Delayed Momentum Aggregation: Communication-efficient Byzantine-robust Federated Learning with Partial Participation

Kaoru Otsuka KAORU.OTSUKA@OIST.JP
Okinawa Institute of Science and Technology, Japan

Yuki Takezawa YUKI-TAKEZAWA@ML.IST.I.KYOTO-U.AC.JP
Kyoto University, Okinawa Institute of Science and Technology, Japan

Makoto Yamada MAKOTO.YAMADA@OIST.JP
Okinawa Institute of Science and Technology, Japan

Abstract

Federated Learning (FL) allows distributed model training across multiple clients while preserving data privacy, but it remains vulnerable to Byzantine clients that exhibit malicious behavior. While existing Byzantine-robust FL methods provide strong convergence guarantees (e.g., to a stationary point in expectation) under Byzantine attacks, they typically assume full client participation, which is unrealistic due to communication constraints and client availability. Under partial participation, existing methods fail immediately after the sampled clients contain a Byzantine majority, creating a fundamental challenge for sparse communication. First, we introduce *delayed momentum aggregation*, a novel principle where the server aggregates the most recently received momentum from non-participating clients alongside fresh momentum from active clients. Our optimizer *D-Byz-SGDM* (Delayed Byzantine-robust SGD with Momentum) implements this delayed momentum aggregation principle for Byzantine-robust FL with partial participation. Experiments on deep learning tasks validated the proposed method, showing stable and robust training under various Byzantine attacks.

Keywords: Federated Learning, Byzantine-robust Optimization, Communication-efficient Distributed Training

1. Introduction

Federated Learning (FL) enables collaborative training across many clients without centralizing raw data, and has become a standard approach when privacy, bandwidth, or governance constraints prevent data pooling [37, 52]. Its central idea is to transmit gradients rather than raw data. Specifically, each client computes the gradient using their local dataset and sends it to the central server. Then, the central server computes the average of the gradients and updates the parameters. Since its proposal, FL has attracted many optimization researchers and has been widely studied in areas such as communication compression [2, 4, 27, 35, 42, 49, 56, 66], data heterogeneity [3, 16, 34, 39, 48, 62, 67, 68, 73, 76], accelerated methods [21, 33, 36, 45, 50, 57, 58], and Byzantine-robust FL, including defenses for homogeneous data [5, 10, 11, 20, 40, 53, 54, 61, 75] and heterogeneous data [1, 7, 14, 22, 23, 25, 47, 63, 65, 71, 74].

Due to the nature of FL, where a large number of clients participate in the training process, it is vulnerable to clients that behave incorrectly, commonly referred to as Byzantine clients [37, 46]. For instance, some clients may be faulty, while others may act maliciously to disrupt training.

Under Byzantine failures, naive averaging is notoriously brittle: even a single Byzantine client can significantly skew the aggregated model updates. To address this issue, a large body of work has proposed Byzantine-robust FL methods [7, 11, 12, 40], which replace simple averaging with robust aggregation rules at the central server. A robust aggregator guarantees that, as long as the majority of inputs come from honest clients, the aggregation output remains close to the true average of the honest clients' parameters, regardless of the values sent by malicious clients. Thanks to these robust aggregation techniques, Byzantine-robust FL can maintain convergence guarantees, despite the presence of Byzantine clients.

However, most of these existing Byzantine-robust FL methods rely on the assumption that all clients participate in every round, which is unrealistic. Some clients may be temporarily unavailable, for example, due to unreliable connections or competing computational tasks [13, 32, 37, 59, 69, 72]. Even if all clients were available, it is common practice to sample only a subset of the clients to reduce the communication overhead between the central server and the clients [38, 39, 60]. When only a subset of clients participates, most existing Byzantine-robust FL methods fail to remain robust against Byzantine clients. Specifically, in the partial participation setting, the majority of the sampled clients can be malicious. In such a case, a robust aggregator may no longer provide a good estimation of the average of the honest clients' parameters. Only a few papers have studied Byzantine-robust FL with partial participation [8, 51]. Malinovsky et al. [51] proposed a variance reduction-based optimizer with a specialized clipping strategy, showing tolerance even in rounds with a Byzantine majority. However, variance reduction methods perform poorly for deep learning models [24]. Allouah et al. [8] proposed replacing the naive averaging in FedAvg [52] with a Byzantine-robust aggregator. Their algorithm, however, relies on vanilla (non-momentum) SGD, which is vulnerable to time-coupled attacks [9, 40], and it offers no mitigation when Byzantine clients form a majority.

In this paper, we tackle the challenge of Byzantine-robust FL with partial participation, aiming for a solution that is practical and effective under real-world constraints. Our proposed method, *D-Byz-SGDM* (Delayed Byzantine-robust SGD with Momentum), is strikingly simple: at each aggregation step, the central server aggregates not only the gradients sent from the sampled clients but also the most recently received gradients from the non-sampled clients. As a result, this effectively aggregates the entire set of clients, thereby preventing rounds where Byzantine clients dominate the aggregation. Experiments on deep learning tasks show stable and robust training under both partial participation and Byzantine attacks.

We defer a comprehensive discussion of related work to Appendix A and proceed with the formal problem setup.

2. Preliminary

Notations. Our notation largely follows [41, 43]. We denote by n the total number of clients, and for any positive integer k, let $[k] \coloneqq \{1, 2, \dots, k\}$. The set of good (non-Byzantine) clients is represented by $\mathcal{G} \subseteq [n]$ with cardinality $G \coloneqq |\mathcal{G}|$. The Byzantine ratio is defined as $\delta \coloneqq (n-G)/n$, and throughout this paper we assume $\delta < 1/2$. For each client i, let $\mathcal{D}_i$ denote the distribution of local data ξ_i over parameter space Ω_i. The local loss function is given by $f_i : \mathbb{R}^d \to \mathbb{R}$, defined as $f_i(x) \coloneqq \mathbb{E}_{\xi_i}[F_i(x; \xi_i)]$ where $F_i : \mathbb{R}^d \times \Omega_i \to \mathbb{R}$ is the sample loss.

Problem Definition. We formalize the problem as follows: $\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{G} \sum_{i \in \mathcal{G}} f_i(x) \right\}$ where $x \in \mathbb{R}^d$ denotes the model parameters and $\mathcal{D}_i$ represents the dataset distribution of client i. In general, $\mathcal{D}_i \neq \mathcal{D}_j$, reflecting data heterogeneity across clients.

Byzantine-robust Learning under Full-Participation The full participation setting serves as the theoretical foundation for Byzantine-robust federated learning, where the fundamental challenge is designing aggregation mechanisms that maintain convergence guarantees despite adversarial behavior. This setting provides clean theoretical analysis by eliminating client sampling complexities, establishing design principles for robust aggregation rules and performance benchmarks that inform practical algorithm design. The case of full client participation has been extensively studied in the literature [7, 31, 41].

In this setting, robustness is typically achieved by replacing the simple average with a robust aggregation rule. While the precise definition of such aggregators may vary across works, we adopt the following notion from Karimireddy et al. [41] and use it throughout this paper.

