t = t$, where η satisfies $4L \le \frac{1}{T\eta} \le \mathcal{O}\left(\left(\frac{\sigma}{D} + \sigma_L\right)\sqrt{T} + \left(\frac{G}{D} + L\right)M\right)$. Then we get an excess loss bound of:*

$$\mathbb{E}f(\mathbf{x}_T) - f^\star \le \mathcal{O}\left(\frac{D\sigma + D^2\sigma_L}{\sqrt{T}} + \frac{D(G + DL)M}{T}\right)$$

Remark 4.3. **Optimality under data-dependent delays.** To the best of our knowledge, Theorem 4.2 establishes the first (i) rate for smooth convex objectives under data-dependent delays and, among existing guarantees in this regime, the first to (ii) exhibit an M-independent dominant term and (iii) match the tight rate under data-independent delays. Indeed, the previous convex guarantee applies to non-smooth objectives and scales as $\mathcal{O}\left(DG\sqrt{M/T}\right)$, incurring a $\mathcal{O}\left(\sqrt{M}\right)$ slowdown in its dominant term (Mishchenko et al., 2022). Similarly, the best-known smooth non-convex guarantee under data-dependent delays retains a dependence on M in its dominant term of $\mathcal{O}\left((GL\Delta M/T)^{2/3}\right)$. In contrast, our smooth convex bound confines the dependence on M to the lower-order term $\mathcal{O}\left((D^2L + DG)M/T\right)$, while its dominant $\mathcal{O}\left(1/\sqrt{T}\right)$ term is independent of M. It therefore matches the best-known data-independent rates (Theorem 4.1; Aviv et al., 2021; Cohen et al., 2021) and recovers their leading-order parallelization benefit despite data-dependent delays.

Remark 4.4 (Robustness to Learning-Rate Selection). Theorems 4.1 and 4.2 show that μ^2-SGD achieves the stated guarantees over a broad range of learning rates. Specifically, any $\eta \in [\eta_{\min}, \eta_{\max}]$ is admissible, and $\eta_{\max}/\eta_{\min} = \Theta\left(\sqrt{T} + M\right)$. Thus, even under highly variable asynchrony and data-dependent delays, the method tolerates learning-rate misspecification by a factor of order $\sqrt{T} + M$. This robustness reduces the need for fine-grained hyperparameter tuning.

Remark 4.5 (Time complexity). Using the framework of Tyurin & Richtárik (2023), the iteration-complexity bounds in Theorems 3.2 and 4.2 translate into the first best-known time-complexity guarantees under general data-dependent delays. Furthermore, when delays are worker-dependent, our method can be combined with a simple subset-selection rule that chooses the fastest workers (see e.g., Maranjyan et al. (2025)), thereby recovering the optimal time-complexity bounds for worker-dependent delays. Thus, one can use our framework to combine the advantages of both settings: establishing optimal time complexity for general data-dependent delays and worker-dependent delays.

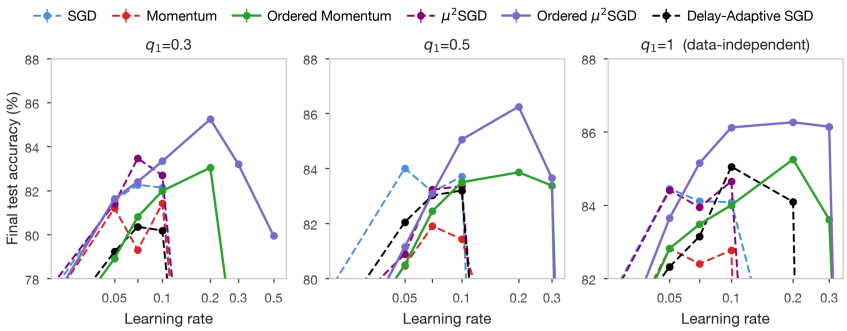


Figure 1: **Learning-rate robustness on Fashion-MNIST with 7 workers.** Each point reports the mean final test accuracy. Both ordered methods remain stable over a wider range of learning rates.

5 Experiments

We evaluate our framework on Fashion-MNIST (Xiao et al., 2017) under controlled delay distributions (Section D.4). We evaluate $M \in \{5, 7\}$ workers and vary the proportion of slow, high-quality samples using $q_1 \in \{0.3, 0.5, 1.0\}$. For $q_1 \in \{0.3, 0.5\}$, higher-quality samples are assigned longer delays, inducing data-dependent asynchrony. When $q_1 = 1.0$, all samples retain their original quality, removing this dependence and providing a data-independent control. We train a convolutional network and simulate asynchrony according to the delay model in Section D.4. We compare Ordered Momentum and Ordered μ^2-SGD against standard asynchronous baselines: Momentum (Equation (3); Polyak, 1964), μ^2-SGD (Dahan & Levy, 2025), vanilla SGD (Equation (1); Koloskova et al., 2022), and Delay-Adaptive SGD (Equation (9); Mishchenko et al., 2022). For both μ^2-SGD variants, we adopt the constant-parameter formulation of Dahan & Levy (2025). The naive asynchronous variant updates

$$\begin{aligned} \mathbf{d}_t &= \mathbf{g}_{t-\tau_t} + (1-\beta)\left(\mathbf{d}_{t-1} - \tilde{\mathbf{g}}_{t-\tau_t-1}\right) \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \eta \mathbf{d}_t, \qquad \mathbf{x}_{t+1} = \gamma \mathbf{w}_{t+1} + (1-\gamma)\mathbf{x}_t \end{aligned} \tag{12}$$

Ordered μ^2-SGD uses the same constant momentum parameters but replaces the naive update of $\mathbf{d}_t$ with the ordered STORM update in Equation (10). We implement all methods in PyTorch and train for 20 epochs using a mini-batch size of 1024. Each point in the learning-rate sweep reports the final test accuracy, averaged over three random seeds. All experiments are conducted on an NVIDIA A100 GPU. Complete experiments, implementation, and hyperparameter details are provided in Appendix D. Results for both worker counts are reported in Figures 1 and 4.

Results. Across both worker counts and all three values of q_1, ordering consistently improves momentum-based methods. Ordered Momentum and Ordered μ^2-SGD remain effective across a wider range of learning rates than their unordered counterparts and asynchronous SGD baselines. This behavior is consistent with the learning-rate robustness guarantees established for the non-convex and convex settings in Remark 3.4 and Remark 4.4. Therefore, the results demonstrate that ordering can enhance the robustness of momentum and enables more stable optimization under both data-independent and data-dependent asynchrony.

6 Conclusions and Future Work

This work introduces a unified momentum-based framework for asynchronous optimization that achieves the best-known convergence rates under both data-independent and data-dependent delays. In particular, we established the first convergence guarantee for smooth convex optimization with data-dependent delays and showed that it matches the tight rate known for the data-independent setting. Our framework uses a fixed, delay-independent learning rate, simplifying hyperparameter tuning, even under highly variable asynchrony. Future work includes extending this framework to adaptive optimizers such as Adam and AdamW, as well as related variance-reduced methods, to enable more robust large-scale asynchronous training.

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

Yossi Arjevani, Ohad Shamir, and Nathan Srebro. A tight convergence analysis for stochastic gradient descent with delayed updates. In *Algorithmic Learning Theory*, pp. 111–132. PMLR, 2020.

Amit Attia, Ofir Gaash, and Tomer Koren. Faster stochastic optimization with arbitrary delays via asynchronous mini-batching. *arXiv preprint arXiv:2408.07503*, 2024.

Rotem Zamir Aviv, Ido Hakimi, Assaf Schuster, and Kfir Yehuda Levy. Asynchronous distributed learning: Adapting to gradient delays without prior knowledge. In *International Conference on Machine Learning*, pp. 436–445. PMLR, 2021.

Saar Barkai, Ido Hakimi, and Assaf Schuster. Gap aware mitigation of gradient staleness. *arXiv preprint arXiv:1909.10802*, 2019.

Alon Cohen, Amit Daniely, Yoel Drori, Tomer Koren, and Mariano Schain. Asynchronous stochastic optimization robust to arbitrary delays. *Advances in Neural Information Processing Systems*, 34: 9024–9035, 2021.

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 Yehuda Levy. Do stochastic, feel noiseless: Stable stochastic optimization via a double momentum mechanism. In *The Thirteenth International Conference on Learning Representations*, 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.

Niv Giladi, Mor Shpigel Nacson, Elad Hoffer, and Daniel Soudry. At stability's edge: How to adjust hyperparameters to preserve minima selection in asynchronous training of neural networks? *arXiv preprint arXiv:1909.12340*, 2019.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Ido Hakimi, Saar Barkai, Moshe Gabel, and Assaf Schuster. Taming momentum in a distributed asynchronous environment. *arXiv preprint arXiv:1907.11612*, 2019.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cecista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan.github.io/posts/muon*, 6, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Anastasiia Koloskova, Sebastian U Stich, and Martin Jaggi. Sharper convergence guarantees for asynchronous sgd for distributed and federated learning. *Advances in Neural Information Processing Systems*, 35:17202–17215, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *7th International Conference on Learning Representations (ICLR)*, 2019.

Horia Mania, Xinghao Pan, Dimitris Papailiopoulos, Benjamin Recht, Kannan Ramchandran, and Michael I Jordan. Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM Journal on Optimization*, 27(4):2202–2229, 2017.

Artavazd Maranjyan, Alexander Tyurin, and Peter Richtárik. Ringmaster asgd: The first asynchronous sgd with optimal time complexity. *arXiv preprint arXiv:2501.16168*, 2025.

Konstantin Mishchenko, Francis Bach, Mathieu Even, and Blake E Woodworth. Asynchronous sgd beats minibatch sgd under arbitrary delays. *Advances in Neural Information Processing Systems*, 35:420–433, 2022.

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

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

Hadi Pouransari, Chun-Liang Li, Jen-Hao Chang, Pavan Kumar Anasosalu Vasu, Cem Koc, Vaishaal Shankar, and Oncel Tuzel. Dataset decomposition: Faster llm training with variable sequence length curriculum. *Advances in Neural Information Processing Systems*, 37:36121–36147, 2024.

Roie Reshef and Kfir Yehuda Levy. Private and federated stochastic convex optimization: efficient strategies for centralized systems. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Roie Reshef and Kfir Yehuda Levy. Privacy-preserving federated convex optimization: Balancing partial-participation and efficiency via noise cancellation. In *Forty-second International Conference on Machine Learning*, 2025.

Chang-Wei Shi, Yi-Rui Yang, and Wu-Jun Li. Ordered momentum for asynchronous sgd. *Advances in Neural Information Processing Systems*, 37:16943–16982, 2024.

Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Sgd with delayed gradients. *Journal of Machine Learning Research*, 21(237):1–36, 2020.

John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9): 803–812, 2003.

Alexander Tyurin and Peter Richtárik. Optimal time complexities of parallel stochastic optimization methods under a fixed computation model. *Advances in Neural Information Processing Systems*, 36:16515–16577, 2023.

Abhinav Venigalla, Atli Kosson, Vitaliy Chiley, and Urs Köster. Adaptive braking for mitigating gradient delay. *arXiv preprint arXiv:2007.01397*, 2020.

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.

Xuyang Wu, Sindri Magnusson, Hamid Reza Feyzmahdavian, and Mikael Johansson. Delay-adaptive step-sizes for asynchronous learning. In *International Conference on Machine Learning*, pp. 24093–24113. PMLR, 2022.

