# Weight for Robustness: A Comprehensive Approach towards Optimal Fault-Tolerant Asynchronous ML

**Tehila Dahan**
Department of Electrical Engineering
Technion
Haifa, Israel
t.dahan@campus.technion.ac.il

**Kfir Y. Levy**
Department of Electrical Engineering
Technion
Haifa, Israel
kfirylevy@technion.ac.il

## Abstract

We address the challenges of Byzantine-robust training in asynchronous distributed machine learning systems, aiming to enhance efficiency amid massive parallelization and heterogeneous computing resources. Asynchronous systems, marked by independently operating workers and intermittent updates, uniquely struggle with maintaining integrity against Byzantine failures, which encompass malicious or erroneous actions that disrupt learning. The inherent delays in such settings not only introduce additional bias to the system but also obscure the disruptions caused by Byzantine faults. To tackle these issues, we adapt the Byzantine framework to asynchronous dynamics by introducing a novel weighted robust aggregation framework. This allows for the extension of robust aggregators and a recent meta-aggregator to their weighted versions, mitigating the effects of delayed updates. By further incorporating a recent variance-reduction technique, we achieve an optimal convergence rate for the first time in an asynchronous Byzantine environment. Our methodology is rigorously validated through empirical and theoretical analysis, demonstrating its effectiveness in enhancing fault tolerance and optimizing performance in asynchronous ML systems.

## 1 Introduction

In recent years, there has been significant growth in the development of large-scale machine learning (ML) models and the volume of data they require [Zhao et al., 2023]. To efficiently accelerate large-scale training processes, Distributed ML has emerged as a crucial approach that can be categorized into synchronous and asynchronous paradigms. In synchronous learning, workers update the model simultaneously using the average of their outputs, similar to the Minibatch approach [Dekel et al., 2012]. Asynchronous learning, however, allows workers to operate independently, sending updates as they are ready without waiting for others [Arjevani et al., 2020]. This prevents slow workers from hindering the process, making it especially practical as the number of workers increases.

A major challenge of distributed ML is fault-tolerance, and Byzantine ML [Alistarh et al., 2018, Lamport et al., 2019, Guerraoui et al., 2023] is a powerful framework for tackling this aspect. Byzantine ML captures a broad spectrum of failures within distributed environments, including random malfunctions or even malicious workers aiming to disrupt the training process. This makes Byzantine ML widely applicable across various domains to ensure robust performance.

Addressing the Byzantine problem in synchronous distributed learning is well-established [Karimireddy et al., 2020, 2021, Allouah et al., 2023, Farhadkhani et al., 2022, Alistarh et al., 2018, Dahan and Levy, 2024]. Two primary ingredients were found to be crucial towards tackling Byzantine ML in synchronous settings: **(i)** *Robust Aggregators* [Yin et al., 2018, Blanchard et al., 2017, Chen et al., 2017]: such aggregators combine the gradient estimates sent by the workers to a single estimate

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

while filtering out the outliers which may hinder the training process. While the use of robust aggregators is crucial, it was found to be insufficient, and an additional ingredient of **(ii)** *learning from history* was shown to be vital in mitigating Byzantine faults [Karimireddy et al., 2021]. And, the performance of robust aggregators was systematically explored within a powerful generic framework [Karimireddy et al., 2020, 2021, Allouah et al., 2023, Farhadkhani et al., 2022, Dahan and Levy, 2024]. Moreover, due to the diversity of Byzantine scenarios [Xie et al., 2020a, Allen-Zhu et al., 2020, Baruch et al., 2019], it was found that relying on a single aggregator is insufficient, making the variety of robust aggregators essential. Unfortunately, many existing aggregators have sub-optimal performance. This drawback was elegantly resolved by the design of meta-aggregators [Karimireddy et al., 2020, Allouah et al., 2023, Dahan and Levy, 2024], that enable to boost the performance of baseline aggregators. Unfortunately, in the asynchronous case, the use of robust aggregators is not straightforward, as updates are typically applied individually per-worker, rather than averaging outputs from all workers at once [Arjevani et al., 2020].

Despite its advantages, asynchronous distributed learning presents unique challenges, particularly when dealing with Byzantine faults. The delays inherent in asynchronous settings introduce additional bias to the system and obscure the disruptions caused by Byzantine faults. In fact, in contrast to the synchronous Byzantine setting, all existing approaches towards the asynchronous Byzantine case do not ensure a generalization error (excess loss) that diminishes with the number of honest data-samples and updates. This applies to works for both convex [Fang et al., 2022] as well as non-convex scenarios [Xie et al., 2020b, Yang and Li, 2023]; as well as to works that further assume the availability of a *trusted dataset* possessed by the central-server [Xie et al., 2020b, Fang et al., 2022]. Furthermore, the performance guarantees of all existing approaches towards that setting include an explicit dependence on the dimension of the problem — a drawback that does not exist for SOTA synchronous Byzantine approaches.

**Contributions.** We explore the asynchronous Byzantine setting under the fundamental framework of Stochastic Convex Optimization (SCO) [Hazan et al., 2016]. Our work is the first to achieve a convergence rate that diminishes with the number of honest data samples and updates and does not explicitly depend on the problem's dimension. In the absence of Byzantine workers, our rate matches the optimal performance of Byzantine-free asynchronous settings. This stands in contrast to previous efforts on Byzantine, which did not attain diminishing rates or dimensionality independence, even without Byzantine workers. We also show the effectiveness of our approaches in practice. Our contributions:

- We quantify the difficulty in asynchronous scenarios by considering the *number of Byzantine updates*, which is more natural than the standard measure of *number of Byzantine workers*.
- We identify the need to utilize weighted aggregators rather than standard ones in favor of asynchronous Byzantine problems. Towards doing so, we extend the robust aggregation framework to allow and include weights and develop appropriate (weighted) rules and a meta-aggregator.
- **Achieving Optimal Convergence**: We incorporate our weighted robust framework with a recent double momentum mechanism, leveraging its unique features to achieve an optimal convergence rate for the first time in asynchronous Byzantine ML.

**Related Work.** A long line of studies has explored the synchronous Byzantine setting (see e.g., Alistarh et al. [2018], Karimireddy et al. [2020, 2021], Allouah et al. [2023], Farhadkhani et al. [2022], Allen-Zhu et al. [2020], El Mhamdi et al. [2021], Dahan and Levy [2024]). Alistarh et al. [2018], Karimireddy et al. [2021] demonstrated that historical information is crucial for optimal performance in Byzantine scenarios; and Karimireddy et al. [2021] introduced the idea of combining generic aggregation rules, together with standard momentum with a parameter of $1/\sqrt{T}$ to effectively incorporates $\sqrt{T}$ iterations of historical gradients. Additionally, Dahan and Levy [2024] showed that a double momentum approach is effective by taking a momentum parameter of $1/T$, capturing the entire gradient history.

Robust aggregators such as Coordinate-wise Trimmed Mean (CWTM) [Yin et al., 2018], Krum [Blanchard et al., 2017], Geometric Median (GM) [Chen et al., 2017], CWMed [Yin et al., 2018], and Minimum Diameter Averaging [Guerraoui et al., 2018] have also proven to be highly beneficial in synchronous settings and have been evaluated within robust frameworks [Allouah et al., 2023, Karimireddy et al., 2020, Farhadkhani et al., 2022, Dahan and Levy, 2024]. However, not all robust aggregators achieve optimal performance, leading to the development of meta-aggregators

[Karimireddy et al., 2020, Allouah et al., 2023, Dahan and Levy, 2024] to enhance their effectiveness. While standard aggregation works well in synchronous settings, where outputs are averaged across all workers, it is less suitable for asynchronous settings, where updates are processed individually as they arrive [Arjevani et al., 2020].

To adapt these approaches to asynchronous settings, Yang and Li [2023] devised BASGDm, an extension of BASGD [Yang and Li, 2021], that groups worker momentums into buckets that are then aggregated using a robust aggregator. Other methods, like Zeno++ [Xie et al., 2020b] and AFLGuard [Fang et al., 2022], rely on a trusted dataset on the central server, which hinders their practicality. Kardam [Damaskinos et al., 2018] uses the Lipschitzness of gradients to filter out outliers. Unfortunately, none of these approaches ensure a generalization error (excess loss) that diminishes with the number of honest data-samples and updates, and suffers from an explicit dependence on the problem's dimension. And this applies even in the absence of Byzantine faults.

Asynchronous Byzantine ML faces unique challenges as inherent delays add bias that obscures Byzantine disruptions. To mitigate this delay-bias in asynchronous, non-Byzantine scenarios, Cohen et al. [2021], Aviv et al. [2021] propose methods to keep model weights relatively close during iterations. Other approaches [Stich and Karimireddy, 2019, Arjevani et al., 2020, Mishchenko et al., 2022] suggest adjusting the step size proportionally to the delay. These strategies have proven useful in reducing the negative impact of delays, and achieve optimal performance.

Our work extends several concepts from Dahan and Levy [2024] to the asynchronous scenario. We devise a novel generalization of their Centered Trimmed Meta Aggregator (CTMA) towards weighted meta-aggregation, making it amenable to asynchronous scenarios. In the spirit of Dahan and Levy [2024], we also adopt a recent variance reduction technique called $\mu^2$-SGD [Levy, 2023]. Nevertheless, while Dahan and Levy [2024] used this technique in a straightforward manner, we found it crucial to appropriately incorporate individual per-worker weights to overcome the challenge of asynchronicity in Byzantine ML.

## 2 Setting

Our discussion focuses on the minimization of a smooth convex objective $f : \mathcal{K} \to \mathbb{R}$:

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

where $\mathcal{K} \subseteq \mathbb{R}^d$ is a compact convex set and $\mathcal{D}$ denotes an unknown distribution from which we can draw i.i.d samples $\{\mathbf{z}_t \sim \mathcal{D}\}_t$. Our work considers first-order methods that iteratively utilize gradient information to approach an optimal point. Such methods output a solution $\mathbf{x}_T$, which is evaluated by the expected excess loss:

$$\text{ExcessLoss} := \mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] \,,$$

where $\mathbf{x}^*$ is a solution that minimizes $f$ over $\mathcal{K}$ and $\mathbf{x}_T \in \mathcal{K}$ approximates this optimal solution.

**Asynchronous Training.** We explore these methods within a distributed environment involving multiple workers. Our discussion focuses on a *centralized* distributed framework characterized by a central Parameter Server ($\mathcal{PS}$) that may communicate with $m$ workers. Each of these workers may draw i.i.d. samples $\mathbf{z} \sim \mathcal{D}$; and based on these samples, compute unbiased gradient estimate $\mathbf{g} \in \mathbb{R}^d$ at a point $\mathbf{x} \in \mathcal{K}$. Concretely, a worker may compute $\mathbf{g} := \nabla f(\mathbf{x};\mathbf{z})$; implying that $\mathbb{E}[\mathbf{g}|\mathbf{x}] = \nabla f(\mathbf{x})$. Specifically, our main focus is on *Asynchronous* systems, where the $\mathcal{PS}$ does not wait to receive the stochastic gradient computations from all machines; instead, it updates its model whenever a worker completes a (stochastic) gradient computation. That worker then proceeds to compute a gradient estimate for the updated model, while the other workers continue to compute gradients based on 'stale' models. This staleness leads to the use of staled (and therefore biased) gradient estimates, which is a major challenge in designing and analyzing asynchronous training methods.

**Asynchronous Byzantine Framework.** We assume that an unknown subset of the $m$ workers are *Byzantine*, implying that these workers may transmit arbitrary or malicious information during the training process, and these "Byzantine" workers may even collaborate to disrupt the training. We assume that the fraction of updates that arrive from Byzantine workers during the asynchronous training process is bounded and strictly smaller than $1/2$ and denote this fraction by $\lambda$.

**Remark 2.1** (Fraction of Byzantine Updates vs. Byzantine Workers)**.** *In both synchronous and asynchronous settings, it is common to consider a bound on the* ***fraction of Byzantine workers*** *(up*

*to ½) [Allouah et al., 2023, Farhadkhani et al., 2022, Karimireddy et al., 2020, 2021, Yang and Li, 2023, 2021, Damaskinos et al., 2018]. In synchronous scenarios this is meaningful since the server equally treats the information from all workers; which is done by equally averaging gradients of all workers in each iteration in a mini-batch fashion [Dekel et al., 2012]. Conversely, in asynchronous scenarios, faster workers contribute to more updates than slower workers, leading to an unequal influence on the training process. In such scenarios, the fraction of Byzantine workers is less relevant; and it is therefore much more natural to consider the* ***fraction of Byzantine updates****. Interestingly, our definition aligns with the standard one (for the synchronous case), which considers the number of Byzantine workers.*

**Notation.** For each worker $i \in [m]$ and iteration $t$, $s_t^{(i)}$ represents the total number of updates by worker $i$ up to $t$, and $\tau_t^{(i)}$ is the delay compared to the current model. $t^{(i)}$ is the last update before $t$, making $\tau_t^{(i)}$ the time since the second last update (Figure 1). $\tau_t$ denotes the delay for the worker arriving at iteration $t$, i.e., if worker $j$ arrives at iteration $t$ then $\tau_t = \tau_t^{(j)}$.

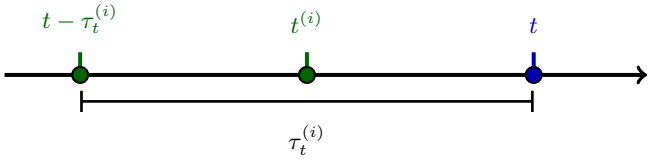


Figure 1: Illustration of the delay interval $\tau_t^{(i)}$ for worker $i$ at iteration $t$, marking $t$ (current iteration), $t^{(i)}$ (most recent update from worker $i$), and $t - \tau_t^{(i)}$ (previous update from worker $i$).

For a given time (iteration) $t$, let $t^{(i)}$ be the last iteration when worker $i$ made an update. We denote $\mathbf{d}_t^{(i)} := \mathbf{d}_{t^{(i)}}$, $\mathbf{g}_t^{(i)} := \mathbf{g}_{t^{(i)}}$, $\tilde{\mathbf{g}}_t^{(i)} := \tilde{\mathbf{g}}_{t^{(i)}}$, and $\mathbf{x}_t^{(i)} = \mathbf{x}_{t^{(i)}}$, where the latter are individual vectors that we will later define for any worker $i$. Throughout, $\|\cdot\|$ represents the $L_2$-norm. For any natural $N$, $[N] = \{1, \ldots, N\}$. We use the compressed sum notation $\alpha_{1:t} = \sum_{k=1}^{t} \alpha_k$. For every $\mathbf{x} \in \mathbb{R}^d$, the orthogonal projection of $\mathbf{x}$ onto a set $\mathcal{K}$ is denoted by $\Pi_{\mathcal{K}}(\mathbf{x}) = \arg\min_{\mathbf{y} \in \mathcal{K}} \|\mathbf{y} - \mathbf{x}\|^2$. We denote $\mathcal{B}$ and $\mathcal{G}$ as the subsets of Byzantine workers and honest workers, respectively, such that $|m| = |\mathcal{G}| + |\mathcal{B}|$.

**Assumptions.** We use the following conventional assumptions:

**Bounded Diameter**: we assume there exists $D > 0$ such that $\max_{\mathbf{x},\mathbf{y} \in \mathcal{K}} \|\mathbf{x} - \mathbf{y}\| \leq D$. (1)

**Bounded Variance**: there exists $\sigma > 0$ such that $\forall \mathbf{x} \in \mathcal{K}$, $\mathbf{z} \in$ **Support**$\{\mathcal{D}\}$,

$$\mathbb{E}\|\nabla f(\mathbf{x}; \mathbf{z}) - \nabla f(\mathbf{x})\|^2 \leq \sigma^2 \,. \tag{2}$$

**Expectation over Smooth Functions**: we assume that $f(\cdot)$ is an expectation of smooth functions, i.e. $\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}$ , $\mathbf{z} \in$ **Support**$\{\mathcal{D}\}$ there exist $L > 0$ such that,

$$\|\nabla f(\mathbf{x}; \mathbf{z}) - \nabla f(\mathbf{y}; \mathbf{z})\| \leq L\|\mathbf{x} - \mathbf{y}\| \,, \tag{3}$$

The above assumption also implies that the expected loss $f(\cdot)$ is $L$ smooth.

**Bounded Smoothness Variance** [Levy, 2023]: in Appendix A we show that Eq. (3) implies that, $\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}$, $\mathbf{z} \in$ **Support**$\{\mathcal{D}\}$ there exists $\sigma_L^2 \in [0, L^2]$ such,

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

**Bounded Delay**: $\exists K > 0$ such that for each worker $i \in [m]$, $\tau_{min}^{(i)} \leq \tau_t^{(i)} \leq K\tau_{min}^{(i)}$ (5)

where $\tau_{min}^{(i)}$ is the minimum delay of worker $i$. $K$ bounds the variance of the delay for each worker.