Assumption 1 (**(δ, c)-Robust Aggregator [41, 51]**) *Let $\{X_1, X_2, \ldots, X_n\}$ be a set of random vectors. Suppose there exists a "good" subset $\mathcal{G} \subseteq [n]$ of size $G = |\mathcal{G}| > n/2$ such that $\mathbb{E}\|X_i - X_j\|^2 \leq \rho^2, \ \forall i, j \in \mathcal{G}$. Then the output $\hat{X}$ of a Byzantine-robust aggregator* Agg *satisfies $\mathbb{E}\| \operatorname{Agg}(X_1, \ldots, X_n) - \bar{X}\|^2 \leq c\delta\rho^2, $ where $\bar{X} = \frac{1}{G}\sum_{i \in \mathcal{G}} X_i$.*

Importantly, this definition is not merely abstract. Karimireddy et al. [41] prove (in Theorem 1) that well-known aggregation rules such as KRUM [11], RFA [61], and the coordinate-wise median, when combined with their proposed *bucketing* technique, indeed satisfy Assumption 1. Thus, concrete and practical instantiations of robust aggregators are available within this framework. In addition, momentum-based or variance reduction-based techniques [31, 64] are necessary to achieve robustness against sophisticated attacks. Without such techniques, Karimireddy et al. [40] showed a fundamental lower bound demonstrating that learning fails when stochastic gradient noise is not properly controlled, making these methods essential for countering time-coupled attacks [9].

Federated Learning with Partial Participation Federated learning with partial participation is a fundamental characteristic of practical federated learning systems. Real-world deployments inherently involve clients with heterogeneous capabilities and intermittent availability due to device constraints, battery limitations, and network connectivity variations [37, 52]. This participation pattern directly impacts communication efficiency and system scalability, making it a critical consideration for algorithm design.

In the usual partial participation setting, all clients are assumed to be non-Byzantine, i.e., $\mathcal{G} = [n]$. The classical FEDAVG algorithm [52] samples a subset of active clients, denoted by $\mathcal{S}_t \subseteq [n]$, uniformly at random at each round t, and aggregates their local updates by naive averaging: $\frac{1}{|\mathcal{S}_t|}\sum_{i \in \mathcal{S}_t} g_i^t$, where g_i^t denotes the local gradient estimator of client i (e.g., a stochastic gradient).

Failure of Byzantine-robust Learning with Partial Participation A natural extension of the full participation setting is to replace the naive averaging step

$$\frac{1}{|\mathcal{S}_t|}\sum_{i \in \mathcal{S}_t} g_i^t \quad \longrightarrow \quad \operatorname{Agg}(\{g_i^t\}_{i \in \mathcal{S}_t}).$$

While appealing, **this strategy fails with partial participation**: in some rounds, the sampled set may contain a Byzantine majority, despite the global condition $\delta < 1/2$. In such cases, no robust aggregator can reliably distinguish adversarial from honest updates. The likelihood of such Byzantine-majority rounds grows with time.

Recent work has sought to address this issue. Allouah et al. [8] provided lower bounds on the subsample size. However, due to a lack of momentum or variance reduction, their method collapses under time-coupled attacks such as ALIE [9]. Malinovsky et al. [51] established convergence guarantees tolerating Byzantine-majority rounds via gradient-difference clipping, but their analysis relies on variance reduction-based optimizers, which are known to be ineffective in deep learning [24].

3. Proposed Method

In this section, we propose **delayed momentum aggregation**, which is to apply the robust aggregator not only to the momentum of sampled clients but also to the cached momentum of non-sampled clients. Then, we propose a delayed momentum aggregation-based optimizer D-Byz-SGDM, which is Byzantine-robust even if only a subset of clients participate in each round. Formally, let x^t denote the global model parameter maintained by the server at round t. The server then updates it using delayed momentum aggregation as follows:

$$x^t = x^{t-1} - \eta \ \mathrm{Agg}\left(\{m_i^t\}_{i \in \mathcal{S}_t} \cup \{m_i^{t-\tau(i,t)}\}_{i \in [n] \setminus \mathcal{S}_t}\right), \qquad \text{(delayed momentum aggregation)}$$

where each m_i^t represents a local momentum estimate, and $\tau(i,t)$ denotes the (possibly stochastic) delay since client i's last update was received. This design maintains that $\mathrm{Agg}(\cdot)$ consistently sees the global Byzantine fraction $\delta<1/2$, ensuring robustness even with partial participation.

As a concrete special case of the main idea, we propose a new method, D-Byz-SGDM, whose full update rule appears in Algorithm 1 in Appendix B. In each round t, the server independently samples each client with probability p (i.e., $z^t \sim \mathrm{Ber}(p)^{\otimes n}$ and $\mathcal{S}_t = \{i : z_i^t = 1\}$). The selected clients refresh their momentum, while non-selected clients retain their cached value:

$$m_i^t = \begin{cases} (1-\alpha) m_i^{t-1} + \alpha \nabla f_i(x^{t-1}, \xi_i^{t-1}), & i \in \mathcal{S}_t, \\ m_i^{t-1}, & i \notin \mathcal{S}_t, \end{cases}$$

where $\alpha \in (0,1]$ is the client momentum parameter. Note that each client i is included in $\mathcal{S}_t$ with probability p. Importantly, D-Byz-SGDM introduces no extra communication overhead. The server simply maintains one vector m_i^t per client while reusing cached momentum for non-sampled clients, resulting in a memory requirement matching the full participation setting.

4. Experiments

We evaluated D-Byz-SGDM under various Byzantine attacks with partial participation ($p = 0.5$) by training a convolutional network on MNIST and a ResNet-18 on CIFAR-10 across IID and non-IID data partitions. We compared four optimizers (FedAvg, FedAvgM, D-Byz-SGDM, and the heuristic momentum extension of Byz-VR-MARINA-PP from Malinovsky et al. [51]) with five robust aggregators under six Byzantine attacks. FedAvg [52] performed single-step SGD per client followed by server-side aggregation, while FedAvgM [16] extended this with client-side momentum ($\beta = 0.9$). In our setting, the standard averaging step in four optimizers was replaced by robust aggregation rules, allowing us to assess performance under Byzantine attacks. Our implementation extended Karimireddy et al. [41]'s codebase[1] with attacks from the ByzFL framework [30] and

1. https://github.com/epfml/byzantine-robust-noniid-optimizer

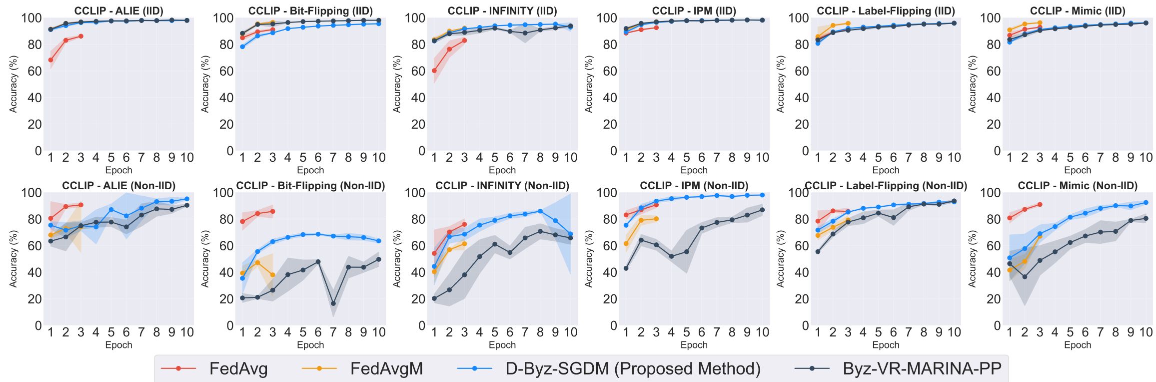

Figure 1: MNIST training dynamics under centered clipping (CCLIP) across six Byzantine attacks. D-Byz-SGDM remained stable and achieved the highest accuracy, while FedAvg/FedAvgM diverged when a Byzantine majority was sampled.