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

Wenjing Yan, Xiangyu Zhong, Xiaolu Wang, and Ying-Jun Angela Zhang. Momentum-driven adaptivity: Towards tuning-free asynchronous federated learning. In *Forty-second International Conference on Machine Learning*, 2025.

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.

Shuxin Zheng, Qi Meng, Taifeng Wang, Wei Chen, Nenghai Yu, Zhi-Ming Ma, and Tie-Yan Liu. Asynchronous stochastic gradient descent with delay compensation. In *International conference on machine learning*, pp. 4120–4129. PMLR, 2017.

A TECHNICAL PRELIMINARIES

Implications Among the Assumptions.

Proof. First, the bounded gradient assumption implies bounded stochastic noise because

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

where we used $\mathbb{E}\left\|X-\mathbb{E}\left[X\right]\right\|^2 \leq \mathbb{E}\left\|X\right\|^2$. Thus, one may take $\sigma \leq G$.

Next, sample smoothness implies the centered increment bound

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

by the same variance inequality and sample smoothness. Hence, one may take $\sigma_L \leq L$.

Finally, Jensen's inequality and sample smoothness give

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

so the expected objective is also L-smooth. □

Smoothness can help us bound the gradient, but we get a different bound in each case.

Lemma A.1. *Let $f:\mathbb{R}^d\to\mathbb{R}$ be a function that satisfies assumption 2.1, then $\forall\mathbf{x}\in\mathbb{R}^d$ we have:*

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

Proof of Lemma A.1. Using L-smoothness, we get:

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

Where we get the equality by setting $\mathbf{y}=\mathbf{x}-\frac{1}{L}\nabla f(\mathbf{x})$. Using $f(\mathbf{y})\geq f^\star$ and a simple rearrangement yields the claim. □

The use of this lemma is to bound $\left\|\nabla f(\mathbf{x}_1)\right\|^2 \leq 2L\Delta$.

Lemma A.2. *Let $f:\mathcal{K}\to\mathbb{R}$ be a function that satisfies assumption 2.2, then $\forall\mathbf{x}\in\mathcal{K}$ we have:*

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

Proof of Lemma A.2. Let us define a new function:

$$h(\mathbf{x}) = f(\mathbf{x})-f^\star-\left\langle\nabla f(\mathbf{x}^\star),\mathbf{x}-\mathbf{x}^\star\right\rangle$$

Since f is convex and L-smooth, we know that:

$$0 \leq h(\mathbf{x}) \leq \frac{L}{2}\left\|\mathbf{x}-\mathbf{x}^\star\right\|^2$$

The gradient of this function is:

$$\nabla h(\mathbf{x}) = \nabla f(\mathbf{x})-\nabla f(\mathbf{x}^\star)$$

We can see that $h(\mathbf{x}^\star)=0, \nabla h(\mathbf{x}^\star)=0$, and that $\mathbf{x}^\star$ is the global minimum. The function h is also convex and L-smooth, since the gradient is the same as f up to a constant translation. We will add to the domain of h to include all $\mathbb{R}^d$, while still being convex and L-smooth. Since h is convex then:

$$h(\mathbf{y}) \geq h(\mathbf{x}^\star)+\langle\nabla h(\mathbf{x}^\star),\mathbf{y}-\mathbf{x}^\star\rangle = 0, \quad \forall\mathbf{y}\in\mathbb{R}^d$$

It is true even for points outside of the original domain, meaning that $\mathbf{x}^\star$ remains the global minimum even after this. For a smooth function, $\forall\mathbf{x},\mathbf{y}\in\mathbb{R}^d$:

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

By picking $\mathbf{y} = \mathbf{x} - \frac{1}{L}\nabla h(\mathbf{x})$, we get:

$$h(\mathbf{x}) - h(\mathbf{y}) \geq \frac{1}{2L} \|\nabla h(\mathbf{x})\|^2$$

Rearranging, we get:

$$\|\nabla h(\mathbf{x})\|^2 \leq 2L(h(\mathbf{x}) - h(\mathbf{y})) \leq 2L \cdot h(\mathbf{x})$$

By using $\mathbf{x} \in \mathcal{K}$ we get:

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

Since $\mathbf{x}^\star$ is the minimum point of f then:

$$\langle \nabla f(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle \geq 0, \quad \forall \mathbf{x} \in \mathcal{K}$$

Thus we get that:

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

□

B Non-Convex Analysis

Ideal Momentum. In the spirit of Mishchenko et al. (2022); Shi et al. (2024), we define the *ideal* momentum $\hat{\mathbf{m}}_t$ (Equation (7)), in which all updates are applied exactly at their designated iterations.

We will now bound the momentum:

Lemma B.1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function that satisfies assumption 2.1, and define the stochastic error as $\boldsymbol{\epsilon}_t := \mathbf{m}_t - \nabla f(\mathbf{x}_t)$. Consider the update rule $\mathbf{x}_{t+1} := \mathbf{x}_t - \eta \mathbf{m}_t$ with a learning rate $\eta \leq \frac{1}{2L}$, then:*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\mathbf{m}_t\|^2 \leq \frac{4\Delta}{\eta T} + \frac{2}{T}\sum_{t=1}^{T} \mathbb{E}\|\boldsymbol{\epsilon}_t\|^2$$

Proof of Lemma B.1. By the smoothness property of f, it follows since,

$$\begin{aligned} f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) &\leq \langle \nabla f(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle + \frac{L}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 = -\eta \langle \nabla f(\mathbf{x}_t), \mathbf{m}_t \rangle + \frac{L\eta^2}{2}\|\mathbf{m}_t\|^2 \\ &= -\eta \|\mathbf{m}_t\|^2 + \eta \langle \boldsymbol{\epsilon}_t, \mathbf{m}_t \rangle + \frac{L\eta^2}{2}\|\mathbf{m}_t\|^2 \\ &\leq -\eta \|\mathbf{m}_t\|^2 + \frac{\eta}{2}\|\boldsymbol{\epsilon}_t\|^2 + \frac{\eta}{2}\|\mathbf{m}_t\|^2 + \frac{L\eta^2}{2}\|\mathbf{m}_t\|^2 \leq -\frac{\eta}{4}\|\mathbf{m}_t\|^2 + \frac{\eta}{2}\|\boldsymbol{\epsilon}_t\|^2 \end{aligned}$$

Here, the first equality employs the definition of the update rule for $\mathbf{x}_{t+1}$ with learning rate η, and the second follows the definition of the stochastic error $\boldsymbol{\epsilon}_t := \mathbf{m}_t - \nabla f(\mathbf{x}_t)$. The next inequality uses Young's inequality $\langle \mathbf{a}, \mathbf{b} \rangle \leq \frac{1}{2}\|\mathbf{a}\|^2 + \frac{1}{2}\|\mathbf{b}\|^2, \forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ and the last inequality holds by setting $\eta \leq \frac{1}{2L}$.

Rearranging the above, we obtain:

$$\|\mathbf{m}_t\|^2 \leq 4\frac{f(\mathbf{x}_t) - f(\mathbf{x}_{t+1})}{\eta} + 2\|\boldsymbol{\epsilon}_t\|^2$$

Averaging over all T iterations and taking the expectation gives us:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\mathbf{m}_t\|^2 \leq \frac{4}{\eta T}\sum_{t=1}^{T} \mathbb{E}\left[f(\mathbf{x}_t) - f(\mathbf{x}_{t+1})\right] + \frac{2}{T}\sum_{t=1}^{T} \mathbb{E}\|\boldsymbol{\epsilon}_t\|^2 \leq \frac{4\Delta}{\eta T} + \frac{2}{T}\sum_{t=1}^{T} \mathbb{E}\|\boldsymbol{\epsilon}_t\|^2$$

□

A similar bound will be used for the gradients:

Lemma B.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function that satisfies assumption 2.1, and define the stochastic error as $\boldsymbol{\epsilon}_t := \mathbf{m}_t - \nabla f(\mathbf{x}_t)$. Consider the update rule $\mathbf{x}_{t+1} := \mathbf{x}_t - \eta \mathbf{m}_t$ with a learning rate $\eta \leq \frac{1}{L}$, then:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{2\Delta}{\eta T} + \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2$$

Proof of Lemma B.2. By the smoothness property of f, it follows since,

$$\begin{aligned}
f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq& \left\langle \nabla f(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \right\rangle + \frac{L}{2}\left\|\mathbf{x}_{t+1} - \mathbf{x}_t\right\|^2 \\
=& -\eta \left\langle \nabla f(\mathbf{x}_t), \nabla f(\mathbf{x}_t) + \boldsymbol{\epsilon}_t \right\rangle + \frac{L\eta^2}{2}\left\|\nabla f(\mathbf{x}_t) + \boldsymbol{\epsilon}_t\right\|^2 \\
=& \frac{\eta}{2}\left(-(2 - L\eta)\left\|\nabla f(\mathbf{x}_t)\right\|^2 - 2(1 - L\eta)\left\langle \nabla f(\mathbf{x}_t), \boldsymbol{\epsilon}_t \right\rangle + L\eta\left\|\boldsymbol{\epsilon}_t\right\|^2\right) \\
\leq& \frac{\eta}{2}\left(((1 - L\eta) - (2 - L\eta))\left\|\nabla f(\mathbf{x}_t)\right\|^2 + ((1 - L\eta) + L\eta)\left\|\boldsymbol{\epsilon}_t\right\|^2\right) \\
=& -\frac{\eta}{2}\left\|\nabla f(\mathbf{x}_t)\right\|^2 + \frac{\eta}{2}\left\|\boldsymbol{\epsilon}_t\right\|^2
\end{aligned}$$

Here, the first equality employs the definition of the update rule for $\mathbf{x}_{t+1}$ with learning rate η, and the second follows the definition of the stochastic error $\boldsymbol{\epsilon}_t := \mathbf{m}_t - \nabla f(\mathbf{x}_t)$. The second inequality uses Young's inequality $\langle \mathbf{a}, \mathbf{b} \rangle \leq \frac{1}{2}\left\|\mathbf{a}\right\|^2 + \frac{1}{2}\left\|\mathbf{b}\right\|^2, \forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ and remembering that $1 - L\eta \geq 0$.