**Bounded Byzantine Iterations**: there exists $0 \leq \lambda < 1/2$ such that $t \in [T]$: $t_{\mathcal{B}} \leq \lambda t$ (6)

where $t_{\mathcal{B}}$ is the total number of iterations made by Byzantine workers up to iteration $t$.

**Sample-Arrival Independence**: we assume that the delays in the system (i.e. $\tau_t^{(i)}$'s) are independent of the data samples. This is a standard assumption in asynchronous training scenarios, see e.g., Arjevani et al. [2020], Aviv et al. [2021].

## 3 Weighted Robust Aggregation Rules

As we have mentioned, robust aggregation rules have played a major role in designing fault-tolerant ML training methods for synchronous settings (see, e.g., Allouah et al. [2023], Karimireddy et al. [2020, 2021], Dahan and Levy [2024]). These existing aggregation rules treat inputs from all workers equally, which makes sense in synchronous cases where all workers contribute the same number of updates and data samples. Conversely, this symmetry breaks down in asynchronous settings, where faster (honest) workers contribute more updates and samples compared to slower workers.

Inspired by this asymmetry, we have identified the need to define a notion of weighted robust aggregators that generalizes the standard definition of robust aggregators. In this section, we provide such a definition, derive weighted variants of standard aggregators that satisfy our new definition, and design a generic meta-approach to derive optimal weighted aggregation rules. Later, in Section 4, we demonstrate the benefits of using weighted robust aggregators as a crucial building block in designing asynchronous fault-tolerant training methods (see Alg. 2).

### 3.1 Robust Weighted Aggregation Framework

Below, we generalize the definition introduced by Dahan and Levy [2024], Karimireddy et al. [2020, 2021] to allow and associate weights to the inputs of the robust aggregation rule, therefore allowing the aggregator to unequally treat its inputs.

**Definition 3.1.** $(c_\lambda, \lambda)$-**weighted robust**. *Assume we have $m$ random vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^d$ and corresponding weights $s_1, \ldots, s_m > 0$. Also assume we have an "honest" subset $\mathcal{G} \subseteq [m]$, implying $\{\mathbf{x}_i\}_{\in \mathcal{G}}$ are independent of each other. Finally, assume that there exists $\lambda \in [0, 1/2)$ such that $\sum_{i\in\mathcal{G}} s_i \geq (1-\lambda) s_{1:m}$. Moreover, assume that for any $i \in \mathcal{G}$ there exist $\rho_i \geq 0$ such that,*

$$\mathbb{E}\|\mathbf{x}_i - \bar{\mathbf{x}}_\mathcal{G}\|^2 \leq \rho_i^2, \quad \forall i \in \mathcal{G} \; .$$

*Then an aggregation rule $\mathcal{A}_\omega$ is called $(c_\lambda, \lambda)$-weighted robust if for any such $m$ random vectors and weights and $\lambda \geq 0$, it outputs $\hat{\mathbf{x}} \leftarrow \mathcal{A}_\omega(\mathbf{x}_1, \ldots, \mathbf{x}_m; s_1, \ldots, s_m)$ such that,*

$$\mathbb{E}\|\hat{\mathbf{x}} - \bar{\mathbf{x}}_\mathcal{G}\| \leq c_\lambda \rho^2$$

*for some $c_\lambda \geq 0$. Above, $\bar{\mathbf{x}}_\mathcal{G} := \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{G}} s_i \mathbf{x}_i$, $\rho^2 := \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{G}} s_i \rho_i^2$, and the expectation is w.r.t. $\{\mathbf{x}_i\}_{i=1}^m$ and (possible) randomization in the $\mathcal{A}_\omega$.*

Here, $\lambda$ represents the fraction of the sum of the non-honest vectors' weights, unlike the unweighted definition (in synchronous cases) [Karimireddy et al., 2020, 2021, Allouah et al., 2023, Farhadkhani et al., 2022] where it indicates the fraction of non-honest vectors. These definitions align when all weights are equal [Dahan and Levy, 2024]. Similarly to the unweighted version, the optimal $c_\lambda$ should be $c_\lambda \leq O(\lambda)$ [Dahan and Levy, 2024].

**Remark 3.1.** *Note that our definition is generic and may be applied in both convex and non-convex scenarios. Moreover, it is natural to consider such weighted aggregators beyond asynchronous settings. For example, in synchronous settings where workers have varying batch sizes, weighted aggregation based on batch sizes may be more effective than uniform aggregation.*

Next, we present two weighted variants of standard (non-weighted) aggregators that satisfy the above definition (we defer the proof into Appendix C). Table 1 summarizes their $c_\lambda$ values.

### 3.2 Weighted Variant of Geometric Median and Coordinate-Wise

**Weighted Geometric Median (WeightedGM)** The Weighted Geometric Median (WeightedGM) minimizes the weighted sum of Euclidean distances to a set of points. Formally, for points $\{\mathbf{x}_i\}_{i=1}^m$ and corresponding weights $\{s_i\}_{i=1}^m$, WeightedGM $\in \arg\min_{\mathbf{y}\in\mathbb{R}^d} \sum_{i\in[m]} s_i \|\mathbf{y} - \mathbf{x}_i\|$ .

**Weighted Coordinate-Wise Median (WeightedCWMed)** The Weighted Coordinate-Wise Median (WeightedCWMed) aggregates multi-dimensional data by finding the weighted median of each coordinate separately. Thus, for given coordinate if $\{\mathbf{x}_i\}_{i=1}^m$ are sorted and weights $\{s_i\}_{i=1}^m$, the weighted median $\mathbf{x}_{j^*}$ is the element where: $j^* = \arg\min_{j\in[m]} \left\{ \sum_{i\in[j]} s_i > \frac{1}{2} \sum_{i\in[m]} s_i \right\}$ . If $\sum_{i=1}^j s_i = \frac{1}{2} \sum_{i=1}^m s_i$ for some $j$, then: WeightedMedian $= \frac{\mathbf{x}_j + \mathbf{x}_{j+1}}{2}$.

| Aggregation | $\omega$-GM | $\omega$-CWMed | $\omega$-GM + $\omega$-CTMA | $\omega$-CWMed + $\omega$-CTMA |
|---|---|---|---|---|
| $c_\lambda$ | $\left(1+\frac{\lambda}{1-2\lambda}\right)^2$ | $\left(1+\frac{\lambda}{1-2\lambda}\right)^2$ | $\lambda\left(1+\frac{\lambda}{1-2\lambda}\right)^2$ | $\lambda\left(1+\frac{\lambda}{1-2\lambda}\right)^2$ |

Table 1: Summary of weighted aggregation rules and their respective $c_\lambda$ values.

**Algorithm 1** Weighted Centered Trimmed Meta Aggregator ($\omega$-CTMA)

**Input:** Set of vectors $\{\mathbf{x}_i\}_{i=1}^m$, weights $\{s_i\}_{i=1}^m$, threshold parameter $\lambda \in [0, 1/2)$,
$(c_\lambda, \lambda)$-weighted robust aggregated vector $\mathbf{x}_0 \leftarrow \mathcal{A}_\omega(\{\mathbf{x}_i\}_{i=1}^m; \{s_i\}_{i=1}^m)$.
Sort the sequence $\{\|\mathbf{x}_i - \mathbf{x}_0\|\}_{i=1}^m$ in non-decreasing order, and then reindex $\{\mathbf{x}_i\}_{i\in[m]}$ and their corresponding weights $\{s_i\}_{i\in[m]}$ according to this new order.
Define $S \leftarrow$ set of indices corresponding to the first $j^*$ elements in the sorted sequence, where $j^*$ is the smallest $j \in [m]$ for which $\sum_{i\in[j]} s_i \geq (1-\lambda)\sum_{i\in[m]} s_i$.
Set $s_{m+1} \leftarrow (1-\lambda)\sum_{i\in[m]} s_i - \sum_{i\in[j^*-1]} s_i, \quad \mathbf{x}_{m+1} \leftarrow \mathbf{x}_{j^*}, \quad S \leftarrow (S\backslash\{j^*\}) \cup \{m+1\}$.
Compute the weighted sum: $\hat{\mathbf{x}} \leftarrow (1/\sum_{i\in S} s_i)\sum_{i\in S} s_i \mathbf{x}_i$.
**Output:** $\hat{\mathbf{x}}$

### 3.3 Weighted Centered Trimmed Meta Aggregator ($\omega$-CTMA)

Table 1 illustrates that $\omega$-GM and $\omega$-CWMed fail to achieve the desired optimal $c_\lambda = O(\lambda)$; typically for $\lambda \leq 1/3$, their $c_\lambda$ remains $\leq O(1)$. To address this suboptimality, we propose $\omega$-CTMA, a weighted extension of the Centered Trimmed Meta Aggregator (CTMA) [Dahan and Levy, 2024]. This extension enables us to achieve the optimal bound $c_\lambda \leq O(\lambda)$ for $\lambda \leq 1/3$ (see Table 1).

The $\omega$-CTMA algorithm (Algorithm 1) operates on a set of vectors along with their associated weights, a threshold $\lambda \in [0, 1/2)$, and a $(c_\lambda, \lambda)$-weighted robust aggregator. It sorts the distances between each vector and the weighted robust aggregator, trims the set based on the threshold to satisfy $\sum_{i\in S} s_i = (1-\lambda)s_{1:m}$, and calculates a weighted average of the vectors, excluding outliers based on their proximity to an anchor point—the weighted robust aggregator.

**Lemma 3.1.** *Under the assumptions outlined in Definition 3.1, if $\omega$-CTMA receives a $(c_\lambda, \lambda)$-weighted robust aggregator, $\mathcal{A}_\omega$; then the output of $\omega$-CTMA, $\hat{\mathbf{x}}$, is $(60\lambda(1+c_\lambda), \lambda)$-robust.*

For the complete analysis, please refer to Appendix C.2. Like CTMA [Dahan and Levy, 2024], $\omega$-CTMA is highly efficient, with a computational complexity of $O(dm + m\log m)$, similar to $\omega$-GM, $\omega$-CWMed, and weighted averaging, differing by at most an additional logarithmic factor.

## 4 Asynchronous Robust Training

We leverage the $\mu^2$-SGD algorithm [Levy, 2023], a double momentum mechanism that enhances variance reduction. By seamlessly incorporating our weighted robust framework as a black box into the $\mu^2$-SGD, we derive an optimal asynchronous Byzantine convergence rate.

**$\mu^2$-SGD:** The $\mu^2$-SGD is a variant of standard SGD, incorporating several key modifications in its update rule:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{K}}\left(\mathbf{w}_t - \eta\alpha_t \mathbf{d}_t\right), \quad \mathbf{x}_{t+1} = \frac{1}{\alpha_{1:t+1}}\sum_{k\in[t+1]} \alpha_k \mathbf{w}_k; \quad \mathbf{w}_1 = \mathbf{x}_1 \in \mathcal{K},\ \forall t > 1.$$

Here, $\{\alpha_t > 0\}_t$ are importance weights that emphasize different update steps, with $\alpha_t \propto t$ to place more weight on recent updates. The sequence $\{\mathbf{x}_t\}_t$ represents weighted averages of the iterates $\{\mathbf{w}_t\}_t$, and $\mathbf{d}_t$ is an estimate of the gradient at the average point, $\nabla f(\mathbf{x}_t)$, differing from standard SGD, which estimates gradients at the iterates, $\nabla f(\mathbf{w}_t)$.

This approach relates to Anytime-GD [Cutkosky, 2019], which is strongly connected to momentum and acceleration concepts [Cutkosky, 2019, Kavis et al., 2019]. While the stochastic version of Anytime-GD typically uses the estimate $\nabla f(\mathbf{x}_t; \mathbf{z}_t)$, $\mu^2$-SGD employs a variance reduction mechanism to produce a *corrected momentum* estimate $\mathbf{d}_t$ [Cutkosky and Orabona, 2019]. Specifically, $\mathbf{d}_1 = \nabla f(\mathbf{x}_1; \mathbf{z}_1)$, and for $t > 2$:

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

**Algorithm 2** Asynchronous Robust $\mu^2$-SGD

**Input:** learning rate $\eta_t > 0$, starting point $\mathbf{x}_1 \in \mathcal{K}$, number of steps $T$, importance weights $\{\alpha_t\}_t$, momentum correction weights $\{\beta_t\}_t$, $(c_\lambda, \lambda)$-robust weighted aggregation function $\mathcal{A}_\omega$.
**Initialize:** $\forall i \in [m]$, set $s_0^{(i)} = 0$. Set $\mathbf{w}_1 = \mathbf{x}_1$. Each honest worker $i \in \mathcal{G}$ draws $\mathbf{z}^{(i)} \sim \mathcal{D}$ and set $\mathbf{d}_1^{(i)} = \nabla f(\mathbf{x}_1; \mathbf{z}^{(i)})$.
**for** $t = 1$ **to** $T$ **do** ▷ *Server update*
  Receive $\mathbf{d}_{t-\tau_t}$ from worker $i \in [m]$ and update:
    $\mathbf{d}_t^{(i)} = \mathbf{d}_{t-\tau_t}$, $s_t^{(i)} = s_{t-1}^{(i)} + 1$; $\forall \underline{j \neq i}$: set $s_t^{(j)} = s_{t-1}^{(j)}$, and for $\underline{t > 1}$:$\mathbf{d}_t^{(j)} = \mathbf{d}_{t-1}^{(j)}$;
  Update server model:
    $\mathbf{w}_{t+1} = \Pi_{\mathcal{K}} \left( \mathbf{w}_t - \eta_t \alpha_t \mathcal{A}_\omega(\{\mathbf{d}_t^{(j)}, s_t^{(j)}\}_{j=1}^m) \right)$, & $\mathbf{x}_{t+1} = \frac{1}{\sum_{k=1}^{t+1} \alpha_k} \sum_{k=1}^{t+1} \alpha_k \mathbf{w}_k$
  Send $\mathbf{x}_t$ to worker $i$. If $i$ is an honest worker, it performs the following update:
    Worker $i$ draws $\mathbf{z}_t \sim \mathcal{D}$, computes $\mathbf{g}_t = \nabla f(\mathbf{x}_t; \mathbf{z}_t)$, & $\tilde{\mathbf{g}}_{t-\tau_t} = \nabla f(\mathbf{x}_{t-\tau_t}; \mathbf{z}_t)$,
    and updates: $\mathbf{d}_t = \mathbf{g}_t + (1 - \beta_t)(\mathbf{d}_{t-\tau_t} - \tilde{\mathbf{g}}_{t-\tau_t})$ ▷ *Worker update*
**end for**
**Output:** $\mathbf{x}_T$

Here, $\beta_t \in [0, 1]$ are *corrected momentum* weights. It can be shown by induction that $\mathbb{E}[\mathbf{d}_t] = \mathbb{E}[\nabla f(x_t)]$; however, in general, $\mathbb{E}[\mathbf{d}_t | x_t] \neq \nabla f(x_t)$, unlike standard SGD estimators. Nevertheless, [Levy, 2023] demonstrates that choosing *corrected momentum* weights $\beta_t := 1/t$ results in significant error reduction, with $\mathbb{E}\|\varepsilon_t\|^2 := \mathbb{E}\|\mathbf{d}_t - \nabla f(\mathbf{x}_t)\|^2 \leq O(\tilde{\sigma}^2/t)$ at step $t$, where $\tilde{\sigma}^2 \leq O(\sigma^2 + D^2\sigma_L^2)$. This indicates that variance decreases with $t$, contrasting with standard SGD where the variance $\mathbb{E}\|\varepsilon_t^{\text{SGD}}\|^2 := \mathbb{E}\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|^2$ remains uniformly bounded.