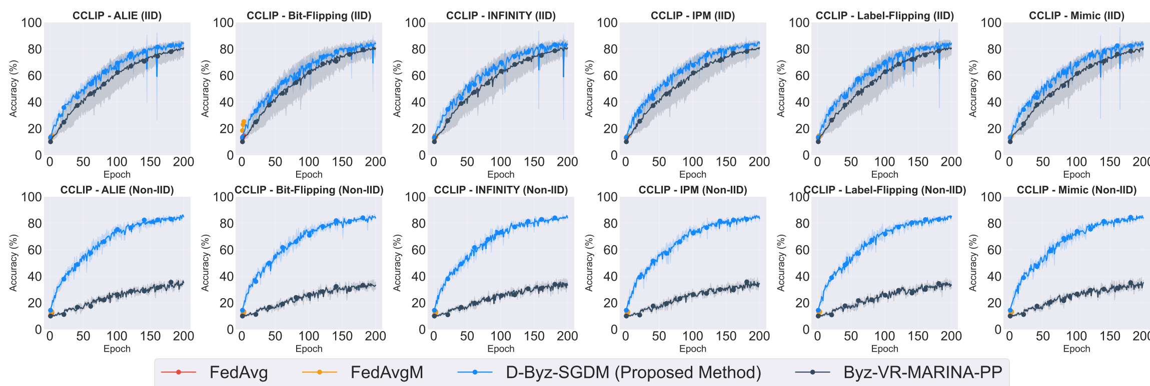

Figure 2: CIFAR-10 (ResNet-18) training dynamics under centered clipping (CCLIP) across six Byzantine attacks. D-Byz-SGDM remained stable and achieved the highest accuracy, whereas FedAvg/FedAvgM collapsed rapidly once a Byzantine majority was sampled and stalled as early as epoch 4.

additional support for CIFAR-10/ResNet-18 training. Appendix C provided complete experimental details.

Hyperparameter selection. For each optimizer (FedAvg, FedAvgM, D-Byz-SGDM) we tuned a global learning rate η over the grid $\{0.1, 0.01, 0.001\}$. Byz-VR-MARINA-PP required tuning both η and the clipping radius $\lambda \in \{10.0, 1.0, 0.1\}$. Every configuration was evaluated over seeds $\{0, 1, 2\}$, and we selected the setting with the highest mean validation accuracy for reporting in both the non-Byzantine and Byzantine settings.

4.1. Byzantine Robustness with Partial Participation (Main Result)

We analyzed partial participation ($p = 0.5$) with $n = 25$ total clients of which 20% were Byzantine ($\delta = 0.2$). All plots in this subsection used centered clipping (`CCLIP`) [40] as the server-side aggregator.

Key findings. Figures 1 and 2 demonstrate the performance of algorithms with the `CCLIP` aggregator under Byzantine attacks with partial participation ($p = 0.5$). Our experiments reveal three critical insights: (1) *D-Byz-SGDM consistently achieved the highest final accuracy across all settings.* On MNIST IID (upper half of Fig. 1), both D-Byz-SGDM and Byz-VR-MARINA-PP achieved near-perfect accuracy, while FedAvg and FedAvgM diverged after three epochs. On CIFAR-10 with ResNet-18 (upper half of Fig. 2), D-Byz-SGDM sustained 80–85% accuracy across all attack types. (2) *Non-IID data exposed critical algorithmic differences.* On non-IID MNIST (lower half of Fig. 1), Byz-VR-MARINA-PP exhibited high variance and unstable convergence, while D-Byz-SGDM maintained consistent performance. The disparity was dramatic on non-IID CIFAR-10 (lower half of Fig. 2): Byz-VR-MARINA-PP catastrophically failed (20–35% accuracy), whereas D-Byz-SGDM maintained 80–85% accuracy. The delayed momentum aggregation principle proved crucial. While standard methods failed when a Byzantine majority was sampled,[2] D-Byz-SGDM maintained stable convergence. (3) *The approach generalizes across aggregators.* Similar trends held across other aggregators (`avg`, `krum`, `cm`, `rfa`) and both datasets, with FedAvg and FedAvgM performing poorly in both IID and non-IID settings (FedAvgM showed marginal improvements only in specific attacks like Bit-Flipping); see Appendix D for the full set of figures.

4.2. Baseline Performance without Byzantine Clients

We also examined the non-Byzantine setting ($\delta = 0$) to establish baseline performance. The setup used $n = 20$ clients with the `avg` aggregator. Detailed figures and discussion are deferred to Appendix C (Baseline Performance Evaluation), where Figs. 3 and 4 present the full results.

5. Conclusion

We proposed *delayed momentum aggregation*, a principle where servers aggregate fresh gradients from participating clients with the most recently received momentum from non-participating clients. Our D-Byz-SGDM optimizer delivers Byzantine-robust training under partial participation, and experiments show consistent improvements over existing methods across various attacks and data distributions. The delayed momentum aggregation principle opens promising avenues for extension to other client selection schemes [15, 17, 28, 29, 48] beyond Bernoulli sampling. While the empirical evidence is encouraging, a complete theoretical analysis of convergence remains open and will be pursued in future work.

Acknowledgement

Makoto Yamada was partly supported by JSPS KAKENHI Grant Number 24K03004 and by JST ASPIRE JPMJAP2302. Yuki Takezawa was supported by JSPS KAKENHI Grant Number 23KJ1336.

References

[1] 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*, 2022.

2. With $p = 0.5$, if many Byzantines were sampled together, they could overwhelm the aggregation.

[2] Alyazeed Albasyoni, Mher Safaryan, Laurent Condat, and Peter Richtárik. Optimal gradient compression for distributed and federated learning. *ArXiv preprint*, abs/2010.03246, 2020.

[3] Sulaiman A. Alghunaim. Local exact-diffusion for decentralized optimization and learning. *IEEE Transactions on Automatic Control*, 69(11):7371–7386, 2024.

[4] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, 2017.

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

[6] Zeyuan Allen-Zhu, Faeze Ebrahimianghazani, Jerry Li, and Dan Alistarh. Byzantine-resilient non-convex stochastic gradient descent. In *International Conference on Learning Representations*, 2021.

[7] Youssef Allouah, Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, and John Stephan. Fixing by mixing: A recipe for optimal byzantine ML under heterogeneity. In *International Conference on Artificial Intelligence and Statistics*, 2023.

[8] Youssef Allouah, Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, Geovani Rizk, and Sasha Voitovych. Byzantine-robust federated learning: Impact of client subsampling and local updates. In *International Conference on Machine Learning*, 2024.

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

[10] Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd with majority vote is communication efficient and fault tolerant. In *International Conference on Learning Representations*, 2019.

[11] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, 2017.

[12] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, 2017.

[13] Kallista A. Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[14] Lingjiao Chen, Hongyi Wang, Zachary B. Charles, and Dimitris S. Papailiopoulos. DRACO: byzantine-resilient distributed training via redundant gradients. In *International Conference on Machine Learning*, 2018.

[15] Wenlin Chen, Samuel Horváth, and Peter Richtárik. Optimal client sampling for federated learning. *Trans. Mach. Learn. Res.*, 2022.

[16] Ziheng Cheng, Xinmeng Huang, Pengfei Wu, and Kun Yuan. Momentum benefits non-iid federated learning simply and provably. In *International Conference on Learning Representations*, 2024.