Rearranging the above, we obtain:

$$\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq 2\frac{f(\mathbf{x}_t) - f(\mathbf{x}_{t+1})}{\eta} + \left\|\boldsymbol{\epsilon}_t\right\|^2$$

Averaging over all T iterations and taking the expectation gives us:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{2}{\eta T}\sum_{t=1}^{T}\mathbb{E}\left[f(\mathbf{x}_t) - f(\mathbf{x}_{t+1})\right] + \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq \frac{2\Delta}{\eta T} + \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2$$

□

Lemma B.3. *We will define the expected weighted gradient $\bar{\mathbf{g}}_{t,k} := \beta(1-\beta)^{t-k}\nabla f(\mathbf{x}_k)$, and the expected error $\bar{\boldsymbol{\epsilon}}_t := \sum_{k=1}^{t}\bar{\mathbf{g}}_{t,k} - \nabla f(\mathbf{x}_t)$. Then:*

$$\frac{1}{T}\sum_{t=1}^{T}\left\|\bar{\boldsymbol{\epsilon}}_t\right\|^2 \leq \frac{2L\Delta}{\beta T} + \frac{L^2\eta^2}{\beta^2 T}\sum_{t=1}^{T}\left\|\mathbf{m}_t\right\|^2$$

Proof of Lemma B.3. To bound the expected error we will find its update step:

$$\begin{aligned}
\bar{\boldsymbol{\epsilon}}_t =& \sum_{k=1}^{t}\bar{\mathbf{g}}_{t,k} - \nabla f(\mathbf{x}_t) = (1-\beta)\sum_{k=1}^{t-1}\bar{\mathbf{g}}_{t-1,k} - (1-\beta)\nabla f(\mathbf{x}_t) \\
=& (1-\beta)\left(\bar{\boldsymbol{\epsilon}}_{t-1} + \nabla f(\mathbf{x}_{t-1}) - \nabla f(\mathbf{x}_t)\right)
\end{aligned}$$

Where we used $\bar{\mathbf{g}}_{t,t} = \beta\nabla f(\mathbf{x}_t)$ and $\bar{\mathbf{g}}_{t,k} = (1-\beta)\bar{\mathbf{g}}_{t-1,k}$. Using this update rule, we get:

$$\begin{aligned}
\left\|\bar{\boldsymbol{\epsilon}}_{t+1}\right\|^2 =& (1-\beta)^2\left\|\bar{\boldsymbol{\epsilon}}_t + \nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t+1})\right\|^2 \\
\leq& (1-\beta)\left\|\bar{\boldsymbol{\epsilon}}_t\right\|^2 + \frac{(1-\beta)^2}{\beta}\left\|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t+1})\right\|^2 \\
\leq& (1-\beta)\left\|\bar{\boldsymbol{\epsilon}}_t\right\|^2 + \frac{(1-\beta)^2 L^2}{\beta}\left\|\mathbf{x}_t - \mathbf{x}_{t+1}\right\|^2 \leq (1-\beta)\left\|\bar{\boldsymbol{\epsilon}}_t\right\|^2 + \frac{L^2\eta^2}{\beta}\left\|\mathbf{m}_t\right\|^2
\end{aligned}$$

Where we used the inequality $\|\mathbf{x}+\mathbf{y}\|^2 \leq \frac{1}{p}\|\mathbf{x}\|^2 + \frac{1}{1-p}\|\mathbf{y}\|^2, \forall \mathbf{x},\mathbf{y} \in \mathbb{R}^d, p \in (0,1)$, with $p = 1-\beta$, the smoothness of f, and the update rule $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{m}_t$. Rearranging the terms and summing over from 1 to $T-1$ yields:

$$\|\bar{\boldsymbol{\epsilon}}_T\|^2 - \|\bar{\boldsymbol{\epsilon}}_1\|^2 + \beta \sum_{t=1}^{T-1} \|\bar{\boldsymbol{\epsilon}}_t\|^2 \leq \frac{L^2\eta^2}{\beta} \sum_{t=1}^{T-1} \|\mathbf{m}_t\|^2 \leq \frac{L^2\eta^2}{\beta} \sum_{t=1}^{T} \|\mathbf{m}_t\|^2$$

Where we added another term to the sum at the end. Since $\bar{\boldsymbol{\epsilon}}_1 = -(1-\beta)\nabla f(\mathbf{x}_1)$, we get:

$$\begin{aligned}\beta \sum_{t=1}^{T} \|\bar{\boldsymbol{\epsilon}}_t\|^2 \leq& (1-\beta)^2 \|\nabla f(\mathbf{x}_1)\|^2 - (1-\beta)\|\bar{\boldsymbol{\epsilon}}_T\|^2 + \frac{L^2\eta^2}{\beta} \sum_{t=1}^{T} \|\mathbf{m}_t\|^2 \\ \leq& 2L\Delta + \frac{L^2\eta^2}{\beta} \sum_{t=1}^{T} \|\mathbf{m}_t\|^2\end{aligned}$$

Dividing by βT:

$$\frac{1}{T}\sum_{t=1}^{T} \|\bar{\boldsymbol{\epsilon}}_t\|^2 \leq \frac{2L\Delta}{\beta T} + \frac{L^2\eta^2}{\beta^2 T} \sum_{t=1}^{T} \|\mathbf{m}_t\|^2$$

□

Data Independent Case. We will bound the error in the data independent case:

Lemma B.4. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function that satisfies assumptions 2.1 and 2.3. Then, for any iteration $t \geq 1$ with momentum parameter $\beta \leq \frac{1}{8(M-1)}$, the cumulative stochastic error of the delayed momentum in Algorithm 1, $\boldsymbol{\epsilon}_t := \mathbf{m}_t - \nabla f(\mathbf{x}_t)$, evaluated over T iterations, satisfies:*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\boldsymbol{\epsilon}_t\|^2 \leq \frac{4L\Delta}{\beta T} + \beta\sigma^2 + \frac{2L^2\eta^2}{\beta^2 T} \sum_{t=1}^{T} \mathbb{E}\|\mathbf{m}_t\|^2 + \frac{1}{4T}\sum_{t=1}^{T} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|^2$$

Proof of Lemma B.4. Since the data is independent of the delay, we can easily separate the bias and variance of $\boldsymbol{\epsilon}_t$, given $\mathcal{T}(t)$:

$$\begin{aligned}\mathbb{E}\|\boldsymbol{\epsilon}_t\|^2 =& \mathbb{E}\left\|\sum_{k\in[t]\setminus\mathcal{T}(t)} \mathbf{g}_{t,k} - \nabla f(\mathbf{x}_t)\right\|^2 \\ =& \mathbb{E}\left\|\sum_{k\in[t]\setminus\mathcal{T}(t)} \bar{\mathbf{g}}_{t,k} - \nabla f(\mathbf{x}_t)\right\|^2 + \sum_{k\in[t]\setminus\mathcal{T}(t)} \mathbb{E}\|\mathbf{g}_{t,k} - \bar{\mathbf{g}}_{t,k}\|^2\end{aligned}$$

The second sum can be bounded using the bounded variance assumption:

$$\sum_{k\in[t]\setminus\mathcal{T}(t)} \mathbb{E}\|\mathbf{g}_{t,k} - \bar{\mathbf{g}}_{t,k}\|^2 \leq \beta^2\sigma^2 \sum_{k\in[t]\setminus\mathcal{T}(t)} (1-\beta)^{2(t-k)} \leq \beta^2\sigma^2 \sum_{k=1}^{t} (1-\beta)^{2(t-k)} \leq \beta\sigma^2$$

Where at the end, we bounded the geometric sum. Now we need to bound the first sum.

$$\mathbb{E}\left\|\sum_{k\in[t]\setminus\mathcal{T}(t)} \bar{\mathbf{g}}_{t,k} - \nabla f(\mathbf{x}_t)\right\|^2 = \mathbb{E}\left\|\bar{\boldsymbol{\epsilon}}_t - \sum_{k\in\mathcal{T}(t)} \bar{\mathbf{g}}_{t,k}\right\|^2 \leq 2\mathbb{E}\|\bar{\boldsymbol{\epsilon}}_t\|^2 + 2\mathbb{E}\left\|\sum_{k\in\mathcal{T}(t)} \bar{\mathbf{g}}_{t,k}\right\|^2$$

Where we used $\|\mathbf{a}+\mathbf{b}\|^2 \leq 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$. We will bound the second term:

$$\begin{aligned}\left\|\sum_{k\in\mathcal{T}(t)} \bar{\mathbf{g}}_{t,k}\right\|^2 \leq& (M-1)\sum_{k\in\mathcal{T}(t)} \|\bar{\mathbf{g}}_{t,k}\|^2 \leq (M-1)\beta^2 \sum_{k=1}^{t} (1-\beta)^{2(t-k)} \|\nabla f(\mathbf{x}_k)\|^2 \\ \leq& \frac{\beta}{8}\sum_{k=1}^{t} (1-\beta)^{2(t-k)} \|\nabla f(\mathbf{x}_k)\|^2\end{aligned}$$

Where the first inequality uses Jensen's inequality and the fact that $|\mathcal{T}(t)| \leq M-1$. In the next inequality we input the definitions of $\bar{\mathbf{g}}_{t,k}$ and add the missing members in the sum. At last, we use the bound $(M-1)\beta \leq \frac{1}{8}$. In total, we get:

$$\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq 2\mathbb{E}\left\|\bar{\boldsymbol{\epsilon}}_t\right\|^2 + \frac{\beta}{4}\sum_{k=1}^{t}(1-\beta)^{2(t-k)}\mathbb{E}\left\|\nabla f(\mathbf{x}_k)\right\|^2 + \beta\sigma^2$$

Averaging over t, we get:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq \frac{2}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\bar{\boldsymbol{\epsilon}}_t\right\|^2 + \frac{\beta}{4T}\sum_{t=1}^{T}\sum_{k=1}^{t}(1-\beta)^{2(t-k)}\mathbb{E}\left\|\nabla f(\mathbf{x}_k)\right\|^2 + \beta\sigma^2$$

The first terms can be bounded using Lemma B.3. For the second term, swapping the order of the sum and solving the geometric sum, gives us $\frac{1}{4T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2$. In total, we get:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq \frac{4L\Delta}{\beta T} + \beta\sigma^2 + \frac{2L^2\eta^2}{\beta^2 T}\sum_{t=1}^{T}\mathbb{E}\left\|\mathbf{m}_t\right\|^2 + \frac{1}{4T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2$$

□

Proof of Theorem 3.1. Starting from Lemma B.4, we input Lemmas B.1 and B.2:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq & \frac{4L\Delta}{\beta T} + \beta\sigma^2 + \frac{2L^2\eta^2}{\beta^2 T}\sum_{t=1}^{T}\mathbb{E}\left\|\mathbf{m}_t\right\|^2 + \frac{1}{4T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \\ \leq & \frac{4L\Delta}{\beta T} + \beta\sigma^2 + \frac{8L^2\Delta\eta}{\beta^2 T} + \frac{4L^2\eta^2}{\beta^2 T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 + \frac{\Delta}{2\eta T} + \frac{1}{4T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2\end{aligned}$$

Using $\eta \leq \frac{\beta}{4L}$, we get:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq \frac{4L\Delta}{\beta T} + \beta\sigma^2 + \frac{8L^2\Delta\eta}{\beta^2 T} + \frac{\Delta}{2\eta T} + \frac{1}{2T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2$$

Self bounding gives us:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq \frac{8L\Delta}{\beta T} + 2\beta\sigma^2 + \frac{16L^2\Delta\eta}{\beta^2 T} + \frac{\Delta}{\eta T}$$

Inputting this result in Lemma B.2, gives us:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{2\Delta}{\eta T} + \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq \frac{8L\Delta}{\beta T} + 2\beta\sigma^2 + \frac{16L^2\Delta\eta}{\beta^2 T} + \frac{3\Delta}{\eta T}$$

Inputting $\eta = \frac{\beta}{4L}$, we get:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{24L\Delta}{\beta T} + 2\beta\sigma^2$$

Picking $\beta = \min\left\{\frac{1}{8(M-1)}, \frac{2\sqrt{3L\Delta}}{\sigma\sqrt{T}}\right\}$, we get:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{192L\Delta(M-1)}{T} + \frac{8\sigma\sqrt{3L\Delta}}{\sqrt{T}}$$

□

Data Dependent Case. We will bound the error in the data dependent case:

Lemma B.5. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function that satisfies assumptions 2.1 and 2.4. Then, for any iteration $t \geq 1$ with momentum parameter $\beta \in (0, 1)$, the cumulative stochastic error of the momentum in Algorithm 1, evaluated over T iterations, satisfies:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq \frac{4L\Delta}{\beta T} + 2\beta\sigma^2 + 2\beta^2(M-1)^2G^2 + \frac{2L^2\eta^2}{\beta^2 T}\sum_{t=1}^{T}\mathbb{E}\left\|\mathbf{m}_t\right\|^2$$

Proof of Lemma B.5. We will define the delay $\mathbf{b}_t := \mathbf{m}_t - \hat{\mathbf{m}}_t = \boldsymbol{\epsilon}_t - \hat{\boldsymbol{\epsilon}}_t$, and note Equation (8):

$$\mathbf{b}_t = -\sum_{k\in\mathcal{T}(t)}\mathbf{g}_{t,k}$$

Next, we use the decomposition $\boldsymbol{\epsilon}_t = \hat{\boldsymbol{\epsilon}}_t + \mathbf{b}_t$ and the inequality $\left\|\mathbf{x}+\mathbf{y}\right\|^2 \leq 2\left\|\mathbf{x}\right\|^2 + 2\left\|\mathbf{y}\right\|^2$, we obtain $\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq 2\mathbb{E}\left\|\hat{\boldsymbol{\epsilon}}_t\right\|^2 + 2\mathbb{E}\left\|\mathbf{b}_t\right\|^2$.

$$\begin{aligned}\left\|\mathbf{b}_t\right\| =& \left\|-\sum_{k\in\mathcal{T}(t)}\mathbf{g}_{t,k}\right\| \leq \sum_{k\in\mathcal{T}(t)}\left\|\mathbf{g}_{t,k}\right\| = \sum_{k\in\mathcal{T}(t)}\beta(1-\beta)^{t-k}\left\|\nabla f(\mathbf{x}_k;\mathbf{z}_k)\right\| \\ \leq& \beta G\sum_{k\in\mathcal{T}(t)}(1-\beta)^{t-k} \leq \beta(M-1)G\end{aligned}$$

Where the first inequality is the triangle inequality, the second inequality is the Lipchitz assumption, and in the last inequality we bound $(1-\beta) \leq 1$ and $|\mathcal{T}(t)| \leq M-1$.

Using Equation (7), we know that the ideal stochastic error is:

$$\hat{\boldsymbol{\epsilon}}_t := \hat{\mathbf{m}}_t - \nabla f(\mathbf{x}_t) = \sum_{k=1}^{t}\mathbf{g}_{t,k} - \nabla f(\mathbf{x}_t)$$

We will define the expected stochastic error as:

$$\bar{\boldsymbol{\epsilon}}_t = \sum_{k=1}^{t}\bar{\mathbf{g}}_{t,k} - \nabla f(\mathbf{x}_t)$$

Since $\hat{\boldsymbol{\epsilon}}_t$ has all the terms, it is not affected by the delay, so we don't care that the delay depends on the data. We can tell that $\mathbb{E}\left[\hat{\boldsymbol{\epsilon}}_t\right] = \bar{\boldsymbol{\epsilon}}_t$ and thus $\mathbb{E}\left\|\hat{\boldsymbol{\epsilon}}_t\right\|^2 = \mathbb{E}\left\|\hat{\boldsymbol{\epsilon}}_t - \bar{\boldsymbol{\epsilon}}_t\right\|^2 + \mathbb{E}\left\|\bar{\boldsymbol{\epsilon}}_t\right\|^2$.

$$\begin{aligned}&\mathbb{E}\left\|\hat{\boldsymbol{\epsilon}}_t - \bar{\boldsymbol{\epsilon}}_t\right\|^2 = \mathbb{E}\left\|\sum_{k=1}^{t}\left(\mathbf{g}_{t,k} - \bar{\mathbf{g}}_{t,k}\right)\right\|^2 = \sum_{k=1}^{t}\mathbb{E}\left\|\mathbf{g}_{t,k} - \bar{\mathbf{g}}_{t,k}\right\|^2 \\ =& \sum_{k=1}^{t}\beta^2(1-\beta)^{2(t-k)}\mathbb{E}\left\|\mathbf{g}_k - \nabla f(\mathbf{x}_k)\right\|^2 \leq \beta^2\sigma^2\sum_{k=1}^{t}(1-\beta)^{2(t-k)} \leq \beta\sigma^2\end{aligned}$$

Where we used the fact that the sum is a martingale sequence, in the first inequality we used the bounded variance, and finally bounded the geometric sum. In total, we get:

$$\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq 2\mathbb{E}\left\|\bar{\boldsymbol{\epsilon}}_t\right\|^2 + 2\beta\sigma^2 + 2\beta^2(M-1)^2G^2$$

Averaging over t and using Lemma B.3, we get:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq \frac{4L\Delta}{\beta T} + 2\beta\sigma^2 + 2\beta^2(M-1)^2G^2 + \frac{2L^2\eta^2}{\beta^2 T}\sum_{t=1}^{T}\mathbb{E}\left\|\mathbf{m}_t\right\|^2$$

□

Proof of Theorem 3.2. Starting from Lemma B.5, we input Lemma B.1:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq & \frac{4L\Delta}{\beta T}+2\beta\sigma^2+2\beta^2(M-1)^2G^2+\frac{2L^2\eta^2}{\beta^2 T}\sum_{t=1}^{T}\mathbb{E}\left\|\mathbf{m}_t\right\|^2 \\ \leq & \frac{4L\Delta}{\beta T}+2\beta\sigma^2+2\beta^2(M-1)^2G^2+\frac{8L^2\Delta\eta}{\beta^2 T}+\frac{4L^2\eta^2}{\beta^2 T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2\end{aligned}$$

Using $\eta \leq \frac{\beta}{2\sqrt{2}L}$ will yield the following:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq \frac{4L\Delta}{\beta T}+2\beta\sigma^2+2\beta^2(M-1)^2G^2+\frac{8L^2\Delta\eta}{\beta^2 T}+\frac{1}{2T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2$$

Moving sides and multiplying by 2:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq \frac{8L\Delta}{\beta T}+4\beta\sigma^2+4\beta^2(M-1)^2G^2+\frac{16L^2\Delta\eta}{\beta^2 T}$$

Inputting this result in Lemma B.2, gives us:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq & \frac{2\Delta}{\eta T}+\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \\ \leq & \frac{8L\Delta}{\beta T}+4\beta\sigma^2+4\beta^2(M-1)^2G^2+\frac{16L^2\Delta\eta}{\beta^2 T}+\frac{2\Delta}{\eta T}\end{aligned}$$

Inputting $\eta = \frac{\beta}{2\sqrt{2}L}$, we get:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{8(1+\sqrt{2})L\Delta}{\beta T}+4\beta\sigma^2+4\beta^2(M-1)^2G^2$$

Bounding $\sqrt{2} \leq 3/2$, we get:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{20L\Delta}{\beta T}+4\beta\sigma^2+4\beta^2(M-1)^2G^2$$

Choosing $\beta = \min\left\{\frac{1}{2}, \frac{\sqrt{5L\Delta}}{\sigma\sqrt{T}}, \left(\frac{5L\Delta}{2G^2(M-1)^2T}\right)^{1/3}\right\}$, we get:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla f(\mathbf{x}_t)\right\|^2 \leq \frac{40L\Delta}{T}+\frac{8\sigma\sqrt{5L\Delta}}{\sqrt{T}}+3\left(\frac{20GL\Delta(M-1)}{T}\right)^{2/3}$$

The condition $\beta \leq 0.5$ added to its value is to guarantee that $\beta < 1$, which was assumed throughout the proof, since none of the other conditions on β guarantee it. □

C Convex Analysis

Theorem C.1 (Cutkosky (2019)). *Let $f : \mathcal{K} \to \mathbb{R}$ be a convex function. Also let $\{\alpha_t > 0\}_t$ and $\{\mathbf{w}_t \in \mathcal{K}\}_t$. Let $\{\mathbf{x}_t\}$ be the $\{\alpha_\tau\}_{\tau=1}^t$ weighted average of $\{\mathbf{w}_\tau\}_{\tau=1}^t$, meaning: $\alpha_{1:t}\mathbf{x}_t = \sum_{\tau=1}^t \alpha_\tau \mathbf{w}_\tau$. Then the following holds for all $t \geq 1, x \in \mathcal{K}$:*

$$\alpha_{1:t}(f(\mathbf{x}_t) - f(\mathbf{x})) \leq \sum_{\tau=1}^{t} \alpha_\tau \left\langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{x}\right\rangle$$

Concretely, the theorem implies that the excess loss of the weighted average $\mathbf{x}_t$ can be related to the weighted regret $\sum_{\tau=1}^t \alpha_\tau \left\langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{x}\right\rangle$.

Proof of Theorem C.1. We will prove by induction.

Induction Basis: For $t = 1$, we get that:

$$\alpha_1(f(\mathbf{x}_1) - f(\mathbf{x})) \leq \alpha_1 \left\langle \nabla f(\mathbf{x}_1), \mathbf{x}_1 - \mathbf{x} \right\rangle = \alpha_1 \left\langle \nabla f(\mathbf{x}_1), \mathbf{w}_1 - \mathbf{x} \right\rangle$$

Where the inequality is the convexity of f, and the equality is because $\mathbf{w}_1 = \mathbf{x}_1$.

Induction Assumption: We will assume that Theorem C.1 hold true for some $t \geq 1$, meaning:

$$\alpha_{1:t}(f(\mathbf{x}_t) - f(\mathbf{x})) \leq \sum_{\tau=1}^{t} \alpha_\tau \left\langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{x} \right\rangle$$

Induction Step: We will prove for $t + 1$.

$$\begin{aligned}\alpha_{1:t+1}(f(\mathbf{x}_{t+1}) - f(\mathbf{x})) =& \alpha_{1:t}(f(\mathbf{x}_t) - f(\mathbf{x})) \\ &+ \alpha_{1:t}(f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t)) + \alpha_{t+1}(f(\mathbf{x}_{t+1}) - f(\mathbf{x}))\end{aligned}$$

We got this by adding and subtracting $\alpha_{1:t} f(\mathbf{x}_t)$. The first term can be bounded by the induction assumption, and the second and third step will each be bounded using the convexity of f.

$$\begin{aligned}\alpha_{1:t+1}(f(\mathbf{x}_{t+1}) - f(\mathbf{x})) = \sum_{\tau=1}^{t} \alpha_\tau \left\langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{x} \right\rangle \\ + \alpha_{1:t} \left\langle \nabla f(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{x}_t \right\rangle + \alpha_{t+1} \left\langle \nabla f(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{x} \right\rangle\end{aligned}$$

Combining the last two terms, we get:

$$\begin{aligned}&\alpha_{1:t} \left\langle \nabla f(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{x}_t \right\rangle + \alpha_{t+1} \left\langle \nabla f(\mathbf{x}_{t+1}), \mathbf{x}_{t+1} - \mathbf{x} \right\rangle \\ &= \left\langle \nabla f(\mathbf{x}_{t+1}), \alpha_{1:t+1}\mathbf{x}_{t+1} - \alpha_{1:t}\mathbf{x}_t - \alpha_{t+1}\mathbf{x} \right\rangle = \alpha_{t+1} \left\langle \nabla f(\mathbf{x}_{t+1}), \mathbf{w}_{t+1} - \mathbf{x} \right\rangle\end{aligned}$$

Where in the last term we used $\alpha_{1:t+1}\mathbf{x}_{t+1} - \alpha_{1:t}\mathbf{x}_t = \alpha_{t+1}\mathbf{w}_{t+1}$. In total, we got:

$$\alpha_{1:t+1}(f(\mathbf{x}_{t+1}) - f(\mathbf{x})) \leq \sum_{\tau=1}^{t+1} \alpha_\tau \left\langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{x} \right\rangle$$

We proved the induction step. □

We will denote the weighted momentum $\mathbf{q}_t := \alpha_t \mathbf{d}_t$.