### 4.1 Asynchronous Robust $\mu^2$-SGD

Building upon these, we integrate the $\mu^2$-SGD with a $(c_\lambda, \lambda)$-weighted robust aggregator $\mathcal{A}_\omega$, as described in Alg. 2. At each iteration $t \in [T]$, the global $\mathcal{PS}$ receives an output from a certain worker and aggregates all workers' recent updates $\left\{\mathbf{d}_t^{(i)}\right\}_{i=1}^m$ by employing weights accordingly to the number of updates of each worker $\left\{s_t^{(i)}\right\}_{i=1}^m$. An honest worker $i$ arriving at iteration $t$ returns its corrected momentum $\mathbf{d}_t^{(i)}$ to the $\mathcal{PS}$, computed as:

$$\mathbf{d}_t^{(i)} = \mathbf{d}_{t-\tau_t} = \mathbf{g}_{t-\tau_t} + (1 - \beta_{t-\tau_t})(\mathbf{d}_{t-\tau_t-\tau_{t-\tau_t}} - \tilde{\mathbf{g}}_{t-\tau_t-\tau_{t-\tau_t}}) ,$$

where $\mathbf{g}_t := \nabla f(\mathbf{x}_t; \mathbf{z}_t)$, and $\tilde{\mathbf{g}}_{t-\tau_t} := \nabla f(\mathbf{x}_{t-\tau_t}; \mathbf{z}_t)$. Afterwards, the $\mathcal{PS}$ performs the AnyTime update step as follows:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{K}} \left( \mathbf{w}_t - \eta \alpha_t \mathcal{A}_\omega(\{\mathbf{d}_t^{(i)}, s_t^{(i)}\}_{i=1}^m) \right), \ \mathbf{x}_{t+1} = \frac{1}{\alpha_{1:t+1}} \sum_{k \in [t+1]} \alpha_k \mathbf{w}_k .$$

In the spirit of Levy [2023], Dahan and Levy [2024], we suggest employing $\beta_t := 1/s_t$, which effectively considers the entire individual gradient's history of each worker; this translates to a stochastic error bound of $\|\varepsilon_t^{(i)}\| \leq O(\tilde{\sigma}/s_t)$ for an honest worker $i$ arriving at iteration $t$. To achieve an error corresponding to the total number of honest iterations $t_{\mathcal{G}}$, specifically $\|\varepsilon_t\| \leq O(\tilde{\sigma}/t_{\mathcal{G}})$, as in the non-distributed setting [Levy, 2023], a weighted collective error across all honest workers should be considered with weights determined by the number of honest worker arrivals, as detailed in Theorem 4.1. The unique characteristics of $\mu^2$-SGD make it well-suited for the asynchronous Byzantine setting, where $\lambda < 1/2$ relates to the fraction of Byzantine iterations. The total iteration number $t$ matches the sum of the workers' frequencies ($\sum_{i \in [\mathcal{G}]} s_t^{(i)} = t_{\mathcal{G}}$), aligning with the weighted robust definition in Definition 3.1. Using other approaches like momentum [Karimireddy et al., 2020, 2021, Allouah et al., 2023] is less straightforward in the asynchronous Byzantine setting with the weighted robust definition. This complexity arises because an individual honest error $\|\varepsilon_t^{(i)}\| \lesssim O(\tilde{\sigma}/\sqrt{s_t})$ implies that weights should be $\sqrt{s_t}$ instead of $s_t$, which can be more challenging.

**Remark 4.1** (Memory and Computational Overhead of Algorithm 2)**.** *Algorithm 2 incurs additional memory and computational costs compared to the asynchronous Byzantine-free setting [Arjevani*

*et al., 2020], where the server stores only one worker's output and the global model. For robust performance, Algorithm 2 stores the latest outputs from all workers, increasing memory usage to $O(dm)$. Robust aggregation methods like $\omega$-CWMed [Yin et al., 2018] and $\omega$-GM [Chen et al., 2017, Acharya et al., 2022] add a computational cost of $O(dm \log m)$ per round, unlike asynchronous Byzantine-free settings where worker outputs are used without aggregation. Comparable overheads are observed in synchronous Byzantine-resilient methods, which similarly aggregate outputs from all workers. This reflects a necessary trade-off: achieving robustness inherently requires leveraging information from all workers to counteract the influence of potentially faulty ones.*

**Theorem 4.1.** *For a convex set $\mathcal{K}$ with bounded diameter $D$ and a function $f : \mathcal{K} \mapsto \mathbb{R}$, and assume the assumptions in Equations (2),(3),(4). Then Alg. 2 with parameters $\{\alpha_t = t\}_t$ and $\{\beta_t = 1/s_t\}_t$ ensures the following for every $t \in [T]$ and each honest worker $i \in \mathcal{G}$:*

$$\mathbb{E}\left\|\varepsilon_t^{(i)}\right\|^2 \leq \frac{\tilde{\sigma}^2}{s_t^{(i)}}, \quad \mathbb{E}\left\|\frac{1}{\sum_{i\in\mathcal{G}} s_t^{(i)}} \sum_{i\in\mathcal{G}} s_t^{(i)} \varepsilon_t^{(i)}\right\|^2 \leq \frac{\tilde{\sigma}^2}{t_{\mathcal{G}}},$$

*where $\varepsilon_t^{(i)} = \mathbf{d}_t^{(i)} - \nabla f(\mathbf{x}_t^{(i)})$, $\tilde{\sigma}^2 = 2\sigma^2 + 32D^2K^2\sigma_L^2$, and $t_{\mathcal{G}}$ is the total number of honest iterations up to the $t^{\text{th}}$ iteration.*

*Proof Sketch of Thm. 4.1.* The complete analysis is provided in App. B.1. It involves several key steps for an honest $i$ worker who arrives at iteration $t$:

1. Following Lemma B.1, the distance between successive query points:$\|\mathbf{x}_t^{(i)} - \mathbf{x}_{t-\tau_t}^{(i)}\| \leq \frac{4K}{s_t^{(i)}-1} D$
2. We analyze the recursive dynamics of the error term $\varepsilon_t^{(i)}$ by setting $\beta_t = \frac{1}{s_t^{(i)}}$ and obtain:
$$s_t^{(i)}\varepsilon_t^{(i)} = (\mathbf{g}_t^{(i)} - \nabla f(\mathbf{x}_t^{(i)})) + (s_t^{(i)} - 1)Z_t^{(i)} + (s_t^{(i)} - 1)\varepsilon_{t-\tau_t}^{(i)},$$
where $Z_t^{(i)} := \mathbf{g}_t^{(i)} - \nabla f(\mathbf{x}_t^{(i)}) - (\tilde{\mathbf{g}}_{t-\tau_t}^{(i)} - \nabla f(\mathbf{x}_{t-\tau_t}^{(i)}))$. Unrolling this recursion provides an explicit expression: $s_t^{(i)}\varepsilon_t^{(i)} = \sum_{k\in[s_t^{(i)}]} \mathcal{M}_k^{(i)}$, where $\mathcal{M}_{s_t^{(i)}}^{(i)} := \mathbf{g}_t^{(i)} - \nabla f(\mathbf{x}_t^{(i)}) + (s_t - 1)Z_t^{(i)}$; thus, $\{\mathcal{M}_k^{(i)}\}_{k\in[s_t^{(i)}]}$ is a martingale difference sequence.
3. Employing the above with Eq. (2) and (4), we have: $\mathbb{E}\|\mathcal{M}_k^{(i)}\|^2 \leq 2\sigma^2 + 32D^2K^2\sigma_L^2 = \tilde{\sigma}^2$.
4. Leveraging the properties of a martingale difference sequence, we have:
$$\mathbb{E}\left\|s_t^{(i)}\varepsilon_t^{(i)}\right\|^2 = \mathbb{E}\left\|\sum_{k\in\left[s_t^{(i)}\right]} \mathcal{M}_k^{(i)}\right\|^2 = \sum_{k\in\left[s_t^{(i)}\right]} \mathbb{E}\left\|\mathcal{M}_k^{(i)}\right\|^2 \leq \tilde{\sigma}^2 s_t^{(i)},$$
$$\mathbb{E}\left\|\sum_{i\in\mathcal{G}} s_t^{(i)}\varepsilon_t^{(i)}\right\|^2 = \mathbb{E}\left\|\sum_{i\in\mathcal{G}}\sum_{k\in\left[s_t^{(i)}\right]} \mathcal{M}_k^{(i)}\right\|^2 = \sum_{i\in\mathcal{G}}\sum_{k\in\left[s_t^{(i)}\right]} \mathbb{E}\left\|\mathcal{M}_k^{(i)}\right\|^2 \leq \tilde{\sigma}^2 \sum_{i\in\mathcal{G}} s_t^{(i)} = \tilde{\sigma}^2 t_{\mathcal{G}}.$$

□

**Remark 4.2.** *Compared to synchronous scenarios [Levy, 2023, Dahan and Levy, 2024], the variance $\tilde{\sigma}$ in Thm. 2 additionally includes the variance in the delay, denoted as $K$ (Eq. (5)). In balanced scheduling methods, like Round Robin [Langford et al., 2009], the impact of $K$ on the error becomes minor, as the delay $\tau_t^{(i)} = m$ is constant. In the case of constant delays, the factor $K$ equals 1.*

**Lemma 4.1.** *Let $\mathcal{A}_\omega$ be $(c_\lambda, \lambda)$-weighted robust aggregation rule and let $f : \mathcal{K} \mapsto \mathbb{R}$, where $\mathcal{K}$ is a convex set with bounded diameter $D$, and presume that the assumption in Equations (2),(3),(4) hold. Then invoking Alg. 2 with $\{\alpha_t = t\}_t$ and $\{\beta_t = 1/s_t\}_t$, ensures the following for any $t \in [T]$,*

$$\mathbb{E}\left\|\hat{\mathbf{d}}_t - \nabla f(\mathbf{x}_t)\right\|^2 \leq O\left(\underbrace{\frac{\tilde{\sigma}^2}{t} + \frac{c_\lambda m \tilde{\sigma}^2}{t}}_{\text{Variance}} + \underbrace{\frac{(\tau_t^{max} DL)^2}{t^2} + \frac{c_\lambda (\tau_t^{max} DL)^2}{t^2}}_{\text{Bias}}\right)$$

*where* $\hat{\mathbf{d}}_t = \mathcal{A}_\omega(\{\mathbf{d}_t^{(i)}, s_t^{(i)}\}_{i=1}^m)$, $\tau_t^{max} = \max_{i\in[m]}\{\tau_t^{(i)}\}$, *and* $\tilde{\sigma}^2 = 2\sigma^2 + 32D^2K^2\sigma_L^2$.

Lemma 4.1 shows that the error between our gradient estimator $\hat{\mathbf{d}}_t$ and the true gradient includes a bias term arising from the aggregation of delayed momentums. This is in contrast to the synchronous scenario [Dahan and Levy, 2024] where the error is solely variance-dependent without any bias component. However, this bias does not affect the overall excess loss (Theorem 4.2), which remains comparable to the optimal rate achieved in synchronous Byzantine settings (see Remark 4.5).

By integrating the weighted robust aggregator with the double momentum mechanism, we achieve the optimal convergence rate for the first time in an asynchronous Byzantine setting—a significant advancement over previous efforts.

**Theorem 4.2** (Asynchronous Byzantine $\mu^2$-SGD Guarantees)**.** *Let $\mathcal{A}_\omega$ be $(c_\lambda, \lambda)$-weighted robust aggregation rule and let $f$ be a convex function. Also, let us make the same assumptions as in Thm. 4.1, and let us denote $G^* := \|\nabla f(\mathbf{x}^*)\|$, where $\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathcal{K}} f(\mathbf{x})$. Then invoking Alg. 2 with $\{\alpha_t = t\}_t$ and $\{\beta_t = 1/s_t\}_t$, and using a learning rate $\eta \le 1/4LT$ guarantees,*

$$\mathbb{E}\left[f(\mathbf{x}_T) - f(\mathbf{w}^*)\right] \le O\left(\frac{G^*D + LD^2\mu_{max}(\sqrt{1+c_\lambda})}{T} + \frac{D\tilde{\sigma}(\sqrt{1+c_\lambda m})}{\sqrt{T}}\right)$$

*where* $\tilde{\sigma}^2 = 2\sigma^2 + 32D^2K^2\sigma_L^2$, $\mu_{max} = \frac{1}{T}\sum_{t\in[T]}\tau_t^{max}$, *and* $\tau_t^{max} = \max_{i\in[m]}\{\tau_t^{(i)}\}$.

**Remark 4.3.** *In the absence of Byzantine iterations ($\lambda = 0$), the parameter $c_\lambda$ of a $(c_\lambda, \lambda)$-weighted robust aggregator can diminish to 0 when we use $\omega$-CTMA (see Table 1). This aligns with the asynchronous SGD analysis [Arjevani et al., 2020] and represents the first work to achieve optimal convergence without Byzantine workers compared to previous efforts [Yang and Li, 2021, 2023, Fang et al., 2022, Zhu et al., 2023, Damaskinos et al., 2018, Xie et al., 2020b, Zhu et al., 2024].*

**Remark 4.4.** *Unlike previous works [Yang and Li, 2021, 2023, Fang et al., 2022, Zhu et al., 2023, Damaskinos et al., 2018, Xie et al., 2020b, Zhu et al., 2024], our convergence rate is independent of data dimensionality $d$ and is sublinear at $T$, even in the presence of Byzantine workers.*

**Remark 4.5.** *This result is consistent with the synchronous scenario [Dahan and Levy, 2024], where the delay is constant $\tau_t = m$ as in Round Robin [Langford et al., 2009]. In this case, the proportion of Byzantine workers is $\lambda$, and the asynchronous excess loss is $\le O\left(\frac{LD^2m}{T} + \frac{D\tilde{\sigma}\sqrt{1+c_\lambda m}}{\sqrt{T}}\right)$. In comparison to the synchronous case, where $m$ workers perform $R$ rounds, here we make $R$ query point updates and $T = Rm$ data-samples, resulting in synchronous excess loss $\le O\left(\frac{LD^2}{R} + \frac{D\tilde{\sigma}\sqrt{1/m+c_\lambda}}{\sqrt{R}}\right) = O\left(\frac{LD^2m}{T} + \frac{D\tilde{\sigma}\sqrt{1+mc_\lambda}}{\sqrt{T}}\right)$ [Dahan and Levy, 2024].*

## 5 Experiments

To evaluate the effectiveness of our proposed approach, we conducted experiments on MNIST [LeCun et al., 2010] and CIFAR-10 [Krizhevsky et al., 2014] datasets—two recognized benchmarks in image classification tasks. We employed a two-layer convolutional neural network architecture for both datasets, implemented using the PyTorch framework. The training was performed using the cross-entropy loss function, and all computations were executed on an NVIDIA RTX 3090 GPU. To ensure the robustness of our findings, each experiment was repeated with three different random seeds, and the results were averaged accordingly. Our experimental results demonstrate consistent performance across both datasets. Further details about the experimental setup and the complete results are provided in Appendix D.

**Weighted vs. Non-Weighted Robust Aggregators**. We evaluated the test accuracy of weighted and non-weighted robust aggregators in imbalanced asynchronous Byzantine environments. Our experiments show that weighted robust aggregators consistently achieved higher test accuracy than the non-weighted ones (see Figure 2 and Figure 5). This highlights the benefit of prioritizing workers who contribute more updates in asynchronous setups.

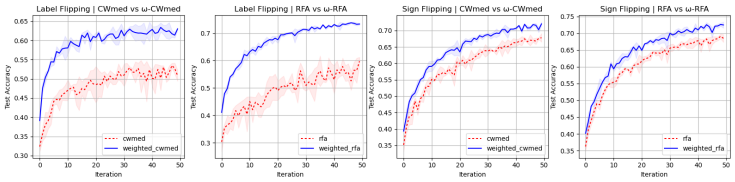


Figure 2: **CIFAR-10**. **Test Accuracy of Weighted vs. Non-Weighted Robust Aggregators**. This scenario involves 17 workers, including 8 Byzantine workers, with workers' arrival probabilities proportional to the square of their IDs. We used the $\mu^2$-SGD in this scenario. Left: *label flipping*, $\lambda = 0.3$. Right: *sign flipping*, $\lambda = 0.4$.

**Effectiveness of $\omega$-CTMA**. We evaluated the test accuracy of weighted robust aggregators with and without the integration of $\omega$-CTMA, as shown in Figure 3 and Figure 6. The results demonstrate that $\omega$-CTMA can enhance the performance of weighted robust aggregators in various Byzantine scenarios. Notably, $\omega$-CTMA may maintain high accuracy even when other robust aggregators fail, as seen with the Empire attack result for both datasets.

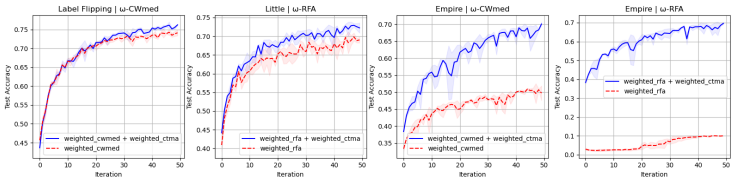


Figure 3: **CIFAR-10**. **Test Accuracy Comparison of Weighted Robust Aggregators With and Without $\omega$-CTMA**. This scenario involves 9 workers with 1 or 3 Byzantine workers. Workers' arrival probabilities are proportional to their IDs, and we used $\mu^2$-SGD. On the *left*, we have the *label flipping* and *little* attacks with $\lambda = 0.1$ and $\lambda = 0.2$ for 1 and 3 Byzantine workers, respectively. On the *right*, the *empire* attack, each with $\lambda = 0.4$ and 3 Byzantine workers.