[17] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *ArXiv preprint*, abs/2010.01243, 2020.

[18] Tehila Dahan and Kfir Y. Levy. Weight for robustness: A comprehensive approach towards optimal fault-tolerant asynchronous ML. In *Advances in Neural Information Processing Systems*, 2024.

[19] Tehila Dahan and Kfir Yehuda Levy. Fault tolerant ML: efficient meta-aggregation and synchronous training. In *International Conference on Machine Learning*, 2024.

[20] Georgios Damaskinos, El-Mahdi El-Mhamdi, Rachid Guerraoui, Arsany Guirguis, and Sébastien Rouault. AGGREGATHOR: byzantine machine learning via robust gradient aggregation. In *Proceedings of Machine Learning and Systems*, 2019.

[21] Alexandre d'Aspremont, Damien Scieur, and Adrien B. Taylor. Acceleration methods. *Found. Trends Optim.*, 5(1-2):1–245, 2021.

[22] Deepesh Data and Suhas N. Diggavi. Byzantine-resilient SGD in high dimensions on heterogeneous data. In *IEEE International Symposium on Information Theory*, 2021.

[23] Deepesh Data and Suhas N. Diggavi. Byzantine-resilient high-dimensional SGD with local iterations on heterogeneous data. In *International Conference on Machine Learning*, 2021.

[24] Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. In *Advances in Neural Information Processing Systems*, 2019.

[25] El-Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyên Hoang, and Sébastien Rouault. Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). In *Advances in Neural Information Processing Systems*, 2021.

[26] 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*, 2022.

[27] Ilyas Fatkhullin, Alexander Tyurin, and Peter Richtárik. Momentum provably improves error feedback! In *Advances in Neural Information Processing Systems*, 2023.

[28] Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In *International Conference on Machine Learning*, 2021.

[29] Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. A general theory for client sampling in federated learning. In *International Workshop on Trustworthy Federated Learning*. Springer, 2022.

[30] Marc González, Rachid Guerraoui, Rafael Pinot, Geovani Rizk, John Stephan, and François Taïani. Byzfl: Research framework for robust federated learning, 2025.

[31] Eduard Gorbunov, Samuel Horváth, Peter Richtárik, and Gauthier Gidel. Variance reduction is an antidote to byzantines: Better rates, weaker assumptions and communication compression as a cherry on the top. In *International Conference on Learning Representations*, 2023.

[32] Xinran Gu, Kaixuan Huang, Jingzhao Zhang, and Longbo Huang. Fast federated learning in the presence of arbitrary device unavailability. In *Advances in Neural Information Processing Systems*, 2021.

[33] Osman Güler. New proximal point algorithms for convex minimization. *SIAM J. Optim.*, 2(4): 649–664, 1992.

[34] Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*, 2021.

[35] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Peter Richtárik, and Sebastian U. Stich. Stochastic distributed learning with gradient quantization and double-variance reduction. *Optim. Methods Softw.*, 38(1):91–106, 2023.

[36] Xiaowen Jiang, Anton Rodomanov, and Sebastian U. Stich. Stabilized proximal-point methods for federated optimization. In *Advances in Neural Information Processing Systems*, 2024.

[37] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021.

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

[39] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 2020.

[40] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *International Conference on Machine Learning*, 2021.

[41] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. In *International Conference on Learning Representations*, 2022.

[42] Sarit Khirirat, Hamid Reza Feyzmahdavian, and Mikael Johansson. Distributed learning with compressed gradients. *ArXiv preprint*, abs/1806.06573, 2018.

[43] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized SGD with changing topology and local updates. In *International Conference on Machine Learning*, 2020.

[44] Anastasia Koloskova, Sebastian U. Stich, and Martin Jaggi. Sharper convergence guarantees for asynchronous SGD for distributed and federated learning. In *Advances in Neural Information Processing Systems*, 2022.

[45] Dmitry Kovalev, Aleksandr Beznosikov, Ekaterina Borodich, Alexander V. Gasnikov, and Gesualdo Scutari. Optimal gradient sliding and its application to optimal distributed optimization under similarity. In *Advances in Neural Information Processing Systems*, 2022.

[46] Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. 2019.

[47] Liping Li, Wei Xu, Tianyi Chen, Georgios B. Giannakis, and Qing Ling. RSA: byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *AAAI Conference on Artificial Intelligence*, 2019.

[48] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, 2020.

[49] Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International Conference on Machine Learning*, 2021.

[50] Hongzhou Lin, Julien Mairal, and Zaïd Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, 2015.

[51] Grigory Malinovsky, Peter Richtárik, Samuel Horváth, and Eduard Gorbunov. Byzantine robustness and partial participation can be achieved at once: Just clip gradient differences. In *Advances in Neural Information Processing Systems*, 2024.

[52] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 2017.

[53] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, 2018.

[54] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. Distributed momentum for byzantine-resilient stochastic gradient descent. In *International Conference on Learning Representations*, 2021.

[55] Konstantin Mishchenko, Francis R. Bach, Mathieu Even, and Blake E. Woodworth. Asynchronous SGD beats minibatch SGD under arbitrary delays. In *Advances in Neural Information Processing Systems*, 2022.

[56] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *Optimization Methods and Software*, pages 1–16, 2024.

[57] Renato D. C. Monteiro and Benar Fux Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM J. Optim.*, 23(2):1092–1125, 2013.

[58] Yurii Nesterov. *Lectures on convex optimization*. Springer, 2018.

[59] Chaoyue Niu, Fan Wu, Shaojie Tang, Lifeng Hua, Rongfei Jia, Chengfei Lv, Zhihua Wu, and Guihai Chen. Billion-scale federated learning on mobile clients: a submodel design with tunable privacy. In *Annual International Conference on Mobile Computing and Networking*, 2020.

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

[61] Krishna Pillutla, Sham M. Kakade, and Zaïd Harchaoui. Robust aggregation for federated learning. *IEEE Trans. Signal Process.*, 70:1142–1154, 2022.

[62] Shi Pu and Angelia Nedić. Distributed stochastic gradient tracking methods. *Mathematical Programming*, 187(1):409–457, 2021.

[63] Shashank Rajput, Hongyi Wang, Zachary B. Charles, and Dimitris S. Papailiopoulos. DETOX: A redundancy-based framework for faster and more robust gradient aggregation. In *Advances in Neural Information Processing Systems*, 2019.

[64] Ahmad Rammal, Kaja Gruntkowska, Nikita Fedin, Eduard Gorbunov, and Peter Richtárik. Communication compression for byzantine robust learning: New efficient algorithms and improved rates. In *International Conference on Artificial Intelligence and Statistics*, 2024.

[65] Felix Sattler, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. On the byzantine robustness of clustered federated learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.

[66] Sebastian U. Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified SGD with memory. In *Advances in Neural Information Processing Systems*, 2018.

[67] Yuki Takezawa, Han Bao, Kenta Niwa, Ryoma Sato, and Makoto Yamada. Momentum tracking: Momentum acceleration for decentralized deep learning on heterogeneous data. *Transactions on Machine Learning*, 2022.

[68] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, 2020.

[69] Shiqiang Wang and Mingyue Ji. A unified analysis of federated learning with arbitrary client participation. In *Advances in Neural Information Processing Systems*, 2022.

[70] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant SGD by inner product manipulation. In *Uncertainty in Artificial Intelligence*, 2019.

[71] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International Conference on Machine Learning*, 2019.