Lemma C.2. *Let $\eta > 0$, and $\mathcal{K} \subset \mathbb{R}^d$ be a convex domain of bounded diameter D, also let $\{\mathbf{q}_t \in \mathbb{R}^d\}_{t=1}^T$ be a sequence of arbitrary vectors. Then for any starting point $\mathbf{w}_1 \in \mathcal{K}$, and an update rule $\mathbf{w}_{t+1} = \Pi_{\mathcal{K}}\left(\mathbf{w}_t - \eta \mathbf{q}_t\right),\ \forall t \geq 1$, the following holds $\forall \mathbf{x} \in \mathcal{K}$:*

$$\sum_{\tau=1}^{t} \left\langle \mathbf{q}_\tau, \mathbf{w}_{\tau+1} - \mathbf{x} \right\rangle \leq \frac{D^2}{2\eta} - \frac{1}{2\eta} \sum_{\tau=1}^{t} \left\| \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \right\|^2$$

Proof of Lemma C.2. The update rule $\mathbf{w}_{t+1} = \Pi_{\mathcal{K}}\left(\mathbf{w}_t - \eta \mathbf{q}_t\right)$ can be re-written as a convex optimization problem over $\mathcal{K}$:

$$\begin{aligned}\mathbf{w}_{t+1} =& \Pi_{\mathcal{K}}\left(\mathbf{w}_t - \eta \mathbf{q}_t\right) = \arg\min_{\mathbf{x} \in \mathcal{K}} \left\{ \left\| \mathbf{w}_t - \eta \mathbf{q}_t - \mathbf{x} \right\|^2 \right\} \\ =& \arg\min_{\mathbf{x} \in \mathcal{K}} \left\{ \left\langle \mathbf{q}_t, \mathbf{x} - \mathbf{w}_t \right\rangle + \frac{1}{2\eta} \left\| \mathbf{x} - \mathbf{w}_t \right\|^2 \right\}\end{aligned}$$

The first equality is our update definition, the second is by the definition of the projection operator, and then we rewrite it in a way that does not affect the minimum point.

Now, since $\mathbf{w}_{t+1}$ is the minimal point of the above convex problem, then from optimality conditions we obtain:

$$\left\langle \mathbf{q}_t + \frac{1}{\eta}(\mathbf{w}_{t+1} - \mathbf{w}_t), \mathbf{x} - \mathbf{w}_{t+1} \right\rangle \geq 0, \quad \forall \mathbf{x} \in \mathcal{K}$$

Re-arranging the above, we get that:

$$\langle \mathbf{q}_t, \mathbf{w}_{t+1} - \mathbf{x} \rangle \leq \frac{1}{\eta} \langle \mathbf{w}_t - \mathbf{w}_{t+1}, \mathbf{w}_{t+1} - \mathbf{x} \rangle$$
$$= \frac{1}{2\eta} \|\mathbf{w}_t - \mathbf{x}\|^2 - \frac{1}{2\eta} \|\mathbf{w}_{t+1} - \mathbf{x}\|^2 - \frac{1}{2\eta} \|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2$$

Where the equality is an algebraic manipulation. After summing over t we get:

$$\sum_{\tau=1}^{t} \langle \mathbf{q}_\tau, \mathbf{w}_{\tau+1} - \mathbf{x} \rangle \leq \frac{1}{2\eta} \sum_{\tau=1}^{t} \left(\|\mathbf{w}_\tau - \mathbf{x}\|^2 - \|\mathbf{w}_{\tau+1} - \mathbf{x}\|^2 - \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 \right)$$
$$= \frac{\|\mathbf{w}_1 - \mathbf{x}\|^2 - \|\mathbf{w}_{t+1} - \mathbf{x}\|^2}{2\eta} - \frac{1}{2\eta} \sum_{\tau=1}^{t} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 \leq \frac{D^2}{2\eta} - \frac{1}{2\eta} \sum_{\tau=1}^{t} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2$$

Where the second line is due to splitting the sum into two sums, and using the fact that the first one is a telescopic sum, and lastly, we use the diameter of $\mathcal{K}$. This establishes the lemma. □

These two lemmas will help us bound the excess loss for an Anytime algorithm.

Lemma C.3. *Let $f : \mathcal{K} \to \mathbb{R}$ be a convex and smooth function with smoothness parameter $L > 0$, and define the stochastic error as $\boldsymbol{\epsilon}_t := \mathbf{q}_t - \alpha_t \nabla f(\mathbf{x}_t)$. Consider the Anytime-GD update rule $\mathbf{w}_{t+1} := \Pi_{\mathcal{K}} (\mathbf{w}_t - \eta \mathbf{q}_t), \mathbf{x}_{t+1} = \mathbf{x}_t + \frac{\alpha_{t+1}}{\alpha_{1:t+1}} (\mathbf{w}_{t+1} - \mathbf{x}_t)$ with $\{\alpha_t = t\}_{t=1}^{T}$ and learning rate $\eta \leq \frac{1}{4LT}$, then:*

$$\mathbb{E}[f(\mathbf{x}_T)] - f^\star \leq \frac{2D^2}{\eta T^2} + \frac{4DG^\star}{T} + \frac{4D}{T^2} \sum_{t=1}^{T} \mathbb{E} \|\boldsymbol{\epsilon}_t\|$$

Where $\mathbf{x}^\star := \arg\min_{\mathbf{x} \in \mathcal{K}} f(\mathbf{x})$, and $G^\star := \|\nabla f(\mathbf{x}^\star)\|$.

Proof of Lemma C.3. Define the expected excess loss at time t as $\Delta_t := \mathbb{E}[f(\mathbf{x}_t)] - f^\star$. Following the analysis framework of Reshef & Levy (2024; 2025), we begin by deriving an upper bound on Δ_t:

$$\alpha_{1:t} \Delta_t := \alpha_{1:t} (\mathbb{E}[f(\mathbf{x}_t)] - f^\star) \leq \sum_{\tau=1}^{t} \alpha_\tau \mathbb{E} \langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{x}^\star \rangle$$
$$= \sum_{\tau=1}^{t} \alpha_\tau \mathbb{E} \langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_{\tau+1} - \mathbf{x}^\star \rangle + \sum_{\tau=1}^{t} \alpha_\tau \mathbb{E} \langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle$$
$$= \sum_{\tau=1}^{t} \mathbb{E} \langle \mathbf{q}_\tau, \mathbf{w}_{\tau+1} - \mathbf{x}^\star \rangle + \sum_{\tau=1}^{t} \mathbb{E} \langle \boldsymbol{\epsilon}_\tau, \mathbf{x}^\star - \mathbf{w}_{\tau+1} \rangle + \sum_{\tau=1}^{t} \alpha_\tau \mathbb{E} \langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle$$
$$\leq \frac{D^2}{2\eta} - \frac{1}{2\eta} \sum_{\tau=1}^{t} \mathbb{E} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 + \sum_{\tau=1}^{t} \mathbb{E} \langle \boldsymbol{\epsilon}_\tau, \mathbf{x}^\star - \mathbf{w}_{\tau+1} \rangle + \sum_{\tau=1}^{t} \alpha_\tau \mathbb{E} \langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle$$
$$\leq \frac{D^2}{2\eta} + \sum_{\tau=1}^{t} \left(\alpha_\tau \mathbb{E} \langle \nabla f(\mathbf{x}_\tau) - \nabla f(\mathbf{x}^\star), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle - \frac{1}{2\eta} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 \right)$$
$$+ \sum_{\tau=1}^{t} \mathbb{E} \langle \alpha_\tau \nabla f(\mathbf{x}^\star), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle + \sum_{\tau=1}^{t} \mathbb{E} \langle \boldsymbol{\epsilon}_\tau, \mathbf{x}^\star - \mathbf{w}_{\tau+1} \rangle$$

The first inequality is Theorem C.1, then we add and subtract $w_{\tau+1}$ and split it into two sums, split the first sum using $\mathbf{q}_t = \alpha_t \nabla f(\mathbf{x}_t) + \boldsymbol{\epsilon}_t$, bound the first sum using Lemma C.2, and finally we add and subtract $\nabla f(\mathbf{x}^\star)$.

The first sum can be bounded as such:

$$\alpha_\tau \mathbb{E} \langle \nabla f(\mathbf{x}_\tau) - \nabla f(\mathbf{x}^\star), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle - \frac{1}{2\eta} \mathbb{E} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2$$
$$\leq \frac{\eta}{2} \alpha_\tau^2 \mathbb{E} \|\nabla f(\mathbf{x}_\tau) - \nabla f(\mathbf{x}^\star)\|^2 \leq 2L\eta\alpha_{1:\tau} \mathbb{E}[f(\mathbf{x}_\tau) - f^\star] = 2L\eta\alpha_{1:\tau}\Delta_\tau$$

Where at first we used Young's inequality, and then we bound $\alpha_\tau^2 \leq 2\alpha_{1:\tau}$ and use Lemma A.2.

For the second sum, we do this:

$$\sum_{\tau=1}^{t} \mathbb{E}\left\langle \alpha_\tau \nabla f(\mathbf{x}^\star), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \right\rangle = \sum_{\tau=1}^{t} (\alpha_\tau - \alpha_{\tau-1}) \mathbb{E}\left\langle \nabla f(\mathbf{x}^\star), \mathbf{w}_\tau \right\rangle - \alpha_t \mathbb{E}\left\langle \nabla f(\mathbf{x}^\star), \mathbf{w}_{t+1} \right\rangle$$

$$= \sum_{\tau=1}^{t} (\alpha_\tau - \alpha_{\tau-1}) \mathbb{E}\left\langle \nabla f(\mathbf{x}^\star), \mathbf{w}_\tau - \mathbf{w}_{t+1} \right\rangle \leq \sum_{\tau=1}^{t} (\alpha_\tau - \alpha_{\tau-1}) \left\| \nabla f(\mathbf{x}^\star) \right\| \mathbb{E}\left\| \mathbf{w}_\tau - \mathbf{w}_{t+1} \right\|$$

$$\leq D \left\| \nabla f(\mathbf{x}^\star) \right\| \sum_{\tau=1}^{t} (\alpha_\tau - \alpha_{\tau-1}) = \alpha_t D \left\| \nabla f(\mathbf{x}^\star) \right\| = \alpha_t D G^\star$$

The first equality is rearrangement of the sum while defining $\alpha_0 = 0$, then we put the last term into the sum, and use Cauchy-Schwarz, and finally, we use the diameter bound and telescope the sum.