**Performance of $\mu^2$-SGD vs. Standard Momentum and SGD**. We evaluated the test accuracy of $\mu^2$-SGD in comparison to standard momentum [Polyak, 1964] and SGD [eon Bottou, 1998] within an asynchronous Byzantine setup. Figure 4 and Figure 7 show that $\mu^2$-SGD performs on par with standard momentum, while SGD generally exhibits poorer performance relative to both. These results underscore the importance of utilizing historical information when addressing Byzantine scenarios.

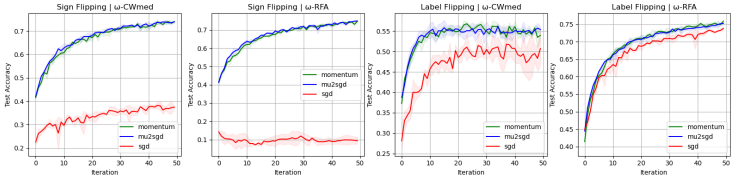


Figure 4: **CIFAR-10**. **Test Accuracy Comparison Among Different Optimizers**. This scenario involves 9 workers (4 Byzantine) with $\lambda = 0.4$ for the first three from *left* to *right*, and $\lambda = 0.3$ for the *label flipping* attack on the *left*. Workers' arrival probabilities are proportional to their IDs.

## Conclusions and Future Work

This paper shows that using a double momentum approach, which incorporates the entire history of each honest worker, improves the error bound to be proportional to the total number of updates when considering their weighted average in asynchronous settings. By integrating this method with a weighted robust framework, $\mu^2$-SGD achieves an optimal convergence rate, making it particularly effective for asynchronous Byzantine environments. However, integrating other optimization algorithms, like momentum, into this weighted robust framework can be challenging, as they do not achieve an error bound proportional to the total number of updates and may complicate the adjustment of weights based on the update count. This highlights the need for further research to adapt different methods to the spirit of this framework in non-convex and convex settings.

## Acknowledgement

This research was partially supported by Israel PBC-VATAT, the Technion Artificial Intelligent Hub (Tech.AI), and the Israel Science Foundation (grant No. 3109/24).

## References

Anish Acharya, Abolfazl Hashemi, Prateek Jain, Sujay Sanghavi, Inderjit S Dhillon, and Ufuk Topcu. Robust training in high dimensions via block coordinate geometric median descent. In *International Conference on Artificial Intelligence and Statistics*, pages 11145–11168. PMLR, 2022.

Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. *Advances in Neural Information Processing Systems*, 31, 2018.

Zeyuan Allen-Zhu, Faeze Ebrahimian, Jerry Li, and Dan Alistarh. Byzantine-resilient non-convex stochastic gradient descent. *arXiv preprint arXiv:2012.14368*, 2020.

Youssef Allouah, Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafaël Pinot, and John Stephan. Fixing by mixing: A recipe for optimal byzantine ml under heterogeneity. In *International Conference on Artificial Intelligence and Statistics*, pages 1232–1300. PMLR, 2023.

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

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*, pages 436–445. PMLR, 2021.

Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017.

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*, pages 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. Fault tolerant ml: Efficient meta-aggregation and synchronous training. In *Forty-first International Conference on Machine Learning*, 2024.

Georgios Damaskinos, Rachid Guerraoui, Rhicheek Patra, Mahsa Taziki, et al. Asynchronous byzantine machine learning (the case of sgd). In *International Conference on Machine Learning*, pages 1145–1154. PMLR, 2018.

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.

El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Louis Alexandre Rouault. Distributed momentum for byzantine-resilient stochastic gradient descent. In *9th International Conference on Learning Representations (ICLR)*, number CONF, 2021.

L eon Bottou. Online learning and stochastic approximations. *Online learning in neural networks*, 17(9):142, 1998.

Minghong Fang, Jia Liu, Neil Zhenqiang Gong, and Elizabeth S Bentley. Aflguard: Byzantine-robust asynchronous federated learning. In *Proceedings of the 38th Annual Computer Security Applications Conference*, pages 632–646, 2022.

Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, and John Stephan. Byzantine machine learning made easy by resilient averaging of momentums. In *International Conference on Machine Learning*, pages 6246–6283. PMLR, 2022.

Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, pages 3521–3530. PMLR, 2018.

Rachid Guerraoui, Nirupam Gupta, and Rafael Pinot. Byzantine machine learning: A primer. *ACM Computing Surveys*, 2023.

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. *arXiv preprint arXiv:2006.09365*, 2020.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *International Conference on Machine Learning*, pages 5311–5319. PMLR, 2021.

Ali Kavis, Kfir Y Levy, Francis Bach, and Volkan Cevher. Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization. *Advances in neural information processing systems*, 32, 2019.

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

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the works of leslie lamport*, pages 203–226. 2019.

John Langford, Alexander Smola, and Martin Zinkevich. Slow learners are fast. *arXiv preprint arXiv:0911.0491*, 2009.

Yann LeCun, Corinna Cortes, Chris Burges, et al. Mnist handwritten digit database, 2010. URL http://yann.lecun.com/exdb/mnist/. Licensed under CC BY-SA 3.0, available at https://creativecommons.org/licenses/by-sa/3.0/.

Kfir Y Levy. $\mu^2$-sgd: Stable stochastic optimization via a double momentum mechanism. *arXiv preprint arXiv:2304.04172*, 2023.

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.

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

Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Uncertainty in Artificial Intelligence*, pages 261–270. PMLR, 2020a.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno++: Robust fully asynchronous sgd. In *International Conference on Machine Learning*, pages 10495–10503. PMLR, 2020b.

Yi-Rui Yang and Wu-Jun Li. Basgd: Buffered asynchronous sgd for byzantine learning. In *International Conference on Machine Learning*, pages 11751–11761. PMLR, 2021.

Yi-Rui Yang and Wu-Jun Li. Buffered asynchronous sgd for byzantine learning. *Journal of Machine Learning Research*, 24(204):1–62, 2023.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018.

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.

Zehan Zhu, Yan Huang, Chengcheng Zhao, and Jinming Xu. Asynchronous byzantine-robust stochastic aggregation with variance reduction for distributed learning. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 151–158. IEEE, 2023.

Zehan Zhu, Yan Huang, Chengcheng Zhao, and Jinming Xu. Asynchronous byzantine-robust stochastic aggregation with variance reduction for distributed learning. 2024.

# A Bounded Smoothness Variance Assumption

We show that Eq. (3) implies that Eq. (4) holds for some $\sigma_L^2 \in [0, L^2]$.

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

Here, we also used $\mathbb{E}[\nabla f(\mathbf{x};\mathbf{z}) - \nabla f(\mathbf{y};\mathbf{z})] = (\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}))$, and followed Eq. (3). Therefore, we establish that $\sigma_L^2 \in [0, L^2]$.

# B Asynchronous Robust Convex Analysis

## B.1 Proof of Thm. 4.1

*Proof of Thm. 4.1.* To simplify the discussion, let's introduce some notations for a worker $i \in \mathcal{G}$, who arrives at time $t$:

$$\tilde{\mathbf{x}}_{s_t} := \mathbf{x}_{t-\tau_t} = \mathbf{x}_t^{(i)}, \quad \tilde{\mathbf{x}}_{s_t-1} := \mathbf{x}_{t-\tau_t-\tau_{t-\tau_t}} = \mathbf{x}_{t-\tau_t}^{(i)} \\ \tilde{\varepsilon}_{s_t} := \varepsilon_{t-\tau_t} = \varepsilon_t^{(i)}, \quad \tilde{\varepsilon}_{s_t-1} := \varepsilon_{t-\tau_t-\tau_{t-\tau_t}} = \varepsilon_{t-\tau_t}^{(i)} \\ \mathbf{h}_{s_t} := \mathbf{g}_{t-\tau_t} = \mathbf{g}_t^{(i)}, \quad \tilde{\mathbf{h}}_{s_t-1} := \tilde{\mathbf{g}}_{t-\tau_t-\tau_{t-\tau_t}} = \tilde{\mathbf{g}}_{t-\tau_t}^{(i)}$$

Next, we will employ the following lemma that bounds the distance between the averages $\mathbf{x}_t, \mathbf{x}_{t-\tau_t}$. Recall that $\mathbf{x}_t$ and $\mathbf{x}_{t-\tau_t}$ are consecutive query points for the worker $i$ that arrives at time $t$.

**Lemma B.1** (Aviv et al. [2021]). *Let $f : \mathcal{K} \mapsto \mathbb{R}$, where $\mathcal{K}$ is a convex set with bounded diameter $D$. Then invoking Alg. 2 with $\{\alpha_t = t\}_t$ ensures the following for any $t \in [T]$,*

$$\|\mathbf{x}_t - \mathbf{x}_{t-\tau_t}\| \leq \frac{4D\tau_t}{t} \, .$$

For completeness, we provide a proof in Section B.1.1.

Next, we define $\mu_t$ be the average delay of the worker $i$ that arrives at iteration $t$, i.e.,

$$s_t = \frac{t}{\mu_t}, \quad s_t - 1 = s_{t-\tau_t} = \frac{t-\tau_t}{\mu_{t-\tau_t}} \, .$$

From Equation (5), we infer that,

$$\tau_{min}^{(i)} \leq \mu_t \leq K\tau_{min}^{(i)} \, . \tag{7}$$

Following this, we analyze the upper bound on the distance between two successive query points for an honest worker $i$ that arrives at time $t$:

$$\|\tilde{\mathbf{x}}_{s_t} - \tilde{\mathbf{x}}_{s_t-1}\| = \|\mathbf{x}_{t-\tau_t} - \mathbf{x}_{t-\tau_t-\tau_{t-\tau_t}}\| \leq \frac{4\tau_{t-\tau_t}}{t-\tau_t}D = \frac{4\tau_{t-\tau_t}}{(\mu_{t-\tau_t})(s_t-1)}D \leq \frac{4K}{s_t-1}D \, , \tag{8}$$

where the first inequality follows Lemma B.1. The second equality utilizes the relation $s_t - 1 = \frac{t-\tau_t}{\mu_{t-\tau_t}}$. The final inequality stems from the assumptions in Eq. (5) and Eq. (7).

**Remark:** Before proceeding with the analysis, we shall condition the (possible randomization) in the delays of all workers; and recall that the data-samples are independent of the delays. Thus, the expectations that we take are only with respect to the randomization in the data-samples and are conditioned on the delays. Thus, this conditioning allows us to treat the delays $\tau_t^{(i)}$'s and number of updates $s_t^{(i)}$'s as fixed and predefined.

We proceed to analyze the recursive dynamics of $\tilde{\varepsilon}_{s_t}$ for each $i \in \mathcal{G}$. Based on the definitions of $\mathbf{d}_t$ and $\varepsilon_t$, we can present the recursive relationship in the following way:

$$\tilde{\varepsilon}_{s_t} = \beta_t(\mathbf{h}_{s_t} - \nabla f(\tilde{\mathbf{x}}_{s_t})) + (1-\beta_t)Z_{s_t} + (1-\beta_t)\tilde{\varepsilon}_{s_t-1} \, ,$$

where $Z_{s_t} := \mathbf{h}_{s_t} - \nabla f(\tilde{\mathbf{x}}_{s_t}) - (\tilde{\mathbf{h}}_{s_t-1} - \nabla f(\tilde{\mathbf{x}}_{s_t-1}))$. Upon choosing $\beta_t = \frac{1}{s_t}$, we can reformulate the above equation as follows:

$$s_t \tilde{\varepsilon}_{s_t} = (\mathbf{h}_{s_t} - \nabla f(\tilde{\mathbf{x}}_{s_t})) + (s_t - 1) Z_{s_t} + (s_t - 1)\tilde{\varepsilon}_{s_t-1} \, .$$

Unrolling this recursion yields an explicit expression for any $s_t \geq 1$:

$$s_t \tilde{\varepsilon}_{s_t} = \sum_{k \in [s_t]} \mathcal{M}_k^{(i)} \, , \tag{9}$$

where we have defined,

$$\mathcal{M}_k^{(i)} := \mathbf{h}_k - \nabla f(\tilde{\mathbf{x}}_k) + (k-1) Z_k \, , \tag{10}$$

and $k$ is a counter for the iterations where worker $i$ makes an update.

Following this, we derive an upper bound for the expected square norm of $\mathcal{M}_k^{(i)}$ as follows:

$$\begin{aligned}
\mathbb{E}\|\mathcal{M}_k^{(i)}\|^2 &\leq 2\mathbb{E}\|\mathbf{h}_k - \nabla f(\tilde{\mathbf{x}}_k)\|^2 + 2(k-1)^2 \mathbb{E}\|(\mathbf{h}_k - \nabla f(\tilde{\mathbf{x}}_k)) - (\tilde{\mathbf{h}}_{k-1} - \nabla f(\tilde{\mathbf{x}}_{k-1}))\|^2 \\
&\leq 2\sigma^2 + 2\sigma_L^2 (k-1)^2 \mathbb{E}\|\mathbf{x}_k - \mathbf{x}_{k-1}\|^2 \\
&\leq 2\sigma^2 + 32 D^2 K^2 \sigma_L^2 = \tilde{\sigma}^2 \, ,
\end{aligned} \tag{11}$$

where the first inequality uses $\|\mathbf{a} + \mathbf{b}\|^2 \leq 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$, which holds $\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^d$. The second inequality aligns with the assumptions outlined in Equations (2) and (4). The third inequality uses Eq. (8).

**Establishing the First Part of the Theorem:** Before continuing, it is natural to define an ordered set of samples $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{t_\mathcal{G}}\}$ such that these samples are associated with honest and consecutive updates (or iterates) of the $\mathcal{PS}$, and $t_\mathcal{G}$ is the total number of honest updates up to time $t$. Concretely, the $\tau^{\text{th}}$ honest update of the $\mathcal{PS}$ is based on an honest worker that utilizes a fresh sample $\mathbf{z}_\tau$.

Now, for a given worker $i$ we shall define the filtration associated with his updates. Concretely, let $k \in \{1, \ldots s_T^{(i)}\}$. Then we define $\mathcal{F}_k^{(i)}$ to be the sigma-algebra induces by the sequence of samples $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{t_\mathcal{G}}\}$ up to the $k^{\text{th}}$ update of worker $i$. And it is easy to see that $\{\mathcal{F}_k^{(i)}\}_{k \in [s_t^{(i)}]}$ is a filtration. Moreover, it can be directly shown that for a given worker $i$, then the above defined sequence $\{\mathcal{M}_k^{(i)}\}_{k \in [s_t^{(i)}]}$ (see Eq. (10)) is a martingale difference sequence with respect to $\{\mathcal{F}_k^{(i)}\}_{k \in [s_t^{(i)}]}$. This allows us to directly employ Lemma B.2 below, which yields,

$$\mathbb{E}\left\| s_t^{(i)} \varepsilon_t^{(i)} \right\|^2 = \mathbb{E}\left\| \sum_{k \in \left[s_t^{(i)}\right]} \mathcal{M}_k^{(i)} \right\|^2 = \sum_{k \in \left[s_t^{(i)}\right]} \mathbb{E}\left\| \mathcal{M}_k^{(i)} \right\|^2 \leq \tilde{\sigma}^2 s_t^{(i)} \, ,$$

where we have also used Equations (9) and (11). Thus, the above bounds establish the first part of the theorem.

**Lemma B.2** (See e.g. Lemma B.1 in Levy [2023]). *Let $\{M_t\}_t$ be a martingale difference sequence with respect to a filtration $\{\mathcal{F}_t\}_t$, then the following holds for any $t$,*

$$\mathbb{E}\left\| \sum_{\tau \in [t]} M_\tau \right\|^2 = \sum_{\tau \in [t]} \mathbb{E}\left\| M_\tau \right\|^2 \, .$$

**Establishing the Second Part of the Theorem:** As before, we define an ordered set of samples $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{t_\mathcal{G}}\}$ such that these samples are associated with honest and consecutive updates (or iterates) of the $\mathcal{PS}$, and $t_\mathcal{G}$ is the total number of honest updates up to time $t$. Concretely, the $\tau^{\text{th}}$ honest update of the $\mathcal{PS}$ is based on an honest worker that utilizes a fresh sample $\mathbf{z}_\tau$. We shall also define $\{\mathcal{F}_\tau\}_{\tau \in [t_\mathcal{G}]}$ be the natural filtration induced by the ordered sequence of data samples.