[72] Yikai Yan, Chaoyue Niu, Yucheng Ding, Zhenzhe Zheng, Shaojie Tang, Qinya Li, Fan Wu, Chengfei Lyu, Yanghe Feng, and Guihai Chen. Federated optimization under intermittent client availability. *INFORMS Journal on Computing*, 36(1):185–202, 2024.

[73] Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. In *International Conference on Learning Representations*, 2021.

[74] Yi-Rui Yang and Wu-Jun Li. BASGD: buffered asynchronous SGD for byzantine learning. In *International Conference on Machine Learning*, 2021.

[75] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, 2018.

[76] Xinwei Zhang, Mingyi Hong, Sairaj V. Dhople, Wotao Yin, and Yang Liu. Fedpd: A federated learning framework with adaptivity to non-iid data. *IEEE Transactions on Signal Processing*, 69:6055–6070, 2021.

[77] Banghua Zhu, Lun Wang, Qi Pang, Shuai Wang, Jiantao Jiao, Dawn Song, and Michael I. Jordan. Byzantine-robust federated learning with optimal statistical rates. In *International Conference on Artificial Intelligence and Statistics*, 2023.

Appendix A. Related Work

Byzantine-robust FL under full participation. Classical defenses replace naive averaging by robust aggregation rules such as Krum [11], coordinate-wise median and trimmed-mean [12], and geometric–median–based RFA [61]; meta-rules like Bulyan further reduce adversarial leverage [53]. Yet these per-round defenses can be vulnerable to time-coupled attacks that inject small, undetectable biases which accumulate across rounds [9, 70]. A key development is to leverage history: Karimireddy et al. [40] formalize such time-coupled failures and prove that momentum (together with robust aggregation) provably restores convergence; subsequent works refine the momentum view and resilient averaging [26]. Heterogeneity (non-IID client data) exacerbates the problem: bucketing [41] and nearest-neighbor mixing (NNM) [7] are pre-aggregation mechanisms that systematically adapt IID-optimal rules (e.g., Krum, median, RFA) to the heterogeneous regime, closing gaps between achievable rates and lower bounds. Beyond aggregation, algorithmic alternatives include coding-theoretic redundancy (DRACO) [14] and filtering for non-convex objectives [5, 6]. Complementing these meta-aggregation approaches that assume full participation, Dahan and Levy [19] propose an efficient *Centered Trimmed Meta-Aggregator* (CTMA) that upgrades base robust aggregators to order-optimal performance at near-averaging cost, and couples it with a double-momentum estimator to obtain convex SCO guarantees in synchronous (full-participation) settings.

Partial participation, and local updates. Partial participation makes robustness strictly harder because the sampled set occasionally contains a Byzantine majority. Early theory coupling Byzantine robustness with local steps shows that convergence can be ensured only when the sampled cohort has a sufficiently large honest fraction at each synchronization—e.g., $\varepsilon \leq 1/3$ corrupted among the K active clients [23, Thm. 1], an assumption strained by client sampling. The interaction between client sampling, multiple local steps, and robust aggregation has since been analyzed in detail by Allouah et al. [8], who quantifies how client sampling reshapes the effective number of Byzantine clients and shows regimes where standard robust aggregators suffice; however, these schemes omit momentum and do not mitigate time-coupled drift. The concurrent line on variance reduction shows another path: by coupling robust aggregation with gradient-difference clipping and periodic anchor steps, Malinovsky et al. [51] proves tolerance even when a sampled round is entirely Byzantine, at the cost of periodic heavier steps. From a statistical-efficiency angle, protocols with near-optimal rates under full participation have been derived via modern robust statistics [77], and recent work explores communication compression jointly with robustness [31, 64].

Asynchrony, delayed gradients, and relevance to our staleness mechanism. Analysis of asynchronous SGD (ASGD) formalizes *delayed/stale* gradients and shows that delays can be controlled via delay-aware stepsizes [44, 55]. In the *Byzantine asynchronous* regime, recent work Dahan and Levy [18] develops a *weighted* robust-aggregation framework and, combined with a double-momentum estimator, proves optimal convergence in the smooth *convex homogeneous* (i.i.d.) setting [18]. Importantly for assumptions, Dahan and Levy [18, 19]'s analysis (both asynchronous and synchronous) operates over a *compact* feasible set (bounded diameter), which is stricter than the bounded-gradient conditions commonly adopted in FL theory.

Our setting is not asynchronous; nevertheless, partial participation induces *server-side staleness* because non-sampled clients contribute historical (per-client) gradients. This places our analysis close to the ASGD toolbox while tackling a distinct failure mode (occasional Byzantine-majority samples under subsampling) without trusted validation data. Technically, we leverage *per-client*

Algorithm 1: Optimizer with delayed momentum aggregation: D-Byz-SGDM

Require: initial vectors x^0, m^0, stepsize η, momentum parameter α, robust aggregator Agg, client sampling probability $p \in (0, 1]$
Initialize m_i^0 and $\tau(i, 0) \leftarrow 0$ for all $i \in [n]$;
for $t = 1, 2, \ldots$ **do**
  Sample $\mathcal{S}_t \subseteq [n]$ by including each $i \in [n]$ independently with prob. p;
  Server broadcasts x^{t-1} to all $i \in \mathcal{S}_t$;
  foreach $i \in \mathcal{S}_t$ ***in parallel*** **do**
    Draw $\xi_i^{t-1} \sim \mathcal{D}_i$ and compute $m_i^t \leftarrow (1-\alpha)m_i^{t-1} + \alpha \nabla F_i(x^{t-1}; \xi_i^{t-1})$;
    Send m_i^t to server;
  end
  foreach $i \notin \mathcal{S}_t$ *(on server)* **do**
    Update $m_i^t \leftarrow m_i^{t-1}$;
  end
  $m^t \leftarrow \mathrm{Agg}\Big(\{m_i^t\}_{i \in \mathcal{S}_t} \cup \{m_i^t\}_{i \notin \mathcal{S}_t}\Big)$ `// delayed momentum aggregation`
  $x^t \leftarrow x^{t-1} - \eta\, m^t$;
end

stale gradients to preserve a history-coupled (global) momentum across rounds, complementing weighted robust aggregation in the asynchronous literature [18].

Relative to prior momentum-based defenses [26, 40] and heterogeneity fixes [7, 41], we study the regime where clients refresh stochastically and adversaries can transiently comprise the sampled majority. Compared to variance reduction-based approaches [51], our method avoids periodic full/anchor gradient computations.

Appendix B. Algorithm Details

We present the detailed algorithm for D-Byz-SGDM (Delayed Byzantine-robust SGD with Momentum), which implements our delayed momentum aggregation principle. The key idea is to apply the robust aggregator not only to the momentum of sampled clients but also to the cached momentum of non-sampled clients, ensuring that the aggregator consistently sees the global Byzantine fraction $\delta < 1/2$ even under partial participation.

In each round t, the server independently samples each client with probability p (i.e., $z^t \sim \mathrm{Ber}(p)^{\otimes n}$ and $\mathcal{S}_t = \{i : z_i^t = 1\}$). The selected clients refresh their momentum using:

$$m_i^t = \begin{cases} (1-\alpha)m_i^{t-1} + \alpha \nabla f_i(x^{t-1}, \xi_i^{t-1}), & i \in \mathcal{S}_t, \\ m_i^{t-1}, & i \notin \mathcal{S}_t, \end{cases}$$

where $\alpha \in (0, 1]$ is the client momentum parameter. Non-selected clients retain their cached momentum values from previous rounds.