For the third sum we just use Cauchy-Schwartz and the bounded diameter assumption:

$$\mathbb{E}\left\langle \boldsymbol{\epsilon}_\tau, \mathbf{x}^\star - \mathbf{w}_{\tau+1} \right\rangle \leq \mathbb{E}\left[\left\| \boldsymbol{\epsilon}_\tau \right\| \cdot \left\| \mathbf{x}^\star - \mathbf{w}_{\tau+1} \right\| \right] \leq D \mathbb{E}\left\| \boldsymbol{\epsilon}_\tau \right\|$$

In total, we get:

$$\alpha_{1:t} \Delta_t \leq 2L\eta \sum_{\tau=1}^{t} \alpha_{1:\tau} \Delta_\tau + \frac{D^2}{2\eta} + \alpha_t D G^\star + D \sum_{\tau=1}^{t} \mathbb{E}\left\| \boldsymbol{\epsilon}_\tau \right\|$$

Notice that if $\eta \leq \frac{1}{4LT}$, we get that:

$$\begin{aligned} \alpha_{1:t} \Delta_t &\leq \frac{1}{2T} \sum_{\tau=1}^{t} \alpha_{1:\tau} \Delta_\tau + \frac{D^2}{2\eta} + \alpha_t D G^\star + D \sum_{\tau=1}^{t} \mathbb{E}\left\| \boldsymbol{\epsilon}_\tau \right\| \\ &\leq \frac{1}{2T} \sum_{\tau=1}^{T} \alpha_{1:\tau} \Delta_\tau + \frac{D^2}{2\eta} + \alpha_T D G^\star + D \sum_{\tau=1}^{T} \mathbb{E}\left\| \boldsymbol{\epsilon}_\tau \right\| \end{aligned}$$

Averaging over t, we get:

$$\frac{1}{2T} \sum_{\tau=1}^{T} \alpha_{1:\tau} \Delta_\tau \leq \frac{D^2}{2\eta} + \alpha_T D G^\star + D \sum_{\tau=1}^{T} \mathbb{E}\left\| \boldsymbol{\epsilon}_\tau \right\|$$

Inputting this back into the previous bound, we get:

$$\alpha_{1:T} \Delta_T \leq \frac{D^2}{\eta} + 2\alpha_T D G^\star + 2D \sum_{\tau=1}^{T} \mathbb{E}\left\| \boldsymbol{\epsilon}_\tau \right\|$$

since $\alpha_{1:T} \geq \frac{T^2}{2}$, we get:

$$\Delta_T \leq \frac{2D^2}{\eta T^2} + \frac{4DG^\star}{T} + \frac{4D}{T^2} \sum_{t=1}^{T} \mathbb{E}\left\| \boldsymbol{\epsilon}_t \right\|$$

□

Note that this part in the bound is only reliant on the Anytime-GD mechanism. Now we just need to bound $\boldsymbol{\epsilon}_t$. We will define these momentum increments: $\hat{\mathbf{s}}_t := \alpha_{t-\tau_t} \mathbf{g}_{t-\tau_t} - \alpha_{t-\tau_t-1} \tilde{\mathbf{g}}_{t-\tau_t-1}$ is the momentum increment given at iteration t, $\mathbf{s}_t := \alpha_t \mathbf{g}_t - \alpha_{t-1} \tilde{\mathbf{g}}_{t-1}$ is the increment calculated in iteration t. For the first input of each worker, they all use $\mathbf{x}_1$, so we refer to it as $\mathbf{s}_1^{(i)}$. We define $\bar{\mathbf{s}}_t := \alpha_t \nabla f(\mathbf{x}_t) - \alpha_{t-1} \nabla f(\mathbf{x}_{t-1})$ as the expected momentum at iteration t. We also define $\alpha_0 = 0$. $\mathcal{T}(t)$ is the group of missing iterations at time t, as in the non-convex analysis. We define $I(t)$, the set of all workers who had at least 1 iteration at time t, and $i(t)$ to be the worker who is acting at iteration t. We can see that $I(t) = \{i(k) | k \in \mathcal{T}(t)\} \cup \{i(t)\}$. The weighted momentum $\mathbf{q}_t$ is:

$$\mathbf{q}_t := \sum_{k=1}^{t} \hat{\mathbf{s}}_k = \sum_{i \in I(t)} \mathbf{s}_1^{(i)} + \sum_{k \in [t] \setminus \{\mathcal{T}(t) \cup \{1\}\}} \mathbf{s}_k$$

Note that $\sum_{k=1}^{t} \bar{\mathbf{s}}_k = \alpha_t \nabla f(\mathbf{x}_t)$. Thus, the weighted momentum error $\boldsymbol{\epsilon}_t$ is:

$$\boldsymbol{\epsilon}_t := \mathbf{q}_t - \alpha_t \nabla f(\mathbf{x}_t) = \sum_{i \in I(t)} \left(\mathbf{s}_1^{(i)} - \bar{\mathbf{s}}_1 \right) + \sum_{k \in [t] \setminus \{\mathcal{T}(t) \cup 1\}} (\mathbf{s}_k - \bar{\mathbf{s}}_k) - \sum_{k \in \mathcal{T}(t)} (\bar{\mathbf{s}}_k - \bar{\mathbf{s}}_1)$$

Lemma C.4. *Let $\mathbf{s}_t = \alpha_t \mathbf{g}_t - \alpha_{t-1} \tilde{\mathbf{g}}_{t-1}$ and $\bar{\mathbf{s}}_t = \alpha_t \nabla f(\mathbf{x}_t) - \alpha_{t-1} \nabla f(\mathbf{x}_{t-1})$, with assumption 2.2 and the relevant assumption between 2.3 or 2.4, and $\alpha_t = t$, then:*

$$\|\mathbf{s}_t\| \leq G + 2DL \quad \& \quad \|\bar{\mathbf{s}}_t - \bar{\mathbf{s}}_1\| \leq 3DL$$
$$\mathbb{E} \|\mathbf{s}_t - \bar{\mathbf{s}}_t\|^2 \leq (\sigma + 2D\sigma_L)^2 \quad \& \quad \mathbb{E} \left\| \mathbf{s}_1^{(i)} - \bar{\mathbf{s}}_1 \right\|^2 \leq \sigma^2$$

Proof of Lemma C.4. We will rewrite the momentum increments as $\mathbf{s}_t = (\alpha_t - \alpha_{t-1})\mathbf{g}_t + \alpha_{t-1}(\mathbf{g}_t - \tilde{\mathbf{g}}_{t-1})$ and $\bar{\mathbf{s}}_t = (\alpha_t - \alpha_{t-1})\nabla f(\mathbf{x}_t) + \alpha_{t-1}(\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1}))$. Note that:

$$\|\mathbf{x}_t - \mathbf{x}_{t-1}\| = \frac{\alpha_t}{\alpha_{1:t}} \|\mathbf{w}_t - \mathbf{x}_{t-1}\| \leq \frac{\alpha_t}{\alpha_{1:t}} D$$

Since $\alpha_t = t$, we get that $\alpha_t - \alpha_{t-1} = 1$ and also:

$$\alpha_{t-1} \|\mathbf{x}_t - \mathbf{x}_{t-1}\| \leq \frac{\alpha_{t-1}\alpha_t}{\alpha_{1:t}} D = \frac{2(t-1)t}{t(t+1)} D = 2\frac{t-1}{t+1} D \leq 2D$$

For the first part (only relevant with assumption 2.4):

$$\|\mathbf{s}_t\| \leq (\alpha_t - \alpha_{t-1}) \|\mathbf{g}_t\| + \alpha_{t-1} \|\mathbf{g}_t - \tilde{\mathbf{g}}_{t-1}\| \leq G + \alpha_{t-1} \|\mathbf{x}_t - \mathbf{x}_{t-1}\| L \leq G + 2DL$$

Where the first inequality is the triangle inequality, the second inequality is the Lipchitz and smoothness assumptions, and the third inequality is the distance between adjacent queries.

For the second part:

$$\begin{aligned}
\|\bar{\mathbf{s}}_t - \bar{\mathbf{s}}_1\| &= \|(\alpha_t - \alpha_{t-1})\nabla f(\mathbf{x}_t) + \alpha_{t-1}(\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1})) - \alpha_1 \nabla f(\mathbf{x}_1)\| \\
&\leq \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_1)\| + \alpha_{t-1} \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1})\| \\
&\leq \|\mathbf{x}_t - \mathbf{x}_1\| L + \alpha_{t-1} \|\mathbf{x}_t - \mathbf{x}_{t-1}\| L \leq 3DL
\end{aligned}$$

Where the first inequality is the triangle inequality, the second inequality is the smoothness assumption, and the third inequality is the distance between adjacent queries.

For the third part:

$$\begin{aligned}
&\sqrt{\mathbb{E} \|\mathbf{s}_t - \bar{\mathbf{s}}_t\|^2} \\
=&\sqrt{\mathbb{E} \|(\alpha_t - \alpha_{t-1})(\mathbf{g}_t - \nabla f(\mathbf{x}_t)) + \alpha_{t-1}((\mathbf{g}_t - \nabla f(\mathbf{x}_t)) - (\tilde{\mathbf{g}}_{t-1} - \nabla f(\mathbf{x}_{t-1})))\|^2} \\
\leq&\sqrt{\mathbb{E} \|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|^2} + \alpha_{t-1} \sqrt{\mathbb{E} \|(\mathbf{g}_t - \nabla f(\mathbf{x}_t)) - (\tilde{\mathbf{g}}_{t-1} - \nabla f(\mathbf{x}_{t-1}))\|^2} \\
\leq&\sigma + \alpha_{t-1} \sqrt{\mathbb{E} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2} \sigma_L \leq \sigma + 2D\sigma_L
\end{aligned}$$

Where the first inequality is Minkowski's inequality $\sqrt{\mathbb{E} \|\mathbf{a} + \mathbf{b}\|^2} \leq \sqrt{\mathbb{E} \|\mathbf{a}\|^2} + \sqrt{\mathbb{E} \|\mathbf{b}\|^2}$, the second inequality is the bounded variance and smoothness variance, and the third inequality is the distance between adjacent queries.

For the fourth part:

$$\mathbb{E} \left\| \mathbf{s}_1^{(i)} - \bar{\mathbf{s}}_1 \right\|^2 = \mathbb{E} \left\| \nabla f\left(\mathbf{x}_1; \mathbf{z}_1^{(i)}\right) - \nabla f(\mathbf{x}_1) \right\|^2 \leq \sigma^2$$

Where we used the bounded variance assumption. □

Data Independent Case. We will bound the error in the data independent case.