Moreover, for a given sample $\mathbf{z}_\tau \in \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{t_\mathcal{G}}\}$, let $i_\tau \in [m]$ be the worker that is associated with the $\tau^{\text{th}}$ honest update of the $\mathcal{PS}$ , with a fresh sample $\mathbf{z}_\tau$. In this case, we shall define:

$$\mathcal{M}_\tau := \mathcal{M}_{s_\tau^{(i_\tau)}}^{(i_\tau)} \, .$$

where $\mathcal{M}_k^{(i)}$ is defined in Eq. (10). It is immediate to show that $\{\mathcal{M}_\tau\}_{\tau\in[t_\mathcal{G}]}$ is a martingale difference sequence with respect to $\{\mathcal{F}_\tau\}_{\tau\in[t_\mathcal{G}]}$. Moreover, the following holds directly be the definition of $\mathcal{M}_\tau$:

$$\sum_{i\in\mathcal{G}}\sum_{k\in\left[s_t^{(i)}\right]}\mathcal{M}_k^{(i)} = \sum_{\tau=1}^{t_\mathcal{G}}\mathcal{M}_\tau \ . \tag{12}$$

Now using Eq. (9) the following holds,

$$\sum_{i\in\mathcal{G}} s_t^{(i)}\varepsilon_t^{(i)} = \sum_{i\in\mathcal{G}}\sum_{k\in\left[s_t^{(i)}\right]}\mathcal{M}_k^{(i)} \ . \tag{13}$$

Combining Equations (12) and (13) together with Lemma B.2 establishes the second part of the theorem,

$$\mathbb{E}\left\|\sum_{i\in\mathcal{G}} s_t^{(i)}\varepsilon_t^{(i)}\right\|^2 = \mathbb{E}\left\|\sum_{i\in\mathcal{G}}\sum_{k\in\left[s_t^{(i)}\right]}\mathcal{M}_k^{(i)}\right\|^2 = \mathbb{E}\left\|\sum_{\tau=1}^{t_\mathcal{G}}\mathcal{M}_\tau\right\|^2 = \sum_{\tau=1}^{t_\mathcal{G}}\mathbb{E}\left\|\mathcal{M}_\tau\right\|^2 \le \tilde{\sigma}^2 t_\mathcal{G} \ .$$

where the inequality uses the bound in Eq. (11). □

### B.1.1 Proof of Lemma B.1

*Proof.* We borrowed the following steps from Aviv et al. [2021]. Let's define $\mathbf{y}\in\mathcal{K}$ as the average of $\mathbf{x}_t$ over the interval $[t-\tau_t, t]$, i.e.,

$$\mathbf{y} := \frac{1}{\alpha_{t-\tau_t+1:t}}\sum_{i=t-\tau_t+1}^{t}\alpha_i\mathbf{w}_i \ .$$

Then we have the following relationship:

$$\alpha_{1:t}\mathbf{x}_t = \sum_{i=1}^{t}\alpha_i\mathbf{w}_i = \sum_{i=1}^{t-\tau_t}\alpha_i\mathbf{w}_i + \sum_{i=t-\tau_t+1}^{t}\alpha_i\mathbf{w}_i = \alpha_{1:t-\tau_t}\mathbf{x}_{t-\tau_t} + \alpha_{t-\tau_t+1:t}\mathbf{y} \ .$$

Hence,

$$\alpha_{1:t-\tau_t}(\mathbf{x}_t - \mathbf{x}_{t-\tau_t}) = \alpha_{t-\tau_t+1:t}(\mathbf{y}-\mathbf{x}_t) \ .$$

By setting $\alpha_t = t$, we have that,

$$\begin{aligned}
\|\mathbf{x}_t - \mathbf{x}_{t-\tau_t}\| &= \frac{\alpha_{t-\tau_t+1:t}}{\alpha_{1:t-\tau_t}}\|\mathbf{y}-\mathbf{x}_t\| \\
&= \frac{\tau_t(t-\tau_t+1+t)}{(t-\tau_t)(t-\tau_t+1)}\|\mathbf{y}-\mathbf{x}_t\| \\
&\le \frac{\tau_t(t-\tau_t+1)}{(t-\tau_t)(t-\tau_t+1)}\|\mathbf{y}-\mathbf{x}_t\| + \frac{t\tau_t}{(t-\tau_t)^2}\|\mathbf{y}-\mathbf{x}_t\| \\
&= \frac{\tau_t}{t-\tau_t}\|\mathbf{y}-\mathbf{x}_t\| + \frac{t\tau_t}{(t-\tau_t)^2}\|\mathbf{y}-\mathbf{x}_t\| \ .
\end{aligned}$$

For $t \ge 3\tau_t$, we have that,

$$\|\mathbf{x}_t - \mathbf{x}_{t-\tau_t}\| \le \frac{3\tau_t D}{2t} + \frac{9\tau_t D}{4t} \le \frac{4\tau_t D}{t} \ .$$

Given that the domain is bounded, $\|\mathbf{x}_t - \mathbf{x}_{t-\tau_t}\| \le D\ \forall t$, for $t < 3\tau_t$, we have $D < \frac{4\tau_t D}{t}$. Combining these results, we conclude:

$$\|\mathbf{x}_t - \mathbf{x}_{t-\tau_t}\| \le \frac{4\tau_t D}{t} \ .$$

□

### B.2 Proof of Lemma 4.1

*Proof of Lemma 4.1.*

**Lemma B.3.** *Let $f : \mathcal{K} \mapsto \mathbb{R}$, where $\mathcal{K}$ is a convex set with bounded diameter $D$. Then invoking Alg. 2 with $\{\alpha_t = t\}_t$ ensures the following for any $t \in [T]$, and every $i, j \in [m]$,*

$$\left\|\mathbf{x}_t^{(i)} - \mathbf{x}_t^{(j)}\right\| \leq \frac{4D\left(\tau_t^{(i)} + \tau_t^{(j)}\right)}{t} .$$

*Proof.*

$$\left\|\mathbf{x}_t^{(i)} - \mathbf{x}_t^{(j)}\right\| \leq \left\|\mathbf{x}_t^{(i)} - \mathbf{x}_t\right\| + \left\|\mathbf{x}_t - \mathbf{x}_t^{(j)}\right\| \leq \frac{4D\tau_t^{(i)}}{t} + \frac{4D\tau_t^{(j)}}{t} ,$$

where the first inequality is a result of the triangle inequality, and the second follows Lemma B.1. □

**Bias Bounds.** Here's a refined version:

We begin by analyzing the upper bound of the bias in the collective gradients of honest workers up to time $t$ in relation to the gradient at that time, denoted as $\mathcal{B}_t^1$. Following this, we derive the upper bound for the bias between the collective gradients of these honest workers and the gradient of an individual honest worker, also up to time $t$, which we denote as $\mathcal{B}_t^2$. For clarity, we define $\bar{\nabla}_{\mathcal{G},t} := \frac{1}{\sum_{i\in\mathcal{G}} s_t^{(i)}} \sum_{i\in\mathcal{G}} s_t^{(i)} \nabla f(\mathbf{x}_t^{(i)})$.

$$\begin{aligned}
\left\|\mathcal{B}_t^1\right\| &:= \mathbb{E}\left\|\bar{\nabla}_{\mathcal{G},t} - \nabla f(\mathbf{x}_t)\right\| \\
&= \mathbb{E}\left\|\frac{1}{\sum_{i\in\mathcal{G}} s_t^{(i)}} \sum_{i\in\mathcal{G}} s_t^{(i)} \nabla f(\mathbf{x}_t^{(i)}) - \nabla f(\mathbf{x}_t)\right\| \\
&\leq \mathbb{E}\left[\frac{1}{\sum_{i\in\mathcal{G}} s_t^{(i)}} \sum_{i\in\mathcal{G}} s_t^{(i)} \left\|\nabla f(\mathbf{x}_t^{(i)}) - \nabla f(\mathbf{x}_t)\right\|\right] \\
&\leq \frac{L}{\sum_{i\in\mathcal{G}} s_t^{(i)}} \mathbb{E}\left[\sum_{i\in\mathcal{G}} s_t^{(i)} \left\|\mathbf{x}_t^{(i)} - \mathbf{x}_t\right\|\right] \\
&\leq \frac{4DL}{\sum_{i\in\mathcal{G}} s_t^{(i)}} \mathbb{E}\left[\sum_{i\in\mathcal{G}} s_t^{(i)} \frac{\tau_t^{(i)}}{t}\right] \\
&\leq \frac{4\tau_t^{max} DL}{t} .
\end{aligned} \tag{14}$$

Here, the first inequality leverages Jensen's inequality, and the second follows the smoothness assumption in Eq. (3). The third follows Lemma B.1, and the last inequality follows that $\tau_t^{max} := \max_{i\in[m]}\{\tau_t^{(i)}\}$.

For the second bias $\mathcal{B}_t^2$, we have:

$$\begin{aligned}
\mathbb{E}\left\|\mathcal{B}_t^2\right\| &:= \mathbb{E}\left\|\nabla f(\mathbf{x}_t^{(i)}) - \bar{\nabla}_{\mathcal{G},t}\right\| \\
&= \mathbb{E}\left\|\nabla f(\mathbf{x}_t^{(i)}) - \frac{1}{\sum_{j\in\mathcal{G}} s_t^{(j)}} \sum_{j\in\mathcal{G}} s_t^{(j)} \nabla f(\mathbf{x}_t^{(j)})\right\| \\
&\leq \frac{1}{\sum_{j\in\mathcal{G}} s_t^{(j)}} \mathbb{E}\left[\sum_{j\in\mathcal{G}} s_t^{(j)} \left\|\nabla f(\mathbf{x}_t^{(i)}) - \nabla f(\mathbf{x}_t^{(j)})\right\|\right] \\
&\leq \frac{L}{\sum_{j\in\mathcal{G}} s_t^{(j)}} \mathbb{E}\left[\sum_{j\in\mathcal{G}} s_t^{(j)} \left\|\mathbf{x}_t^{(i)} - \mathbf{x}_t^{(j)}\right\|\right] \\
&\leq \frac{4DL}{\sum_{j\in\mathcal{G}} s_t^{(j)}} \mathbb{E}\left[\sum_{j\in\mathcal{G}} s_t^{(j)} \left(\frac{\tau_t^{(i)} + \tau_t^{(j)}}{t}\right)\right] \\
&\leq \frac{8\tau_t^{max} DL}{t}\,. \qquad (15)
\end{aligned}$$

Like before, the first inequality leverages Jensen's inequality, and the second follows the smoothness assumption in Eq. (3), the third inequality follows Lemma B.3, and the last one follows that $\tau_t^{max} := \max_{i\in[m]}\{\tau_t^{(i)}\}$.

**Variance Bound.** We start by determining $\rho_i$ as outlined in Definition 3.1:

$$\begin{aligned}
\mathbb{E}\|\mathbf{d}_t^{(i)} - \bar{\mathbf{d}}_{\mathcal{G},t}\|^2 &\leq 3\mathbb{E}\|\mathbf{d}_t^{(i)} - \nabla f(\mathbf{x}_t^{(i)})\|^2 + 3\mathbb{E}\|\bar{\nabla}_{\mathcal{G},t} - \bar{\mathbf{d}}_{\mathcal{G},t}\|^2 + 3\mathbb{E}\left\|\mathcal{B}_t^2\right\|^2 \\
&= 3\mathbb{E}\left\|\varepsilon_t^{(i)}\right\|^2 + 3\mathbb{E}\left\|\frac{1}{\sum_{i\in\mathcal{G}} s_t^{(i)}} \sum_{i\in\mathcal{G}} s_t^{(i)} \varepsilon_t^{(i)}\right\|^2 + 3\mathbb{E}\left\|\mathcal{B}_t^2\right\|^2 \\
&\leq \frac{3\tilde{\sigma}^2}{s_t^{(i)}} + \frac{3\tilde{\sigma}^2}{t_{\mathcal{G}}} + \frac{192(\tau_t^{max} DL)^2}{t^2} \\
&\leq \frac{6\tilde{\sigma}^2}{s_t^{(i)}} + \frac{192(\tau_t^{max} DL)^2}{t^2}\,,
\end{aligned}$$

where $\bar{\mathbf{d}}_{\mathcal{G},t} := \frac{1}{\sum_{i\in\mathcal{G}} s_t^{(i)}} \sum_{i\in\mathcal{G}} s_t^{(i)} \mathbf{d}_t^{(i)}$. The first and inequality uses $\|\mathbf{a}+\mathbf{b}+\mathbf{c}\|^2 \leq 3\|\mathbf{a}\|^2 + 3\|\mathbf{b}\|^2 + 3\|\mathbf{c}\|^2$, which holds $\forall \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^d$. The second inequality follows Theorem 4.1 and employs the second bias bound in Eq. (15). The third uses the fact that $s_t^{(i)} \leq t_{\mathcal{G}}, \forall i \in \mathcal{G}$. Accordingly, we set $\rho_i^2 := \frac{6\tilde{\sigma}^2}{s_t^{(i)}} + \frac{192(\tau_t^{max} DL)^2}{t^2}$.

Following this, we derive $\rho$ as outlined in Definition 3.1:

$$\rho^2 = \frac{1}{\sum_{i\in\mathcal{G}} s_t^{(i)}} \sum_{i\in\mathcal{G}} s_t^{(i)} \rho_i^2 = \frac{1}{t_{\mathcal{G}}} \sum_{i\in\mathcal{G}} s_t^{(i)} \left(\frac{6\tilde{\sigma}^2}{s_t^{(i)}} + \frac{192(\tau_t^{max} DL)^2}{t^2}\right) = \frac{6m\tilde{\sigma}^2}{t_{\mathcal{G}}} + \frac{192(\tau_t^{max} DL)^2}{t^2}\,. \qquad (16)$$

Next, we establish an upper bound for $\mathbb{E}\|\mathcal{E}_t\|^2$:

$$\begin{aligned}
\mathbb{E}\|\mathcal{E}_t\|^2 &= \mathbb{E}\left\|\hat{\mathbf{d}}_t - \nabla f(\mathbf{x}_t)\right\|^2 \\
&\leq 2\mathbb{E}\left\|\hat{\mathbf{d}}_t - \bar{\mathbf{d}}_{\mathcal{G},t}\right\|^2 + 2\mathbb{E}\left\|\bar{\mathbf{d}}_{\mathcal{G},t} - \nabla f(\mathbf{x}_t)\right\|^2 \\
&\leq 2c_\lambda\left(\frac{6m\tilde{\sigma}^2}{t_{\mathcal{G}}} + \frac{192(\tau_t^{max}DL)^2}{t^2}\right) + 4\mathbb{E}\left\|\bar{\mathbf{d}}_{\mathcal{G},t} - \bar{\nabla}_{\mathcal{G},t}\right\|^2 + 4\mathbb{E}\left\|\mathcal{B}_t^1\right\|^2 \\
&= 2c_\lambda\left(\frac{6m\tilde{\sigma}^2}{t_{\mathcal{G}}} + \frac{192(\tau_t^{max}DL)^2}{t^2}\right) + 4\mathbb{E}\left\|\frac{1}{\sum_{i\in\mathcal{G}} s_t^{(i)}}\sum_{i\in\mathcal{G}} s_t^{(i)}\varepsilon_t^{(i)}\right\|^2 + 4\mathbb{E}\left\|\mathcal{B}_t^1\right\|^2 \\
&\leq \frac{12c_\lambda m\tilde{\sigma}^2}{t_{\mathcal{G}}} + \frac{4\tilde{\sigma}^2}{t_{\mathcal{G}}} + \frac{(\tau_t^{max}DL)^2(384c_\lambda + 64)}{t^2} \\
&\leq \frac{8\tilde{\sigma}^2}{t} + \frac{24c_\lambda m\tilde{\sigma}^2}{t} + \frac{64(\tau_t^{max}DL)^2}{t^2} + \frac{384c_\lambda(\tau_t^{max}DL)^2}{t^2}\,,
\end{aligned}$$

where the first inequality uses $\|\mathbf{a}+\mathbf{b}\|^2 \leq 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$, which holds $\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^d$. The second inequality utilizes the same inequality and is further supported by Definition 3.1 and Equation (16). The third aligns with Theorem 4.1, and employs the first bias bound in Eq. (14). The last one utilizes the fact that $t_{\mathcal{G}} \geq (1-\lambda)t \geq t/2$, given $\lambda < 1/2$. □

### B.3 Proof of Thm. 4.2

*Proof of Thm. 4.2.* Following Lemma 4.1 and applying Jensen's inequality, we derive the following bound:

$$\mathbb{E}\|\mathcal{E}_t\| = \mathbb{E}\sqrt{\|\mathcal{E}_t\|^2} \leq \sqrt{\mathbb{E}\|\mathcal{E}_t\|^2} \leq O\left(\frac{\tilde{\sigma}\sqrt{1+mc_\lambda}}{\sqrt{t}} + \frac{\tau_t^{max}DL\sqrt{1+c_\lambda}}{t}\right)\,, \tag{17}$$

where the third inequality uses $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for non-negative $a, b \in \mathbb{R}$. The explanation behind this can be seen through the following steps:

$$\left(\sqrt{a} + \sqrt{b}\right)^2 = a + 2\sqrt{ab} + b \geq a + b\,,$$

whereby taking the square root of both sides of this equation, we obtain the desired inequality.

Next, let's revisit the AnyTime guarantee as outlined in Cutkosky [2019] and proceed to delve into the regret analysis of the update rule.