The server then performs delayed momentum aggregation by applying the robust aggregator Agg to the union of fresh momentum from sampled clients and cached momentum from non-sampled clients:

$$m^t = \text{Agg}\Big(\{m_i^t\}_{i\in\mathcal{S}_t} \;\cup\; \{m_i^t\}_{i\notin\mathcal{S}_t}\Big)$$

This design ensures that even when partial participation might lead to a Byzantine majority among sampled clients, the aggregator always operates on the full set of clients (fresh and cached), maintaining robustness.

To see how this corresponds to the delayed momentum aggregation principle, note that the delay function $\tau(i,t)$ represents the number of rounds since client i's momentum was last updated. Formally:

$$\tau(i,t) = \min\{s \geq 0 : i \in \mathcal{S}_{t-s}\}$$

This is a random variable that depends on the sampling history. When $i \in \mathcal{S}_t$, we have $\tau(i,t) = 0$ (fresh update), and when $i \notin \mathcal{S}_t$, we have $\tau(i,t) > 0$ (stale update). The algorithm effectively implements:

$$x^t = x^{t-1} - \eta \text{ Agg}\left(\{m_i^t\}_{i\in\mathcal{S}_t} \cup \{m_i^{t-\tau(i,t)}\}_{i\in[n]\setminus\mathcal{S}_t}\right)$$

where for non-sampled clients, $m_i^{t-\tau(i,t)}$ is their most recent momentum update, which is exactly what we store as m_i^t in the algorithm.

Importantly, D-Byz-SGDM does not incur additional communication costs compared to standard partial participation methods: the server only queries sampled clients and stores one momentum vector m_i^t per client, matching the memory requirements of full participation settings.

Appendix C. Additional Experimental Details

C.1. Common Experimental Settings

All experiments covered two vision workloads: MNIST with a convolutional neural network architecture (CONV-CONV-DROPOUT-FC-DROPOUT-FC) and CIFAR-10 with a standard ResNet-18. Training employed cross-entropy (negative log-likelihood) loss with batch size 32 per client and client participation probability $p = 0.5$. We evaluated both IID and non-IID data partitions, with the latter following the class-based approach of Karimireddy et al. [41]. Four optimizers were compared: FedAvg, FedAvgM, D-Byz-SGDM, and the heuristic momentum extension of Byz-VR-MARINA-PP (with $\lambda \in \{10.0, 1.0, 0.1\}$) introduced in [51], all using momentum parameter $\alpha = 0.9$ where applicable. Training ran for 10 epochs (300 iterations total) for MNIST and 200 epochs for CIFAR-10, with results averaged over seeds $\{0, 1, 2\}$. For each optimizer we tuned the learning rate $\eta \in \{0.1, 0.01, 0.001\}$; additionally Byz-VR-MARINA-PP tuned the clipping radius $\lambda \in \{10.0, 1.0, 0.1\}$. We selected the configuration with the highest mean validation accuracy across the three seeds for both the non-Byzantine and Byzantine experiments. Tables 1–4 provided complete configuration details.

C.2. Baseline Performance Evaluation

This experiment established baseline performance under partial participation without Byzantine clients across both MNIST (ConvNet) and CIFAR-10 (ResNet-18). We used $n = 20$ clients with no Byzantine clients ($\delta = 0$) and naive averaging aggregation. The objective was to validate that D-Byz-SGDM maintains competitive performance in non-Byzantine settings and to establish reference performance levels for subsequent robustness comparisons. Results in Figs. 3 and 4

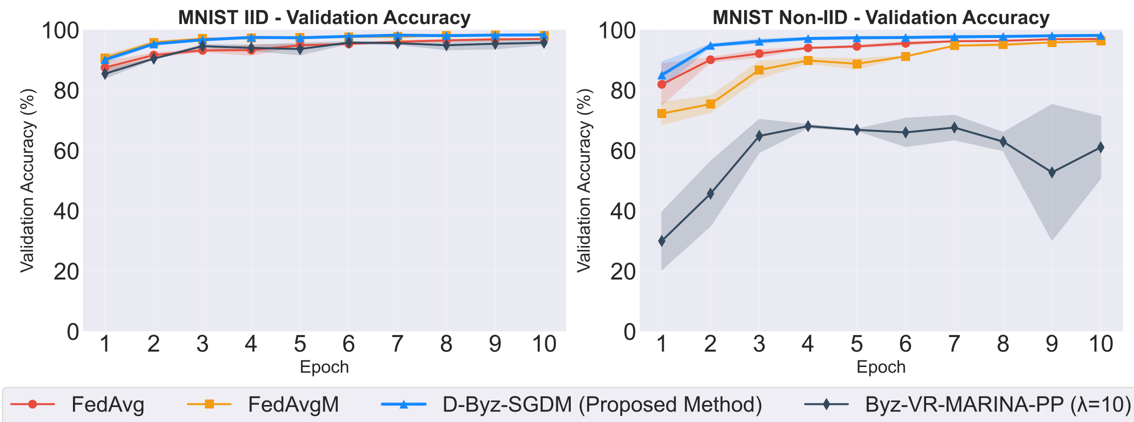

Figure 3: MNIST (non-Byzantine) training dynamics across optimizers. All methods reached near-saturated IID accuracy, yet D-Byz-SGDM retained a clear margin in the non-IID split, indicating that delayed momentum aggregation mitigated heterogeneity-induced drift even without Byzantine clients.

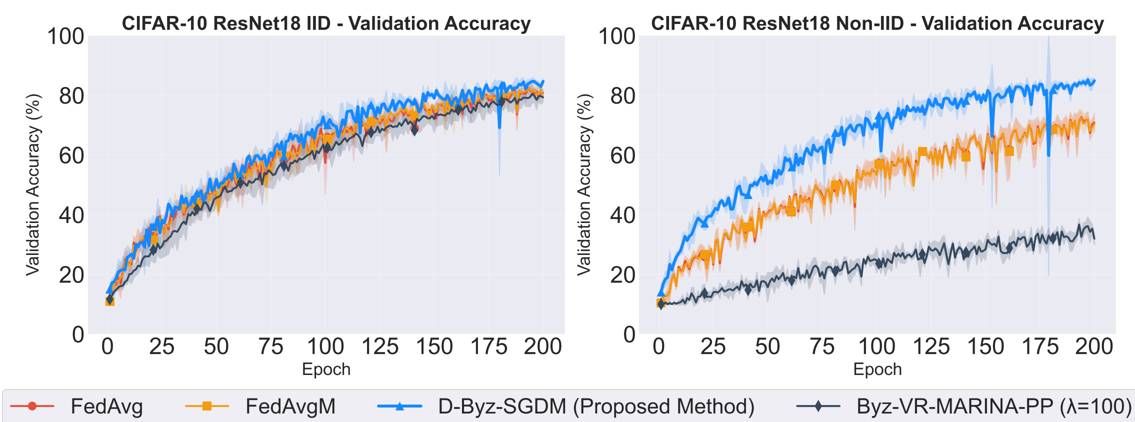

Figure 4: CIFAR-10 (ResNet-18, non-Byzantine) training dynamics across optimizers. D-Byz-SGDM converged faster and finished 5–10 points higher than momentum baselines on both IID and non-IID partitions, whereas Byz-VR-MARINA-PP remained far below the other methods throughout training.

demonstrated that D-Byz-SGDM outperformed standard momentum methods on both MNIST and CIFAR-10 even without adversaries, suggesting that delayed momentum aggregation provided implicit regularization benefits under heterogeneous data distributions.