Lemma C.5. *Let $f : \mathcal{K} \to \mathbb{R}$ be a function that satisfies assumptions 2.2 and 2.3. Then, for any iteration $t \geq 1$ with weights $\alpha_t = t$, the weighted error of the delayed momentum in Algorithm 2, $\boldsymbol{\epsilon}_t := \alpha_t(\mathbf{d}_t - \nabla f(\mathbf{x}_t))$, satisfies:*

$$\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\| \leq (\sigma + 2D\sigma_L)\sqrt{t} + 3DL(M-1)$$

Proof of Lemma C.5. Since the data is independent of the delay, we can easily separate the bias and variance of $\boldsymbol{\epsilon}_t$, since each $\mathbf{s}_k - \bar{\mathbf{s}}_k$ is orthogonal:

$$\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 = \sum_{i \in I(t)} \mathbb{E}\left\|\mathbf{s}_1^{(i)} - \bar{\mathbf{s}}_1\right\|^2 + \sum_{k \in [t] \setminus \{\mathcal{T}(k) \cup 1\}} \mathbb{E}\left\|\mathbf{s}_k - \bar{\mathbf{s}}_k\right\|^2 + \mathbb{E}\left\|\sum_{k \in \mathcal{T}(t)} (\bar{\mathbf{s}}_k - \bar{\mathbf{s}}_1)\right\|^2$$

The terms in the first and second sums can be bounded using Lemma C.4 to σ^2, and $(\sigma + 2D\sigma_L)^2$, since there are t terms in both sums, they are bounded together by $(\sigma + 2D\sigma_L)^2 t$. For the second term we can use the triangle inequality:

$$\left\|\sum_{k \in \mathcal{T}(t)} (\bar{\mathbf{s}}_k - \bar{\mathbf{s}}_1)\right\| \leq \sum_{k \in \mathcal{T}(t)} \left\|\bar{\mathbf{s}}_k - \bar{\mathbf{s}}_1\right\| \leq 3DL(M-1)$$

Where at the end, we used Lemma C.4 and the fact that there are at most $M-1$ terms. In total, we have:

$$\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2 \leq (\sigma + 2D\sigma_L)^2 t + 9D^2L^2(M-1)^2$$

Since $\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\| \leq \sqrt{\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\|^2}$ and $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we get:

$$\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\| \leq (\sigma + 2D\sigma_L)\sqrt{t} + 3DL(M-1)$$

□

Proof of Theorem 4.1. Continuing from Lemma C.3, and using Lemma C.5, we get this bound:

$$\begin{aligned}\Delta_T \leq & \frac{2D^2}{\eta T^2} + \frac{4DG^\star}{T} + \frac{4D}{T^2}\sum_{t=1}^{T}(\sigma + 2D\sigma_L)\sqrt{t} + \frac{12D^2L(M-1)}{T} \\ \leq & \frac{2D^2}{\eta T^2} + \frac{4DG^\star}{T} + \frac{4D\left(\sigma + 2D\sigma_L\right)}{\sqrt{T}} + \frac{12D^2L(M-1)}{T}\end{aligned}$$

Note that our only constraint on the learning rate was $\eta \leq \frac{1}{4LT}$, and as long as we use a learning that satisfy this condition and $\frac{1}{T\eta} \leq \mathcal{O}\left(\frac{G^\star}{D} + \left(\frac{\sigma}{D} + \sigma_L\right)\sqrt{T} + LM\right)$, we achieve an excess loss bound of $\mathcal{O}\left(\frac{DG^\star}{T} + \frac{D(\sigma + D\sigma_L)}{\sqrt{T}} + \frac{D^2LM}{T}\right)$. □

Data Dependent Case. We will bound the error in the data dependent case.

Lemma C.6. *Let $\mathbf{d}_t$ be as in Algorithm 2, and $\boldsymbol{\epsilon}_t := \alpha_t(\mathbf{d}_t - \nabla f(\mathbf{x}_t))$, with assumptions 2.2 and 2.4 and $\alpha_t = t$, with M workers, then:*

$$\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\| \leq (\sigma + 2D\sigma_L)\sqrt{t} + 2(G + DL)(M-1)$$

Proof of Lemma C.6. The ideal momentum $\hat{\mathbf{q}}_t$ is:

$$\hat{\mathbf{q}}_t := \mathbf{s}_1^{(1)} + \sum_{k=2}^{t} \mathbf{s}_k$$

Where we use the first iterate of worker 1. The ideal momentum error is $\hat{\boldsymbol{\epsilon}}_t := \hat{\mathbf{q}}_t - \alpha_t \nabla f(\mathbf{x}_t)$. $\mathbf{b}_t$ is defined as the difference between the real and ideal momentum:

$$\mathbf{b}_t := \mathbf{q}_t - \hat{\mathbf{q}}_t = \sum_{i \in I(t)} \mathbf{s}_1^{(i)} - \mathbf{s}_1^{(1)} - \sum_{k \in \mathcal{T}(t)} \mathbf{s}_k$$

Note that we write it like this, because there is a possibility that $1 \notin I(t)$. Since $\boldsymbol{\epsilon}_t - \hat{\boldsymbol{\epsilon}}_t = \mathbf{q}_t - \hat{\mathbf{q}}_t = \mathbf{b}_t$, we can bound $\|\boldsymbol{\epsilon}_t\| \leq \|\hat{\boldsymbol{\epsilon}}_t\| + \|\mathbf{b}_t\|$. The ideal momentum error $\hat{\boldsymbol{\epsilon}}_t := \hat{\mathbf{q}}_t - \alpha_t \nabla f(\mathbf{x}_t)$ is:

$$\hat{\boldsymbol{\epsilon}}_t := \hat{\mathbf{q}}_t - \alpha_t \nabla f(\mathbf{x}_t) = \left(\mathbf{s}_1^{(1)} - \bar{\mathbf{s}}_1\right) + \sum_{k=2}^{t} (\mathbf{s}_k - \bar{\mathbf{s}}_k)$$

Note that the ideal momentum $\hat{\boldsymbol{\epsilon}}_t$ is independent of the delay, thus even with data dependent delays, it will not affect $\hat{\boldsymbol{\epsilon}}_t$, which we just wrote as a Martingale difference sequence. Thus:

$$\mathbb{E}\left\|\hat{\boldsymbol{\epsilon}}_t\right\|^2 = \mathbb{E}\left\|\mathbf{s}_1^{(1)} - \bar{\mathbf{s}}_1\right\|^2 + \sum_{k=2}^{t} \mathbb{E}\left\|\mathbf{s}_k - \bar{\mathbf{s}}_k\right\|^2 \leq \sigma^2 + (\sigma + 2D\sigma_L)^2(t-1) \leq (\sigma + 2D\sigma_L)^2 t$$

Where we used Lemma C.4. With Jensen's inequality, we can bound:

$$\mathbb{E}\left\|\hat{\boldsymbol{\epsilon}}_t\right\| \leq \sqrt{\mathbb{E}\left\|\hat{\boldsymbol{\epsilon}}_t\right\|^2} \leq (\sigma + 2D\sigma_L)\sqrt{t}$$

For the delay, we can use the triangle inequality. For the case where $1 \in I(t)$:

$$\begin{aligned}\|\mathbf{b}_t\| &\leq \sum_{k \in \mathcal{T}(t)} \|\mathbf{s}_k\| + \sum_{i \in I(t) \setminus \{1\}} \left\|\mathbf{s}_1^{(i)}\right\| \\ &\leq \sum_{k \in \mathcal{T}(t)} (G + 2DL) + \sum_{i \in I(t) \setminus \{1\}} G \leq 2(G + DL)(M-1)\end{aligned}$$

Where we used Lemma C.4, the Lipchitz assumption, and the fact that $|\mathcal{T}(t)| = |I(t)| - 1 \leq M - 1$. If $1 \notin I(t)$, then we get:

$$\begin{aligned}\|\mathbf{b}_t\| &\leq \sum_{k \in \mathcal{T}(t)} (G + 2DL) + \sum_{i \in I(t) \cup \{1\}} G \\ &\leq (G + 2DL)(M-2) + GM \leq 2(G + DL)(M-1)\end{aligned}$$

Where we used $|\mathcal{T}(t)| = |I(t)| - 1 \leq M - 2$, since $1 \notin I(t)$. Putting the two bounds together, we got that:

$$\mathbb{E}\left\|\boldsymbol{\epsilon}_t\right\| \leq (\sigma + 2D\sigma_L)\sqrt{t} + 2(G + DL)(M-1)$$

□

Proof of Theorem 4.2. Continuing from Lemma C.3, and using Lemma C.6, we get this bound:

$$\begin{aligned}\Delta_T &\leq \frac{2D^2}{\eta T^2} + \frac{4DG^\star}{T} + \frac{4D}{T^2}\sum_{t=1}^{T}(\sigma + 2D\sigma_L)\sqrt{t} + \frac{8D}{T^2}\sum_{t=1}^{T}(G + DL)(M-1) \\ &\leq \frac{2D^2}{\eta T^2} + \frac{4DG^\star}{T} + \frac{4D\,(\sigma + 2D\sigma_L)}{\sqrt{T}} + \frac{8D(G + DL)(M-1)}{T}\end{aligned}$$

Note that our only constraint on the learning rate was $\eta \leq \frac{1}{4LT}$, and as long as we use a learning that satisfy this condition and $\frac{1}{T\eta} \leq \mathcal{O}\left(\left(\frac{\sigma}{D} + \sigma_L\right)\sqrt{T} + \left(\frac{G}{D} + L\right)M\right)$, we achieve an excess loss bound of $\mathcal{O}\left(\frac{D(\sigma + D\sigma_L)}{\sqrt{T}} + \frac{D(G+DL)M}{T}\right)$, since $G^\star \leq G$. □

D EXPERIMENTS

D.1 EXPERIMENTAL SETUP

We evaluate the proposed momentum framework on the Fashion-MNIST (Xiao et al., 2017) dataset in asynchronous training environments with $M \in \{5, 7\}$ workers and $q_1 \in \{0.3, 0.5, 1.0\}$.

Fashion-MNIST Dataset. Fashion-MNIST consists of 70,000 grayscale images of clothing items from 10 classes, each with a resolution of 28×28 pixels. The dataset is split into 60,000 training images and 10,000 test images.

D.2 MODEL ARCHITECTURE

We use a convolutional architecture with two 5×5 convolutional layers with 20 and 50 output channels, each followed by a ReLU activation and 2×2 max pooling, and two fully connected layers of sizes $800 \to 50$ and $50 \to 10$.

D.3 BASELINES

We compare the proposed Ordered Momentum (Algorithm 1) and Ordered μ^2-SGD (Algorithm 2) with constant parameters (Section 5) against the following baselines:

- Momentum (Equation (3), Polyak (1964))
- μ^2-SGD (Dahan & Levy (2025), Equation (12)).
- Vanilla SGD (Equation (1), Koloskova et al. (2022))
- Delay-Adaptive SGD (Equation (9), Mishchenko et al. (2022)).

For all methods, the stepsize is selected from $\eta \in \{0.001, 0.005, 0.01, 0.05, 0.07, 0.1, 0.2, 0.3, 0.5\}$. We use $\beta = 0.1$, corresponding to the conventional momentum coefficient $1 - \beta = 0.9$, and set the query-momentum parameter to $\gamma = 0.1$. For every configuration, we train the model for 20 epochs using a mini-batch size of 1024 and report the final test accuracy averaged over three random seeds.

D.4 DATA-DEPENDENT DELAY MODEL

We design a controlled simulation of data-dependent delays to reflect heterogeneous and input-dependent computation times commonly observed in asynchronous training.

Imbalanced worker arrivals. We simulate asynchrony by sampling the identity of the next arriving worker at each server iteration. Specifically, for M workers indexed by $i \in \{1, \dots, M\}$, worker i is sampled with probability proportional to its index, i.e.,

$$p_i = \frac{i}{\sum_{j=1}^{M} j} = \frac{2i}{M(M+1)}$$

The number of server iterations between consecutive arrivals of worker i therefore follows a geometric distribution with parameter p_i and determines that worker's delay. Workers with larger indices are sampled more frequently and thus have shorter expected delays. This setup induces heterogeneous delays across workers and captures variability in system-level scheduling.

Data-dependent delay times. To model variability in computation time induced by the data itself, we introduce delays that depend explicitly on sample characteristics.