**Theorem B.1** (Rephrased from Theorem 1 in Cutkosky [2019])**.** *Let $f : \mathcal{K} \to \mathbb{R}$ be a convex function with a minimum $\mathbf{x}^* \in \arg\min_{\mathbf{w}\in\mathcal{K}} f(\mathbf{w})$. Also let $\{\alpha_t \geq 0\}_t$, and $\{\mathbf{w}_t \in \mathcal{K}\}_t$, $\{\mathbf{x}_t \in \mathcal{K}\}_t$, such that $\{\mathbf{x}_t\}_t$ is an $\{\alpha_t\}_t$ weighted averaged of $\{\mathbf{w}_t\}_t$, i.e. such that $\mathbf{x}_1 = \mathbf{w}_1$, and for any $t \geq 1$,*

$$\mathbf{x}_{t+1} = \frac{1}{\alpha_{1:t+1}}\sum_{\tau\in[t+1]} \alpha_\tau \mathbf{w}_\tau\,.$$

*Then the following holds for any $t \geq 1$:*

$$\alpha_{1:t}(f(\mathbf{x}_t) - f(\mathbf{x}^*)) \leq \sum_{\tau\in[t]} \alpha_\tau \nabla f(\mathbf{x}_\tau)(\mathbf{w}_\tau - \mathbf{x}^*)\,.$$

**Lemma B.4.** *Let $f : \mathcal{K} \to \mathbb{R}$ be a convex function with a minimum $\mathbf{x}^* \in \arg\min_{\mathbf{w}\in\mathcal{K}} f(\mathbf{w})$, and assume that the assumption in Eq. (1) holds. Also let $\{\alpha_t \geq 0\}_t$, and $\{\mathbf{w}_t \in \mathcal{K}\}_t$. Then, for any $t \geq 1$, an arbitrary vector $\hat{\mathbf{d}}_t \in \mathbb{R}^d$, and the update rule:*

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{K}}\left(\mathbf{w}_t - \eta\alpha_t\hat{\mathbf{d}}_t\right)\,,$$

*we have,*

$$\sum_{\tau=1}^{t} \alpha_\tau\langle\hat{\mathbf{d}}_\tau, \mathbf{w}_{\tau+1} - \mathbf{x}^*\rangle \leq \frac{D^2}{2\eta} - \frac{1}{2\eta}\sum_{\tau=1}^{t}\|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2\,.$$

**Lemma B.5.** *let $f : \mathcal{K} \to \mathbb{R}$ be an L-smooth and convex function, and let $\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathcal{K}} f(\mathbf{x})$, then for any $\mathbf{x} \in \mathbb{R}^d$ we have,*

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

Next, for every iteration $t \leq T$, we define:

$$\hat{\mathbf{d}}_t := \mathcal{A}_\omega(\{\mathbf{d}_t^{(i)}, s_t^{(i)}\}_{i=1}^m)$$
$$\mathcal{E}_t := \hat{\mathbf{d}}_t - \nabla f(\mathbf{x}_t)$$

Thus, combining Theorem B.1 with Lemma B.4, we have that,

$$\begin{aligned}
\alpha_{1:t}(f(\mathbf{x}_t) - f(\mathbf{x}^*)) &\leq \sum_{\tau\in[t]} \alpha_\tau \langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{x}^* \rangle \\
&= \sum_{\tau\in[t]} \alpha_\tau \langle \hat{\mathbf{d}}_\tau, \mathbf{w}_{\tau+1} - \mathbf{x}^* \rangle + \sum_{\tau\in[t]} \alpha_\tau \langle \hat{\mathbf{d}}_\tau, \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle - \sum_{\tau\in[t]} \alpha_\tau \langle \mathcal{E}_\tau, \mathbf{w}_\tau - \mathbf{x}^* \rangle \\
&\leq \frac{D^2}{2\eta} - \frac{1}{2\eta} \sum_{\tau\in[t]} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 + \sum_{\tau\in[t]} \alpha_\tau \langle \hat{\mathbf{d}}_\tau, \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle - \sum_{\tau\in[t]} \alpha_\tau \langle \mathcal{E}_\tau, \mathbf{w}_\tau - \mathbf{x}^* \rangle \\
&= \frac{D^2}{2\eta} - \frac{1}{2\eta} \sum_{\tau\in[t]} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 + \sum_{\tau\in[t]} \alpha_\tau \langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle - \sum_{\tau\in[t]} \alpha_\tau \langle \mathcal{E}_\tau, \mathbf{w}_{\tau+1} - \mathbf{x}^* \rangle \\
&\leq \frac{D^2}{2\eta} \underbrace{- \frac{1}{2\eta} \sum_{\tau\in[t]} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 + \sum_{\tau\in[t]} \alpha_\tau \langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle}_{\text{(A)}} + D \sum_{\tau\in[t]} \alpha_\tau \|\mathcal{E}_\tau\| \,,
\end{aligned} \tag{18}$$

where the first inequality is derived from the Anytime guarantee, as outlined in Theorem B.1. The second inequality follows Lemma B.4. The third inequality is a result of applying the Cauchy-Schwarz inequality and the assumption in Eq. (1).

$$\begin{aligned}
\text{(A)} &:= -\frac{1}{2\eta} \sum_{\tau\in[t]} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 + \sum_{\tau\in[t]} \alpha_\tau \langle \nabla f(\mathbf{x}_\tau), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle \\
&= -\frac{1}{2\eta} \sum_{\tau\in[t]} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 + \sum_{\tau\in[t]} \alpha_\tau \langle \nabla f(\mathbf{x}_\tau) - \nabla f(\mathbf{x}^*), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle + \sum_{\tau\in[t]} \alpha_\tau \langle \nabla f(\mathbf{x}^*), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle \\
&\leq \frac{\eta}{2} \sum_{\tau\in[t]} \alpha_\tau^2 \|\nabla f(\mathbf{x}_\tau) - \nabla f(\mathbf{x}^*)\|^2 + \sum_{\tau\in[t]} \alpha_\tau \langle \nabla f(\mathbf{x}^*), \mathbf{w}_\tau - \mathbf{w}_{\tau+1} \rangle \\
&\leq 2\eta L \sum_{\tau\in[t]} \alpha_{1:\tau} \Delta_\tau + \sum_{\tau\in[t]} (\alpha_\tau - \alpha_{\tau-1}) \langle \nabla f(\mathbf{x}^*), \mathbf{w}_\tau \rangle - \alpha_t \langle \nabla f(\mathbf{x}^*), \mathbf{w}_{t+1} \rangle \\
&= 2\eta L \sum_{\tau\in[t]} \alpha_{1:\tau} \Delta_\tau + \sum_{\tau\in[t]} (\alpha_\tau - \alpha_{\tau-1}) \langle \nabla f(\mathbf{x}^*), \mathbf{w}_\tau - \mathbf{w}_{t+1} \rangle \\
&\leq 2\eta L \sum_{\tau\in[t]} \alpha_{1:\tau} \Delta_\tau + \sum_{\tau\in[t]} (\alpha_\tau - \alpha_{\tau-1}) \|\nabla f(\mathbf{x}^*)\| \|\mathbf{w}_\tau - \mathbf{w}_{t+1}\| \\
&\leq \frac{1}{2T} \sum_{\tau\in[T]} \alpha_{1:\tau} \Delta_\tau + \alpha_t G^* D \,.
\end{aligned}$$

Here, the first inequality employs the Young's inequality. For the second inequality, we introduce the notation $\Delta_t := f(\mathbf{x}_t) - f(\mathbf{x}^*)$, and we follow Lemma B.5, which relates to the smoothness of the function $f$. In this step, we also set $\alpha_0 = 0$ and utilizes the property $\alpha_\tau^2 \leq 2\alpha_{1:\tau}$, given that $\alpha_\tau = \tau$. The third inequality uses the Cauchy-Schwarz inequality. The last inequality follows the assumption in Eq. (1). It uses the fact that $t \leq T$ and $\Delta_t \geq 0, \forall t$. This step also incorporates the choice of an appropriate learning rate parameter $\eta \leq 1/4LT$.

Plugging (A) into Eq. (18), gives us,

$$\alpha_{1:t}\Delta_t \leq \frac{1}{2T}\sum_{\tau\in[T]}\alpha_{1:\tau}\Delta_\tau + \frac{D^2}{2\eta} + \alpha_t G^* D + D\sum_{\tau\in[t]}\alpha_\tau\|\mathcal{E}_\tau\| \,. \tag{19}$$

**Lemma B.6** (Lemma C.2 in Levy [2023]). *let $\{A_t\}_{t\in[T]}, \{B_t\}_{t\in[T]}$ be sequences of non-negative elements, and assume that for any $t \leq T$,*

$$A_t \leq B_T + \frac{1}{2T}\sum_{t\in[T]} A_t \,.$$

*Then the following bound holds,*

$$A_T \leq 2B_T \,.$$

In the next step, let us define two terms: $A_t := \alpha_{1:t}\mathbb{E}\left[f(\mathbf{x}_t) - f(\mathbf{x}^*)\right]$ and $B_t := \frac{D^2}{2\eta} + \alpha_t G^* D + D\sum_{\tau\in[t]}\alpha_\tau\mathbb{E}\|\mathcal{E}_\tau\|$. Note that the series $\{B_t\}_t$ forms a non-decreasing series of non-negative values, implying $B_t \leq B_T$ for any $t \in [T]$. As a result of Eq. (19), we have that $A_t \leq B_T + \frac{1}{2T}\sum_{\tau\in[T]} A_\tau$.

Leveraging Lemma B.6, Eq. (17), and acknowledging that $\alpha_{1:T} = \Theta(T^2)$, as $\alpha_t = t$, it follows that:

$$\begin{aligned}
\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] &\leq \frac{2}{T^2}\mathcal{B}_T \\
&= \frac{D^2}{T^2\eta} + \frac{2G^*D}{T} + \frac{2D}{T^2}\sum_{t\in[T]}\alpha_t\mathbb{E}\|\mathcal{E}\| \\
&\leq O\left(\frac{D^2}{T^2\eta} + \frac{G^*D}{T} + \frac{D}{T^2}\sum_{t\in[T]}\left(\sqrt{t}\tilde{\sigma}\sqrt{1+mc_\lambda} + \tau_t^{max}DL\sqrt{1+c_\lambda}\right)\right) \\
&\leq O\left(\frac{D^2}{T^2\eta} + \frac{G^*D}{T} + \frac{D\tilde{\sigma}\sqrt{1+mc_\lambda}}{\sqrt{T}} + \frac{\mu^{max}D^2L\sqrt{1+c_\lambda}}{T}\right) \,,
\end{aligned}$$

where $\mu^{max} := \frac{1}{T}\sum_{t\in[T]}\tau_t^{max}$. Finally, choosing the optimal $\eta \leq \frac{1}{4TL}$ gives us:

$$\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] \leq O\left(\frac{LD^2\mu^{max}\sqrt{1+c_\lambda}}{T} + \frac{G^*D}{T} + \frac{D\tilde{\sigma}\sqrt{1+mc_\lambda}}{\sqrt{T}}\right) \,.$$

□

### B.3.1 Proof of Lemma B.4

*Proof of Lemma B.4.* The update rule $\mathbf{w}_{\tau+1} = \Pi_{\mathcal{K}}(\mathbf{w}_\tau - \eta\alpha_\tau\hat{\mathbf{d}}_\tau)$ can be expressed as a convex optimization problem within the set $\mathcal{K}$:

$$\begin{aligned}
\mathbf{w}_{\tau+1} &= \Pi_{\mathcal{K}}\left(\mathbf{w}_\tau - \eta\alpha_\tau\hat{\mathbf{d}}_\tau\right) \\
&= \arg\min_{\mathbf{w}\in\mathcal{K}}\|\mathbf{w}_\tau - \eta\alpha_\tau\hat{\mathbf{d}}_\tau - \mathbf{w}\|^2 \\
&= \arg\min_{\mathbf{w}\in\mathcal{K}}\{\alpha_\tau\langle\hat{\mathbf{d}}_\tau, \mathbf{w} - \mathbf{w}_\tau\rangle + \frac{1}{2\eta}\|\mathbf{w} - \mathbf{w}_\tau\|^2\} \,.
\end{aligned}$$

Here, the first equality is derived from the definition of the update rule, the second stems from the property of projection, and the final equality is obtained by reformulating the optimization problem in a way that does not affect the minimum value.

Given that $\mathbf{w}_{\tau+1}$ is the optimal solution of the above convex problem, by the optimality conditions, we have that:

$$\left\langle\alpha_\tau\hat{\mathbf{d}}_\tau + \frac{1}{\eta}(\mathbf{w}_{\tau+1} - \mathbf{w}_\tau), \mathbf{w} - \mathbf{w}_{\tau+1}\right\rangle \geq 0, \quad \forall\mathbf{w}\in\mathcal{K} \,.$$

Rearranging this, summing over $t \geq 1$ iterations, and taking $\mathbf{w} = \mathbf{x}^*$, we derive:

$$\begin{aligned}
\sum_{\tau\in[t]} \alpha_\tau \langle \hat{\mathbf{d}}_\tau, \mathbf{w}_{\tau+1} - \mathbf{x}^* \rangle &\leq \frac{1}{\eta} \sum_{\tau\in[t]} \langle \mathbf{w}_\tau - \mathbf{w}_{\tau+1}, \mathbf{w}_{\tau+1} - \mathbf{x}^* \rangle \\
&= \frac{1}{2\eta} \sum_{\tau\in[t]} \left( \|\mathbf{w}_\tau - \mathbf{x}^*\|^2 - \|\mathbf{w}_{\tau+1} - \mathbf{x}^*\|^2 - \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 \right) \\
&= \frac{1}{2\eta} \left( \|\mathbf{w}_1 - \mathbf{x}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{x}^*\|^2 - \sum_{\tau\in[t]} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 \right) \\
&\leq \frac{D^2}{2\eta} - \frac{1}{2\eta} \sum_{\tau\in[t]} \|\mathbf{w}_\tau - \mathbf{w}_{\tau+1}\|^2 .
\end{aligned}$$

The first equality equality is achieved through algebraic manipulation, and the last inequality follows the assumption in Eq. (1). □

#### B.3.2 Proof of Lemma B.5

*Proof of Lemma B.5.*

**Lemma B.7** (Lemma C.1 in Levy [2023]). *let $f : \mathbb{R}^d \to \mathbb{R}$ be an $L$-smooth function with a global minimum $\mathbf{x}^*$, then for any $\mathbf{x} \in \mathbb{R}^d$ we have,*

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

Let us define the function $h(\mathbf{x}) = f(\mathbf{x}) - f(\mathbf{x}^*) - \langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle$. Due to the convexity of $f(\mathbf{x})$, we have the gradient inequality $f(\mathbf{x}) - f(\mathbf{x}^*) \geq \langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle$, which implies $h(\mathbf{x}) \geq 0$. As $h(\mathbf{x}^*) = 0$, this implies that $\mathbf{x}^*$ is the global minimum of $h$. Applying Lemma B.7, gives us,

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^*)\|^2 = \|\nabla h(\mathbf{x})\|^2 \leq 2L(f(\mathbf{x}) - f(\mathbf{x}^*) - \langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle) \leq 2L(f(\mathbf{x}) - f(\mathbf{x}^*)) ,$$

where last inequality holds due to the convexity of $f$, which implies that $\langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq 0$. □

## C Robust Aggregators Analysis

### C.1 Weighted Robust Aggregators

#### C.1.1 Weighted Geometric Median (WeightedGM)

The Weighted Geometric Median (WeightedGM) is an aggregation method that seeks a point minimizing the weighted sum of Euclidean distances to a set of points. Formally, for a given set of points $\{\mathbf{x}_i\}_{i=1}^m$ and corresponding weights $\{s_i\}_{i=1}^m$, the WeightedGM aggregator is defined as follows:

$$\text{WeightedGM} \in \arg\min_{\mathbf{y}\in\mathbb{R}^d} \sum_{i\in[m]} s_i \|\mathbf{y} - \mathbf{x}_i\|$$

**Lemma C.1.** *Let $\hat{\mathbf{x}}$ be a WeightedGM aggregator then $\hat{\mathbf{x}}$ is $(c_\lambda, \lambda)$-weighted robust with $c_\lambda = \left(1 + \frac{\lambda}{1-2\lambda}\right)^2$.*

*Proof.*

$$
\begin{aligned}
\|\hat{\mathbf{x}} - \bar{\mathbf{x}}_{\mathcal{G}}\| &= \left\| \hat{\mathbf{x}} - \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{G}} s_i \mathbf{x}_i \right\| \\
&\leq \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{G}} s_i \|\hat{\mathbf{x}} - \mathbf{x}_i\| \\
&= \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in[m]} s_i \|\hat{\mathbf{x}} - \mathbf{x}_i\| - \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{B}} s_i \|\hat{\mathbf{x}} - \mathbf{x}_i\| \\
&\leq \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in[m]} s_i \|\bar{\mathbf{x}}_{\mathcal{G}} - \mathbf{x}_i\| - \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{B}} s_i \|\hat{\mathbf{x}} - \mathbf{x}_i\| \\
&= \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{G}} s_i \|\bar{\mathbf{x}}_{\mathcal{G}} - \mathbf{x}_i\| + \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{B}} s_i \|\bar{\mathbf{x}}_{\mathcal{G}} - \mathbf{x}_i\| - \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{B}} s_i \|\hat{\mathbf{x}} - \mathbf{x}_i\| \\
&\leq \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{G}} s_i \|\bar{\mathbf{x}}_{\mathcal{G}} - \mathbf{x}_i\| + \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{B}} s_i \|\bar{\mathbf{x}}_{\mathcal{G}} - \hat{\mathbf{x}}\| \\
&\leq \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{G}} s_i \|\bar{\mathbf{x}}_{\mathcal{G}} - \mathbf{x}_i\| + \frac{\lambda}{1-\lambda} \|\bar{\mathbf{x}}_{\mathcal{G}} - \hat{\mathbf{x}}\| \ .
\end{aligned}
$$

The first inequality leverages Jensen's inequality. The second inequality follows the WeightedGM definition. The third is derived using the following triangle inequality: $\|\bar{\mathbf{x}}_{\mathcal{G}} - \mathbf{x}_i\| \leq \|\bar{\mathbf{x}}_{\mathcal{G}} - \hat{\mathbf{x}}\| + \|\hat{\mathbf{x}} - \mathbf{x}_i\|$. The final inequality is based on the assumptions that $\sum_{i\in\mathcal{B}} s_i \leq \lambda s_{1:m}$ and $\sum_{i\in\mathcal{G}} s_i \geq (1-\lambda) s_{1:m}$.