C.3. Byzantine Robustness Assessment

This experiment evaluated robustness against Byzantine attacks under partial participation on both datasets (MNIST with the ConvNet backbone and CIFAR-10 with ResNet-18). We configured $n = 25$ clients with 5 Byzantine clients (20%). Five robust aggregators were evaluated: Krum, coordinate-wise median, CCLIP (centered clipping), RFA, and naive averaging as baseline. The experimental design included both IID and non-IID data partitions, with bucketing applied in the

Table 1: MNIST (non-Byzantine) configuration used in Fig. 3.

Dataset	MNIST (IID and non-IID partitions)
Model	CONV-CONV-DROPOUT-FC-DROPOUT-FC
Clients	$n = 20$ (all honest)
Participation	$p = 0.5$ (partial participation)
Aggregator	`avg`
Batch size	32 per client
Training horizon	10 epochs (300 rounds)
Optimizers	FedAvg, FedAvgM, D-Byz-SGDM, Byz-VR-MARINA-PP
Learning-rate tuning	grid search on $\{0.1, 0.01, 0.001\}$
Byz-VR-MARINA-PP tuning	joint grid search $\eta \in \{0.1, 0.01, 0.001\}$, $\lambda \in \{10.0, 1.0, 0.1\}$
Seeds	$\{0, 1, 2\}$
Attacks	none

Table 2: MNIST (Byzantine) configuration used in Fig. 1.

Dataset	MNIST (IID and non-IID with bucketing $s = 2$)
Model	CONV-CONV-DROPOUT-FC-DROPOUT-FC
Clients	$n = 25$ (20 honest, 5 Byzantine; $\delta = 0.2$)
Participation	$p = 0.5$ (partial participation)
Aggregators	`avg`, `krum`, `cm`, `CCLIP`, `rfa`
Batch size	32 per client
Training horizon	10 epochs (300 rounds)
Attacks	BF, LF, mimic, IPM, ALIE, INF
Optimizers	FedAvg, FedAvgM, D-Byz-SGDM, Byz-VR-MARINA-PP
Learning-rate tuning	grid search on $\{0.1, 0.01, 0.001\}$
Byz-VR-MARINA-PP tuning	joint grid search $\eta \in \{0.1, 0.01, 0.001\}$, $\lambda \in \{10.0, 1.0, 0.1\}$
Seeds	$\{0, 1, 2\}$

Notation: `avg`=naive average, `krum`=Krum [11], `cm`=coordinate-wise median, `CCLIP`=centered clipping [40], `rfa`=geometric median (RFA) [61].

Byzantine non-IID setting to mitigate extreme heterogeneity. This comprehensive evaluation spanned 6,480 total experimental runs across all combinations of attacks, aggregators, optimizers, data partitions, and random seeds (3,240 runs per dataset).

C.4. Non-IID data partition

We constructed the non-IID split following Karimireddy et al. [41] in the *balanced* case: (i) sorted the training sets by label; (ii) split it into G equal, contiguous shards (where G is the number of good/honest clients); (iii) assigned one shard to each honest client and shuffle examples within each client. We partitioned the test set analogously.

C.5. Computing Environment

Experiments ran on NVIDIA A100-SXM4-80GB GPUs (CUDA 12.2) and AMD EPYC 7763 CPUs. Table 5 provides detailed hardware and software specifications.

Table 3: CIFAR-10 (non-Byzantine) configuration used in Fig. 4.

Dataset	CIFAR-10 (IID and non-IID partitions)
Model	ResNet-18
Clients	$n = 20$ (all honest)
Participation	$p = 0.5$ (partial participation)
Aggregator	`avg`
Batch size	32 per client
Training horizon	200 epochs
Optimizers	FedAvg, FedAvgM, D-Byz-SGDM, Byz-VR-MARINA-PP
Learning-rate tuning	grid search on $\{0.1, 0.01, 0.001\}$
Byz-VR-MARINA-PP tuning	joint grid search $\eta \in \{0.1, 0.01, 0.001\}$, $\lambda \in \{10.0, 1.0, 0.1\}$
Seeds	$\{0, 1, 2\}$
Attacks	none

Table 4: CIFAR-10 (Byzantine) configuration used in Fig. 2.

Dataset	CIFAR-10 (IID and non-IID with bucketing $s = 2$)
Model	ResNet-18
Clients	$n = 25$ (20 honest, 5 Byzantine; $\delta = 0.2$)
Participation	$p = 0.5$ (partial participation)
Aggregators	`avg`, `krum`, `cm`, `CCLIP`, `rfa`
Batch size	32 per client
Training horizon	200 epochs
Attacks	BF, LF, mimic, IPM, ALIE, INF
Optimizers	FedAvg, FedAvgM, D-Byz-SGDM, Byz-VR-MARINA-PP
Learning-rate tuning	grid search on $\{0.1, 0.01, 0.001\}$
Byz-VR-MARINA-PP tuning	joint grid search $\eta \in \{0.1, 0.01, 0.001\}$, $\lambda \in \{10.0, 1.0, 0.1\}$
Seeds	$\{0, 1, 2\}$

Notation: `avg`=naive average, `krum`=Krum [11], `cm`=coordinate-wise median, `CCLIP`=centered clipping [40], `rfa`=geometric median (RFA) [61].

Appendix D. Extended Results

Per-aggregator curves with Byzantine clients. This section complemented Figs. 1 and 2 by showing training dynamics for the other robust aggregators across the same attacks, data partitions, and optimizers on MNIST (ConvNet) and CIFAR-10 (ResNet-18).

Table 5: Runtime hardware and software.

CPU	
Model name	AMD EPYC 7763 64-Core Processor
# CPU(s)	128
GPU	
Product Name	NVIDIA A100-SXM4-80GB
CUDA Version	12.2
PyTorch	
Version	2.7.1