Samples assigned to the *long-delay* group are only made available after a worker-specific delay threshold τ_i, while preserving the overall sampling proportions across groups. Let T_i denote the waiting time until worker i arrives, which follows a geometric distribution with parameter p_i, so that

$$\Pr\{T_i > \tau_i\} = (1 - p_i)^{\tau_i}$$

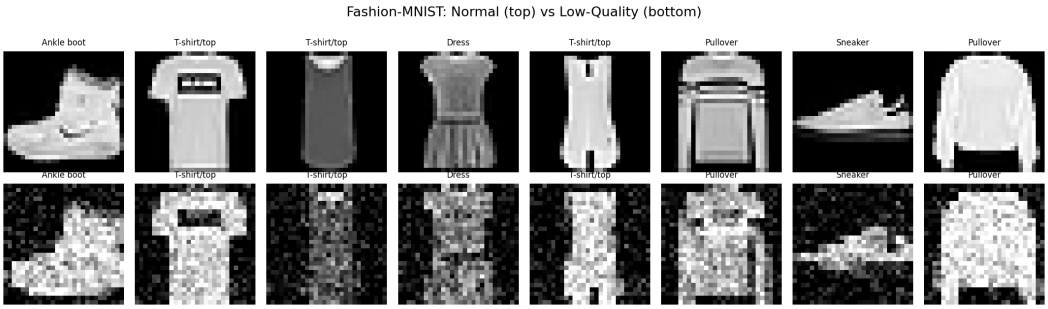


Figure 2: Representative high- and low-quality Fashion-MNIST samples. The top row shows the original images from $\mathcal{D}_{\text{high}}$, and the bottom row shows the corresponding degraded images from $\mathcal{D}_{\text{low}}$. The degradation consists of downsampling, nearest-neighbor upsampling, and additive Gaussian noise, while the labels remain unchanged.

To sample from a prescribed mixture distribution $\mathcal{D} = q_1\mathcal{D}_1 + q_2\mathcal{D}_2$, with $q_1 + q_2 = 1$, we select a sample from $\mathcal{D}_1$ if $T_i > \tau_i$ and from $\mathcal{D}_2$ otherwise. Enforcing $\Pr\{T_i > \tau_i\} = q_1$ yields

$$\tau_i = \frac{\log(q_1)}{\log(1 - p_i)}$$

As a result, if worker i arrives after τ_i, we sample $z \sim \mathcal{D}_1$; otherwise, we sample $z \sim \mathcal{D}_2$. This procedure ensures that long-delay samples are introduced according to the desired mixture proportions while respecting worker-specific arrival patterns.

D.5 Data-Dependent Delays in the Experiments

We instantiate the data-dependent delay model from Section D.4 on Fashion-MNIST by associating longer delays with higher-quality images. We sweep $q_1 \in \{0.3, 0.5, 1.0\}$ in the mixture

$$\mathcal{D} = q_1\mathcal{D}_{\text{high}} + (1 - q_1)\mathcal{D}_{\text{low}},$$

where $\mathcal{D}_{\text{high}}$ is the original training distribution and $\mathcal{D}_{\text{low}}$ is a degraded version of the same distribution. The degradation modifies only the input image and preserves its label. For $q_1 = 1.0$, every sample is drawn from $\mathcal{D}_{\text{high}}$, so image quality is independent of the delay; this is our data-independent setting.

Image-quality distributions. Samples from $\mathcal{D}_{\text{high}}$ undergo only the standard dataset-specific normalization. To construct $\mathcal{D}_{\text{low}}$, we first downsample each image by a factor of two using strided subsampling and then restore it to its original resolution through nearest-neighbor replication. We subsequently add independent Gaussian noise with standard deviation 0.1. Fashion-MNIST images are reduced from 28×28 to 14×14 before being restored to their original resolution. Representative high- and low-quality samples are shown in Figure 2.

Coupling image quality and delay. We identify the long-delay distribution $\mathcal{D}_1$ in Section D.4 with the high-quality distribution and the short-delay distribution $\mathcal{D}_2$ with the low-quality distribution:

$$\mathcal{D}_1 = \mathcal{D}_{\text{high}}, \qquad \mathcal{D}_2 = \mathcal{D}_{\text{low}}.$$

Accordingly, when worker i has waiting time T_i, its sample is drawn according to

$$\mathbf{z} \sim \begin{cases} \mathcal{D}_{\text{high}}, & T_i > \tau_i, \\ \mathcal{D}_{\text{low}}, & T_i \le \tau_i, \end{cases} \qquad \tau_i = \frac{\log(q_1)}{\log(1 - p_i)}.$$

Since $\Pr\{T_i > \tau_i\} = q_1$, this construction preserves the prescribed marginal fraction of high-quality samples while making image quality statistically dependent on the delay: original images are associated with longer waiting times, whereas degraded images are associated with shorter waiting times.

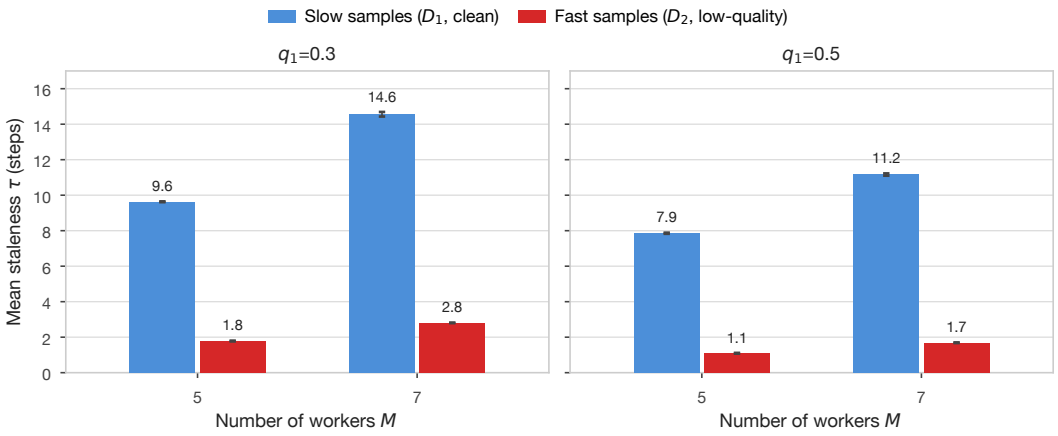


Figure 3: **Empirical coupling between image quality and delay on Fashion-MNIST.** Mean staleness is reported for clean samples from $\mathcal{D}_1 = \mathcal{D}_{\text{high}}$ and degraded samples from $\mathcal{D}_2 = \mathcal{D}_{\text{low}}$, for $M \in \{5, 7\}$ workers and $q_1 \in \{0.3, 0.5\}$. Clean samples have larger mean staleness than degraded samples, confirming that the construction in Section D.4 induces the intended data-dependent delays.

D.6 Complete Results

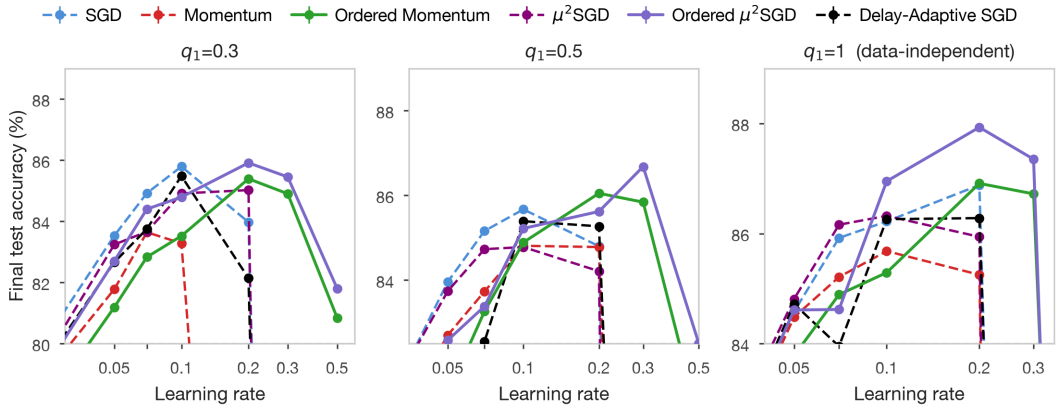


Figure 4: **Learning-rate robustness on Fashion-MNIST with $M = 5$ workers (means only).** We report the same mean final test accuracies as in Figure 5, omitting error bars for clarity. This view highlights the relative performance of the methods and their stable learning-rate ranges.

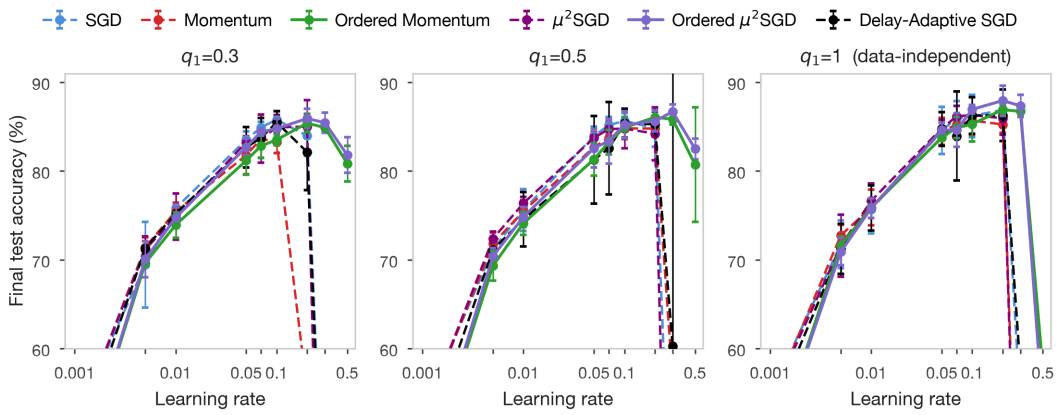


Figure 5: **Learning-rate robustness on Fashion-MNIST with $M = 5$ workers and the error bars indicate one standard deviation.**

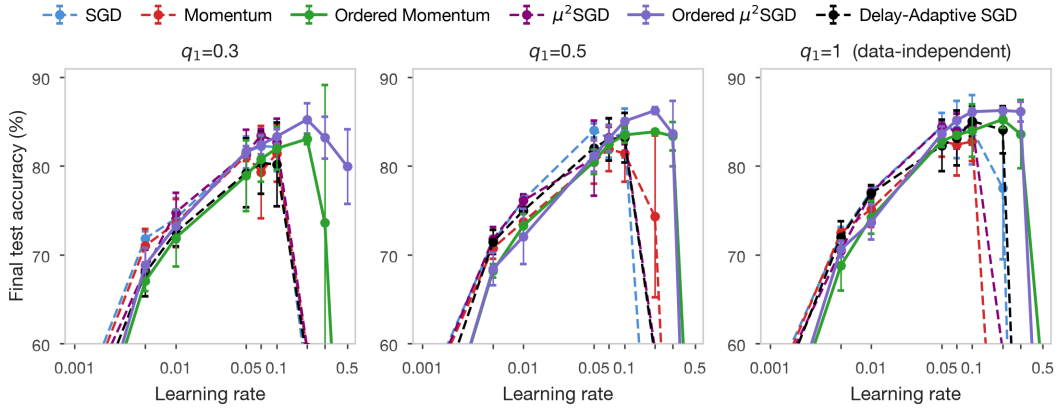


Figure 6: **Learning-rate robustness on Fashion-MNIST with $M = 7$ workers and the error bars indicate one standard deviation.**