By rearranging, we obtain:

$$
\|\hat{\mathbf{x}} - \bar{\mathbf{x}}_{\mathcal{G}}\| \leq \left(1 + \frac{\lambda}{1-2\lambda}\right) \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{G}} s_i \|\bar{\mathbf{x}}_{\mathcal{G}} - \mathbf{x}_i\| \ .
$$

Taking the square of both sides and applying Jensen's inequality gives us:

$$
\|\hat{\mathbf{x}} - \bar{\mathbf{x}}_{\mathcal{G}}\|^2 \leq \left(1 + \frac{\lambda}{1-2\lambda}\right)^2 \frac{1}{\sum_{i\in\mathcal{G}} s_i} \sum_{i\in\mathcal{G}} s_i \|\bar{\mathbf{x}}_{\mathcal{G}} - \mathbf{x}_i\|^2 \ .
$$

Taking the exception of both sides gives us the following:

$$
\mathbb{E}\|\hat{\mathbf{x}} - \bar{\mathbf{x}}_{\mathcal{G}}\|^2 \leq \left(1 + \frac{\lambda}{1-2\lambda}\right)^2 \rho^2 \ .
$$

□

### C.1.2 Weighted Coordinate-Wise Median (WeightedCWMed)

The Weighted Coordinate-Wise Median (WeightedCWMed) is an aggregation technique that operates on a per-coordinate basis. For a given set of multi-dimensional data points $\{\mathbf{x}_i\}_{i=1}^m$ and corresponding weights $\{s_i\}_{i=1}^m$, the WeightedCWMed is computed by independently finding the weighted median of each coordinate across all points. Formally, for the $k^{\text{th}}$ dimension, the WeightedCWMed aggregator is defined as:

$$
[\text{WeightedCWMed}]_k := \text{WeightedMedian}(\{[\mathbf{x}_i]_k\}_{i=1}^m; \{[s_i]_k\}_{i=1}^m)
$$

where $[\mathbf{x}]_k$ is the $k^{\text{th}}$ element of a vector $\mathbf{x}$ and the WeightedMedian is defined as follows: given the elements $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ of each dimension are sorted in ascending order and their corresponds weights $\{s_1, \ldots, s_m\}$, the weighted median is the element $\mathbf{x}_{j^*}$, where $j^*$ is determined by the condition:

$$
j^* \in \arg\min_{j\in[m]} \left\{ \sum_{i\in[j]} s_i > \frac{1}{2} \sum_{i\in[m]} s_i \right\}
$$

If there exists a value $j$ such that:

$$\sum_{i\in[j]} s_i = \frac{1}{2}\sum_{i\in[m]} s_i$$

Then, the WeightedMedian is the average of the $j$-th and $(j+1)$-th elements:

$$\text{WeightedMedian} = \frac{\mathbf{x}_j + \mathbf{x}_{j+1}}{2}$$

Here, we extend the theoretical guarantee of the Coordinate-Wise Median (CWMed) to its weighted version, following the procedure in Allouah et al. [2023].

**Lemma C.2.** *Let $A_\omega : \mathbb{R}^{d\times m} \to \mathbb{R}^d$ be a weighted coordinate-wise aggregation function. Given set of points $\{\mathbf{x}_i\}_{i=1}^m$ and corresponding weights $\{s_i\}_{i=1}^m$, this function incorporates $d$ real-valued functions $A_\omega^1, \ldots, A_\omega^d$, where each $[A_\omega(\{\mathbf{x}_i\}_{i=1}^m; \{s_i\}_{i=1}^m)]_k = A_\omega^k(\{[\mathbf{x}_i]_k\}_{i=1}^m; \{[s_i]_k\}_{i=1}^m)$. If for each $k \in [d]$, $A_\omega^k$ is $(c_\lambda, \lambda)$-weighted robust that satisfies:*

$$\mathbb{E}\left|A_\omega^k(\{[\mathbf{x}_i]_k\}_{i=1}^m; \{[s_i]_k\}_{i=1}^m) - [\bar{\mathbf{x}}_\mathcal{G}]_k\right|^2 \leq \frac{c_\lambda}{\sum_{i\in\mathcal{G}} s_i}\sum_{i\in\mathcal{G}} s_i \mathbb{E}\left|[\mathbf{x}_i]_k - [\bar{\mathbf{x}}_\mathcal{G}]_k\right|^2 .$$

*Then $A_\omega$ is $(c_\lambda, \lambda)$-weighted robust.*

*Proof.* Since $A_\omega$ is a coordinate-wise aggregator, it applies the same aggregation rule across each dimension. Therefore,

$$\|A_\omega(\{\mathbf{x}_i\}_{i=1}^m; \{s_i\}_{i=1}^m) - \bar{\mathbf{x}}_\mathcal{G}\|^2 = \sum_{k\in[d]} \left|A_\omega^k(\{[\mathbf{x}_i]_k\}_{i=1}^m; \{[s_i]_k\}_{i=1}^m) - [\bar{\mathbf{x}}_\mathcal{G}]_k\right|^2 .$$

Given that each $A_\omega^k$, for $k \in [d]$, is $(c_\lambda, \lambda)$-weighted robust that satisfies:

$$\mathbb{E}\left|A_\omega^k(\{[\mathbf{x}_i]_k\}_{i=1}^m; \{[s_i]_k\}_{i=1}^m) - [\bar{\mathbf{x}}_\mathcal{G}]_k\right|^2 \leq \frac{c_\lambda}{\sum_{i\in\mathcal{G}} s_i}\sum_{i\in\mathcal{G}} s_i \mathbb{E}\left|[\mathbf{x}_i]_k - [\bar{\mathbf{x}}_\mathcal{G}]_k\right|^2 .$$

We can express the overall aggregation function $A_\omega$ as follows:

$$\begin{aligned}
\sum_{k\in[d]} \mathbb{E}\left|A_\omega^k(\{[\mathbf{x}_i]_k\}_{i=1}^m; \{[s_i]_k\}_{i=1}^m) - [\bar{\mathbf{x}}_\mathcal{G}]_k\right|^2 &\leq \sum_{k\in[d]} \frac{c_\lambda}{\sum_{i\in\mathcal{G}} s_i}\sum_{i\in\mathcal{G}} s_i \mathbb{E}\left|[\mathbf{x}_i]_k - [\bar{\mathbf{x}}_\mathcal{G}]_k\right|^2 \\
&= \frac{c_\lambda}{\sum_{i\in\mathcal{G}} s_i}\sum_{i\in\mathcal{G}} s_i \mathbb{E}\sum_{k\in[d]}\left|[\mathbf{x}_i]_k - [\bar{\mathbf{x}}_\mathcal{G}]_k\right|^2 \\
&= \frac{c_\lambda}{\sum_{i\in\mathcal{G}} s_i}\sum_{i\in\mathcal{G}} s_i \mathbb{E}\left\|\mathbf{x}_i - \bar{\mathbf{x}}_\mathcal{G}\right\|^2 \\
&\leq c_\lambda \rho^2 ,
\end{aligned}$$

where the first inequality is derived from the assumption stated in this lemma. The second aligns with the definition of $(c_\lambda, \lambda)$-weighted robust as detailed in Definition 3.1. □

**Lemma C.3.** *Let $\hat{\mathbf{x}}$ be a WeightedCWMed aggregator then $\hat{\mathbf{x}}$ is $(c_\lambda, \lambda)$-weighted robust with $c_\lambda = \left(1 + \frac{\lambda}{1-2\lambda}\right)^2$.*

*Proof.* In the context of the $k^{\text{th}}$ coordinate, $[\text{WeightedCWMed}]_k$ functions equivalently to WeightedGM for a one-dimensional case. Consequently, each coordinate of the WeightedCWMed aggregator is $(c_\lambda, \lambda)$-weighted robust with $c_\lambda = \left(1 + \frac{\lambda}{1-2\lambda}\right)^2$ as established in Lemma C.1. Furthermore, since the WeightedCWMed functions on a coordinate-wise basis, it follows from Lemma C.2 that the entire WeightedCWMed aggregator is $(c_\lambda, \lambda)$-weighted robust with $c_\lambda = \left(1 + \frac{\lambda}{1-2\lambda}\right)^2$. □

### C.2 Proof of Lemma 3.1

*Proof of Lemma 3.1.* We denote $\mathbf{y}_i := \mathbf{x}_i - \mathbf{x}_0$, $\Sigma_{\mathcal{G}} := \sum_{i\in\mathcal{G}} s_i$, $\Sigma_S := \sum_{i\in S} s_i$, $\Sigma_{\mathcal{B}} := \sum_{i\in\mathcal{B}} s_i$, and $\Sigma_m := \sum_{i\in[m]} s_i$. Recall that $\Sigma_{\mathcal{G}} \geq (1-\lambda)\Sigma_m$, and $\Sigma_S = (1-\lambda)\Sigma_m$ (Alg. 1).

$$\begin{aligned}
\hat{\mathbf{x}} - \bar{\mathbf{x}}_{\mathcal{G}} &= \frac{1}{\Sigma_S}\sum_{i\in S} s_i\mathbf{x}_i - \bar{\mathbf{x}}_{\mathcal{G}} \\
&= \mathbf{x}_0 - \bar{\mathbf{x}}_{\mathcal{G}} + \frac{1}{\Sigma_S}\sum_{i\in S} s_i(\mathbf{x}_i - \mathbf{x}_0) \\
&= -\frac{1}{\Sigma_{\mathcal{G}}}\sum_{i\in\mathcal{G}} s_i(\mathbf{x}_i - \mathbf{x}_0) + \frac{1}{\Sigma_S}\sum_{i\in S} s_i(\mathbf{x}_i - \mathbf{x}_0) \\
&= -\frac{1}{\Sigma_{\mathcal{G}}}\sum_{i\in\mathcal{G}} s_i\mathbf{y}_i + \frac{1}{\Sigma_S}\sum_{i\in S} s_i\mathbf{y}_i \\
&= \left(\frac{1}{\Sigma_S} - \frac{1}{\Sigma_{\mathcal{G}}}\right)\sum_{i\in\mathcal{G}} s_i\mathbf{y}_i - \frac{1}{\Sigma_S}\sum_{i\in\mathcal{G}} s_i\mathbf{y}_i + \frac{1}{\Sigma_S}\sum_{i\in S} s_i\mathbf{y}_i \\
&= \left(\frac{1}{\Sigma_S} - \frac{1}{\Sigma_{\mathcal{G}}}\right)\sum_{i\in\mathcal{G}} s_i\mathbf{y}_i - \frac{1}{\Sigma_S}\sum_{i\in\mathcal{G}\setminus S} s_i\mathbf{y}_i + \frac{1}{\Sigma_S}\sum_{i\in S\setminus\mathcal{G}} s_i\mathbf{y}_i \\
&= \left(\frac{\Sigma_{\mathcal{G}} - \Sigma_S}{\Sigma_S\Sigma_{\mathcal{G}}}\right)\sum_{i\in\mathcal{G}} s_i\mathbf{y}_i - \frac{1}{\Sigma_S}\sum_{i\in\mathcal{G}\setminus S} s_i\mathbf{y}_i + \frac{1}{\Sigma_S}\sum_{i\in S\setminus\mathcal{G}} s_i\mathbf{y}_i.
\end{aligned}$$

Taking the squared norm of both sides, we obtain:

$$\begin{aligned}
\|\hat{\mathbf{x}} - \bar{\mathbf{x}}_{\mathcal{G}}\|^2 &= \left\|\left(\frac{\Sigma_{\mathcal{G}} - \Sigma_S}{\Sigma_S\Sigma_{\mathcal{G}}}\right)\sum_{i\in\mathcal{G}} s_i\mathbf{y}_i - \frac{1}{\Sigma_S}\sum_{i\in\mathcal{G}\setminus S} s_i\mathbf{y}_i + \frac{1}{\Sigma_S}\sum_{i\in S\setminus\mathcal{G}} s_i\mathbf{y}_i\right\|^2 \\
&\leq 3\left\|\left(\frac{\Sigma_{\mathcal{G}} - \Sigma_S}{\Sigma_S\Sigma_{\mathcal{G}}}\right)\sum_{i\in\mathcal{G}} s_i\mathbf{y}_i\right\|^2 + 3\left\|\frac{1}{\Sigma_S}\sum_{i\in\mathcal{G}\setminus S} s_i\mathbf{y}_i\right\|^2 + 3\left\|\frac{1}{\Sigma_S}\sum_{i\in S\setminus\mathcal{G}} s_i\mathbf{y}_i\right\|^2 \\
&\leq 3\Sigma_{\mathcal{G}}\left(\frac{\Sigma_{\mathcal{G}} - \Sigma_S}{\Sigma_S\Sigma_{\mathcal{G}}}\right)^2\sum_{i\in\mathcal{G}} s_i\|\mathbf{y}_i\|^2 + \frac{3\sum_{i\in S\setminus\mathcal{G}} s_i}{\Sigma_S^2}\sum_{i\in S\setminus\mathcal{G}} s_i\|\mathbf{y}_i\|^2 + \frac{3\sum_{i\in\mathcal{G}\setminus S} s_i}{\Sigma_S^2}\sum_{i\in\mathcal{G}\setminus S} s_i\|\mathbf{y}_i\|^2\,,
\end{aligned} \tag{20}$$

where the first inequality follows the inequality $\|\mathbf{a}+\mathbf{b}+\mathbf{c}\|^2 \leq 3\|\mathbf{a}\|^2+3\|\mathbf{b}\|^2+3\|\mathbf{c}\|^2, \forall \mathbf{a},\mathbf{b},\mathbf{c}\in\mathbb{R}^2$ and the second follow Jensen's inequality.

Note that,

$$\begin{aligned}
\left(\frac{\Sigma_{\mathcal{G}} - \Sigma_S}{\Sigma_S\Sigma_{\mathcal{G}}}\right)^2 &= \left(\frac{\Sigma_m - \Sigma_{\mathcal{B}} - (1-\lambda)\Sigma_m}{\Sigma_S\Sigma_{\mathcal{G}}}\right)^2 \\
&= \left(\frac{\lambda\Sigma_m - \Sigma_{\mathcal{B}}}{\Sigma_S\Sigma_{\mathcal{G}}}\right)^2 \\
&\leq \left(\frac{\lambda\Sigma_m}{(1-\lambda)\Sigma_m\Sigma_{\mathcal{G}}}\right)^2 \\
&\leq \left(\frac{2\lambda}{\Sigma_{\mathcal{G}}}\right)^2 \\
&< \frac{2\lambda}{\Sigma_{\mathcal{G}}^2}\,,
\end{aligned} \tag{21}$$

where the first inequality holds because $\Sigma_{\mathcal{B}} \leq \lambda\Sigma_m$ and $\Sigma_S = (1-\lambda)\Sigma_m$. The second inequality follows from the fact that $1-\lambda \geq 1/2$, and the last since $\lambda < 1/2$.

In addition, we have that,

$$\begin{aligned}\frac{\sum_{i\in S\setminus\mathcal{G}} s_i}{\Sigma_S^2} &= \frac{\sum_{i\in S\cup\mathcal{G}} s_i - \sum_{i\in\mathcal{G}} s_i}{\Sigma_S^2} \\ &\leq \frac{\Sigma_m - (1-\lambda)\Sigma_m}{\Sigma_S^2} \\ &= \frac{\lambda\Sigma_m}{\Sigma_S^2} \\ &= \frac{\lambda\Sigma_m}{(1-\lambda)^2\Sigma_m^2} \\ &\leq \frac{4\lambda}{\Sigma_m} \\ &\leq \frac{4\lambda}{\Sigma_\mathcal{G}}\,, \end{aligned} \tag{22}$$

where the first inequality follows the facts that $S\cup\mathcal{G}\subseteq[m]$, $\{s_i\geq 0\}_{i\in[m]}$ and $\Sigma_\mathcal{G}\geq(1-\lambda)\Sigma_m$. The second inequality is based on that $1-\lambda\geq 1/2$, and the last since $\Sigma_\mathcal{G}\leq\Sigma_m$. And in a similar way,

$$\frac{\sum_{i\in\mathcal{G}\setminus S} s_i}{\Sigma_S^2} \leq \frac{4\lambda}{\Sigma_\mathcal{G}} \tag{23}$$

Applying Eq. (21), Eq. (22) and Eq. (23) into Eq. (20), gives us,

$$\begin{aligned}\|\hat{\mathbf{x}}-\bar{\mathbf{x}}_\mathcal{G}\|^2 &\leq \frac{6\lambda}{\Sigma_\mathcal{G}}\sum_{i\in\mathcal{G}} s_i\|\mathbf{y}_i\|^2 + \frac{12\lambda}{\Sigma_\mathcal{G}}\sum_{i\in S\setminus\mathcal{G}} s_i\|\mathbf{y}_i\|^2 + \frac{12\lambda}{\Sigma_\mathcal{G}}\sum_{i\in\mathcal{G}\setminus S} s_i\|\mathbf{y}_i\|^2 \\ &\leq \frac{12\lambda}{\Sigma_\mathcal{G}}\sum_{i\in S\setminus\mathcal{G}} s_i\|\mathbf{x}_i-\mathbf{x}_0\|^2 + \frac{18\lambda}{\Sigma_\mathcal{G}}\sum_{i\in\mathcal{G}} s_i\|\mathbf{x}_i-\mathbf{x}_0\|^2\,,\end{aligned}$$

where the latter holds since $\sum_{i\in\mathcal{G}\setminus S} s_i\|\mathbf{y}_i\|^2 \leq \sum_{i\in\mathcal{G}} s_i\|\mathbf{y}_i\|^2$.

Next, we define:

$$S^* := \bigcup_{i\in S}\{i\}_{j=1}^{s_i}, \quad \mathcal{G}^* := \bigcup_{i\in\mathcal{G}}\{i\}_{j=1}^{s_i}\,.$$

Note that $\sum_{i\in S\setminus\mathcal{G}} s_i\|\mathbf{x}_i-\mathbf{x}_0\|^2 = \sum_{i\in S^*\setminus\mathcal{G}^*}\|\mathbf{x}_i-\mathbf{x}_0\|^2$. We'll show that there exists an injective function $\Phi: S^*\setminus\mathcal{G}^* \to \mathcal{G}^*\setminus S^*$ such that, $\forall i\in S^*\setminus\mathcal{G}^*$, $\|\mathbf{x}_{\Phi(i)}-\mathbf{x}_0\|\geq\|\mathbf{x}_i-\mathbf{x}_0\|$ is satisfied. This follows from our selection of $S$, which consists of the closest elements $\{\mathbf{x}_i\}_{i\in S}$ to $\mathbf{x}_0$ (see Alg. 1), and from:

$$|S^*\setminus\mathcal{G}^*| = \sum_{i\in S\setminus\mathcal{G}} s_i = \sum_{i\in S} s_i + \sum_{i\in\mathcal{G}\setminus S} s_i - \sum_{i\in\mathcal{G}} s_i \leq \sum_{i\in\mathcal{G}\setminus S} s_i = |\mathcal{G}^*\setminus S^*|\,,$$

where the last inequality follows that $\sum_{i\in S} s_i - \sum_{i\in\mathcal{G}} s_i = (1-\lambda)\Sigma_m - \Sigma_\mathcal{G} \leq 0$.

Thus,

$$\begin{aligned}
\|\hat{\mathbf{x}} - \bar{\mathbf{x}}_{\mathcal{G}}\|^2 &\leq \frac{12\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in S \setminus \mathcal{G}} s_i \|\mathbf{x}_i - \mathbf{x}_0\|^2 + \frac{18\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}} s_i \|\mathbf{x}_i - \mathbf{x}_0\|^2 \\
&= \frac{12\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in S^* \setminus \mathcal{G}^*} \|\mathbf{x}_i - \mathbf{x}_0\|^2 + \frac{18\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}} s_i \|\mathbf{x}_i - \mathbf{x}_0\|^2 \\
&\leq \frac{12\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in S^* \setminus \mathcal{G}^*} \|\mathbf{x}_{\Phi(i)} - \mathbf{x}_0\|^2 + \frac{18\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}} s_i \|\mathbf{x}_i - \mathbf{x}_0\|^2 \\
&\leq \frac{12\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}^*} \|\mathbf{x}_i - \mathbf{x}_0\|^2 + \frac{18\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}} s_i \|\mathbf{x}_i - \mathbf{x}_0\|^2 \\
&= \frac{12\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}} s_i \|\mathbf{x}_i - \mathbf{x}_0\|^2 + \frac{18\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}} s_i \|\mathbf{x}_i - \mathbf{x}_0\|^2 \\
&= \frac{30\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}} s_i \|\mathbf{x}_i - \mathbf{x}_0\|^2 \\
&\leq \frac{60\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}} s_i \|\mathbf{x}_i - \bar{\mathbf{x}}_{\mathcal{G}}\|^2 + \frac{60\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}} s_i \|\bar{\mathbf{x}}_{\mathcal{G}} - \mathbf{x}_0\|^2 ,
\end{aligned}$$

where the second inequality follows from the definition of the injective function $\Phi$. The third inequality is justified by the fact that $\sum_{i \in \mathcal{G}^* \setminus S^*} \|\mathbf{y}_i\|^2 \leq \sum_{i \in \mathcal{G}^*} \|\mathbf{y}_i\|^2$. Finally, the last inequality leverages the property $\|\mathbf{a} + \mathbf{b}\|^2 \leq 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$, which holds $\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^d$.

Taking the expectations of both sides gives us the following:

$$\begin{aligned}
\mathbb{E}\|\hat{\mathbf{x}} - \bar{\mathbf{x}}_{\mathcal{G}}\|^2 &\leq \frac{60\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}} s_i \mathbb{E}\|\mathbf{x}_i - \bar{\mathbf{x}}_{\mathcal{G}}\|^2 + \frac{60\lambda}{\Sigma_{\mathcal{G}}} \sum_{i \in \mathcal{G}} s_i \mathbb{E}\|\bar{\mathbf{x}}_{\mathcal{G}} - \mathbf{x}_0\|^2 \\
&\leq 60\lambda\rho^2 + 60\lambda c_\lambda \rho^2 \\
&= 60\lambda(1 + c_\lambda)\rho^2 ,
\end{aligned}$$

where the last inequaility stems from Def. 3.1. □

# D Experiments

## D.1 Technical Details

**Datasets.** We simulated over the MNIST [LeCun et al., 2010] and CIFAR-10 [Krizhevsky et al., 2014] datasets. The datasets were accessed through `torchvision` (version 0.16.2).

- **MNIST Dataset**. MNIST is a widely used benchmark dataset in the machine learning community, consisting of 70,000 grayscale images of handwritten digits (0-9) with a resolution of 28x28 pixels. The dataset is split into 60,000 training images and 10,000 test images.
- **CIFAR-10 Dataset**. CIFAR-10 is a widely recognized benchmark dataset in the machine learning community, containing 60,000 color images categorized into 10 different classes. Each image has a resolution of 32x32 pixels and represents objects such as airplanes, automobiles, birds, cats, and more. The dataset is evenly split into 50,000 training images and 10,000 test images.

**Imbalanced Arrival Scenarios.** We simulated two types of imbalanced arrival scenarios:

- **Proportional Arrival Probability**: The probability of arrival for the $i$-th worker in the honest group was set proportionally to $i / \sum_{j \in \mathcal{G}} j$, ensuring that workers with higher indices had a higher chance of arriving. The same distribution method was applied to Byzantine workers.
- **Squared ID Arrival Probability**: In a more skewed scenario, the arrival probability was proportional to the square of the worker's ID, i.e., $i^2 / \sum_{j \in \mathcal{G}} j^2$. This setup further accentuated the imbalance by favoring workers with larger IDs.

| Parameter | MNIST | CIFAR-10 |
|---|---|---|
| Model Architecture | Conv(1,20,5), ReLU, MaxPool(2x2), Conv(20,50,5), ReLU, MaxPool(2x2), FC(800→50), BatchNorm, ReLU, FC(50→10) | Conv(3,20,5), ReLU, MaxPool(2x2), Conv(20,50,5), ReLU, MaxPool(2x2), FC(1250→50), BatchNorm, ReLU, FC(50→10) |
| Learning Rate | 0.01 | 0.01 |
| Batch Size | 64 | 64 |
| Data Processing & Augmentation | Normalize(mean=(0.1307), std=(0.3081)) | RandomCrop(size=32, padding=2), RandomHorizontalFlip(p=0.5), Normalize(mean=(0.4914, 0.4822, 0.4465), std=(0.2023, 0.1994, 0.2010)) |

Table 2: Experimental Setup for MNIST and CIFAR-10

For simplicity, Byzantine workers were introduced after a fixed number of iterations, controlled by a parameter $\lambda$. However, it is worth noting that when Byzantine iterations are concentrated, they can cause significant performance degradation. Such patterns can lead to increased delays for honest updates, ultimately affecting the overall convergence of the algorithm.

**Optimization Setup.** We optimized the cross-entropy loss across all experiments. For comparisons, we configured $\mu^2$-SGD with fixed parameters $\gamma = 0.1$ and $\beta = 0.25$. This was tested against Standard SGD, and Momentum-based SGD, where the momentum parameter was set to $\beta = 0.9$ as recommended by Karimireddy et al. [2021], and also fine-tuning $\beta$ to 0.8.

**Attack Simulations.** We simulated four types of attacks to evaluate the robustness of our approach:

1. **Label Flipping** [Allen-Zhu et al., 2020]: The labels of the data were flipped to incorrect values, by subtracting the original labels from 9.
2. **Sign Flipping** [Allen-Zhu et al., 2020]: The signs of the workers' output were flipped.
3. **Little** [Baruch et al., 2019]: Adapted from the synchronous case. It computes the maximum allowable deviation $z_{\max}$ based on iterations count rather than the number of workers. Then, it perturbs the honest updates by subtracting the product of the weighted standard deviation and $z_{\max}$ from the weighted mean of the honest updates.

   $$\text{Byzantine\_update} = \text{weighted\_mean}(\text{honest\_momentums}) - \text{weighted\_std}(\text{honest\_momentums}) \cdot z_{\max}.$$
4. **Empire** [Xie et al., 2020a]: Adapted from the synchronous case. This attack scales the weighted mean of the honest momentums by a factor $\epsilon$ in the negative direction,

   $$\text{Byzantine\_update} = -\epsilon \cdot \text{weighted\_mean}(\text{honest\_momentums}).$$

In the two latter attacks, the mean and standard deviation are calculated coordinate-wise with respect to weights, setting $\epsilon = 0.1$.

**AnyTime Update Formulation.** Regarding the AnyTime update, defined as $\mathbf{x}_t := \frac{\alpha_t \mathbf{w}_t + \alpha_{1:t-1}\mathbf{x}_{t-1}}{\alpha_{1:t}}$, we employed a momentum-based formulation that equivalent to the standard AnyTime update. Specifically, we updated the model parameters according to the formula:

$$\mathbf{x}_t = \gamma_t \mathbf{w}_t + (1 - \gamma_t)\mathbf{x}_{t-1}$$

where $\gamma_t$ is difined as $\gamma_t := \frac{\alpha_t}{\alpha_{1:t}}$. By setting $\alpha_t = C\alpha_{1:t-1}$ with $C > 0$ being a constant, we derived that $\gamma_t = \frac{C}{C+1}$ and remains consistent across all time steps $t \geq 1$.

For more details, please visit our GitHub repository.[1]

[1] https://github.com/dahan198/asynchronous-fault-tolerant-ml

## D.2 Experimental Results on MNIST

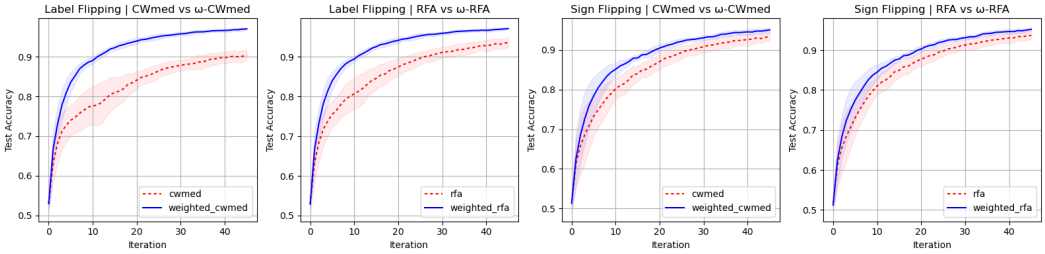


Figure 5: **MNIST**. **Test Accuracy of Weighted vs. Non-Weighted Robust Aggregators**. This scenario involves 17 workers, including 8 Byzantine workers, with workers' arrival probabilities proportional to the square of their IDs. We used the $\mu^2$-SGD in this scenario. Left: *label flipping*, $\lambda = 0.3$. Right: *sign flipping*, $\lambda = 0.4$.

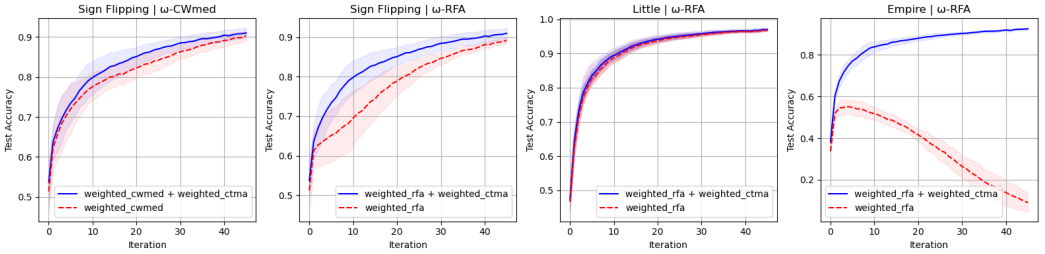


Figure 6: **MNIST**. **Test Accuracy Comparison of Weighted Robust Aggregators With and Without $\omega$-CTMA**. This scenario involves 9 workers, with a very fast Byzantine worker, and workers' arrival probabilities proportional to their IDs. On the *left*, we have a *sign flipping* attack with standard *momentum* ($\beta = 0.9$, $\lambda = 0.4$), and on the *right*, we have *little* ($\lambda = 0.2$) and *empire* ($\lambda = 0.4$) attacks with $\mu^2$-SGD.

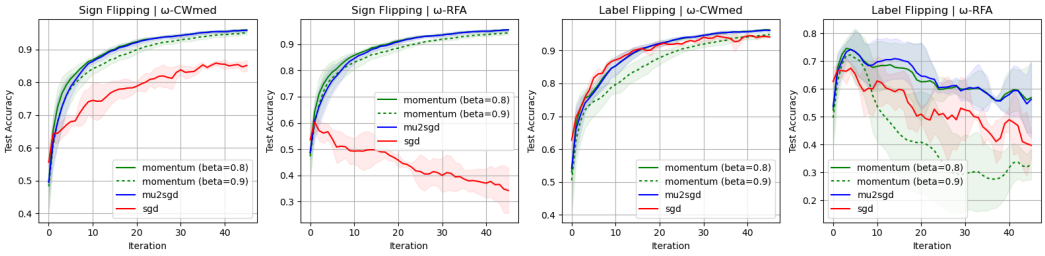


Figure 7: **MNIST**. **Test Accuracy Comparison Among Different Optimizers**. This scenario involves 9 workers, with $\lambda = 0.4$, 4 Byzantine workers, and workers' arrival probabilities proportional to their IDs. We also compared between momentum with the standard parameter $\beta = 0.9$ suggested by Karimireddy et al. [2021] and a fine-tuned parameter $\beta = 0.8$.

## NeurIPS paper checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: we ensured that our paper conforms with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: we do not foresee any special societal impact that arise due to our work

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate

deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: our work does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.