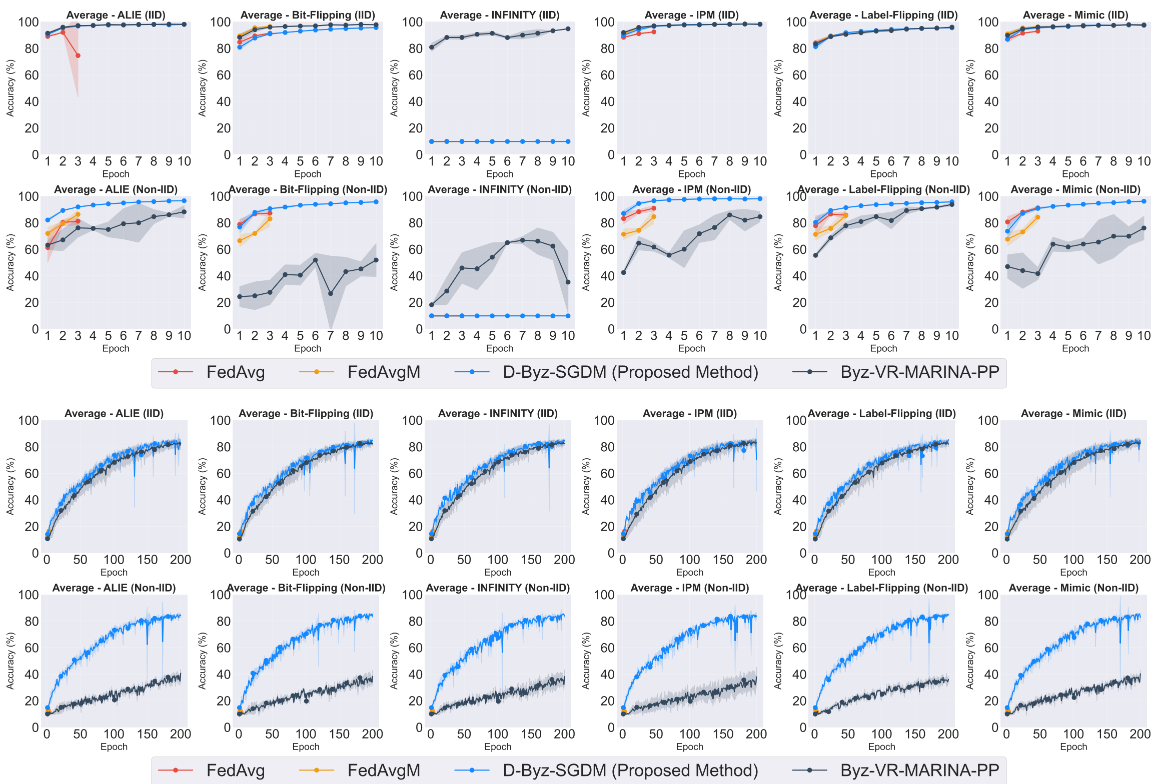

Figure 5: `avg` (naive average) under Byzantine attacks with partial participation. Top: MNIST. Bottom: CIFAR-10.

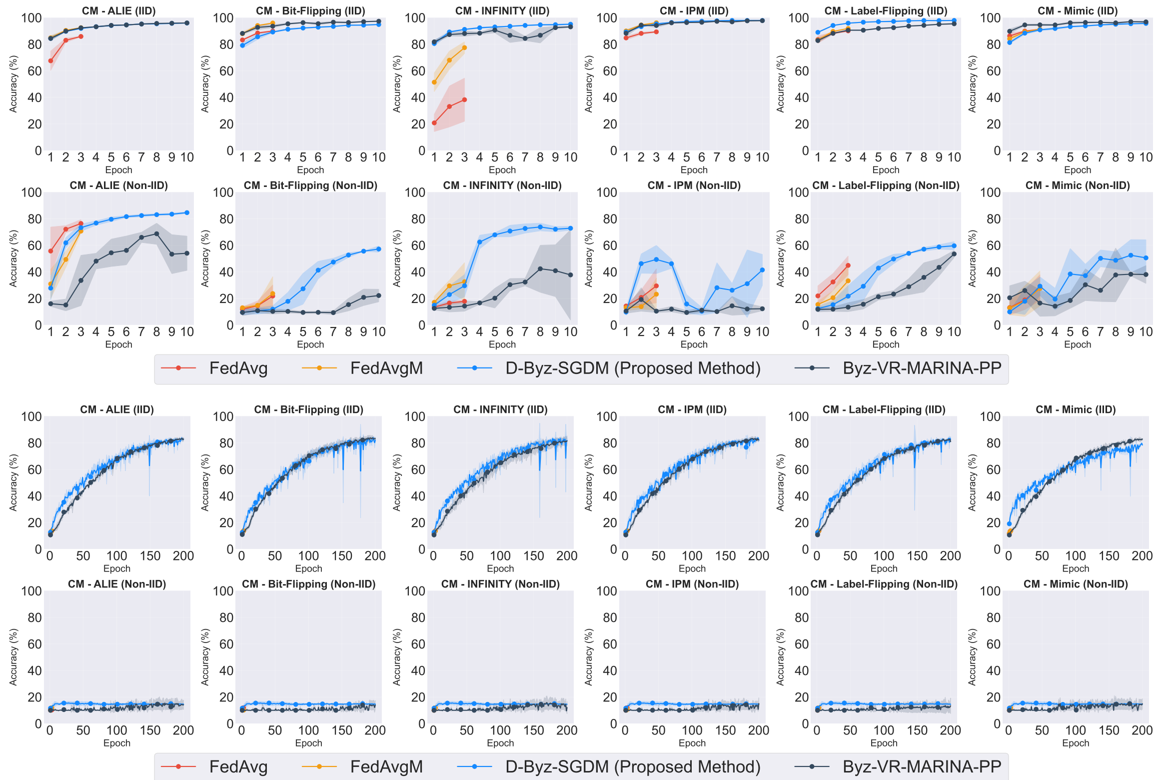

Figure 6: `cm` (coordinate-wise median) under Byzantine attacks with partial participation. Top: MNIST. Bottom: CIFAR-10.

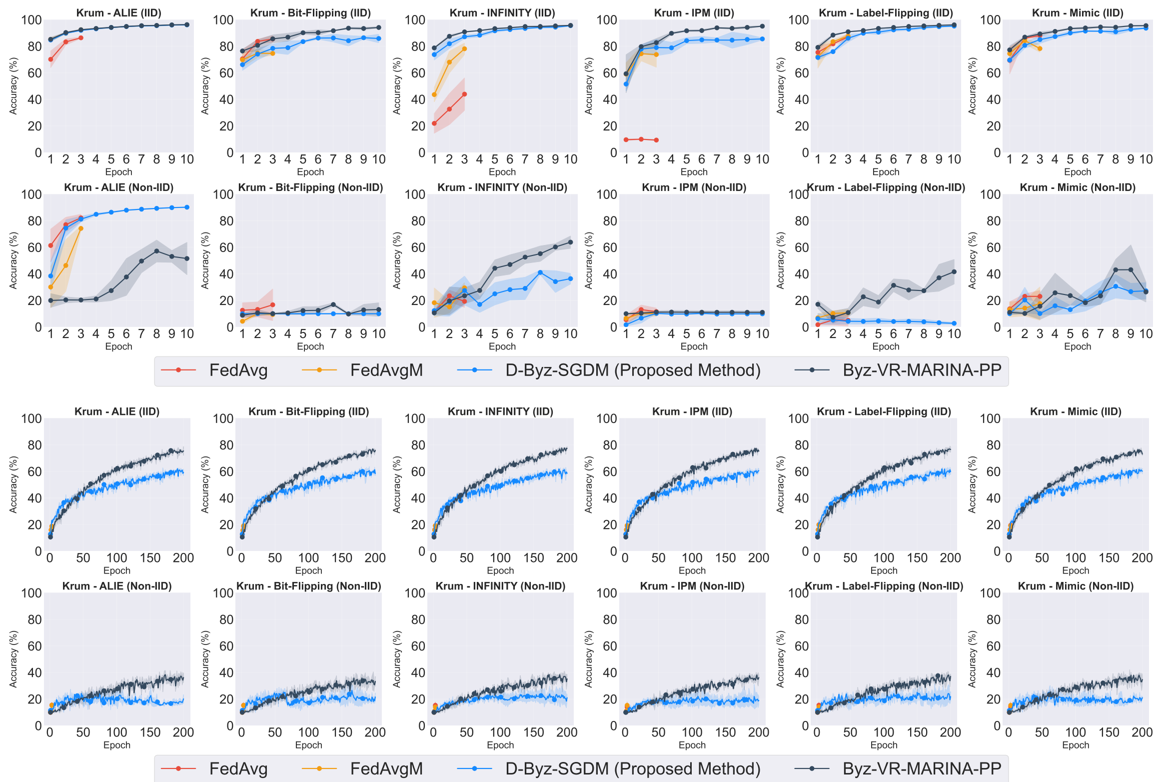

Figure 7: `krum` / Multi-Krum under Byzantine attacks with partial participation. Top: MNIST. Bottom: CIFAR-10.

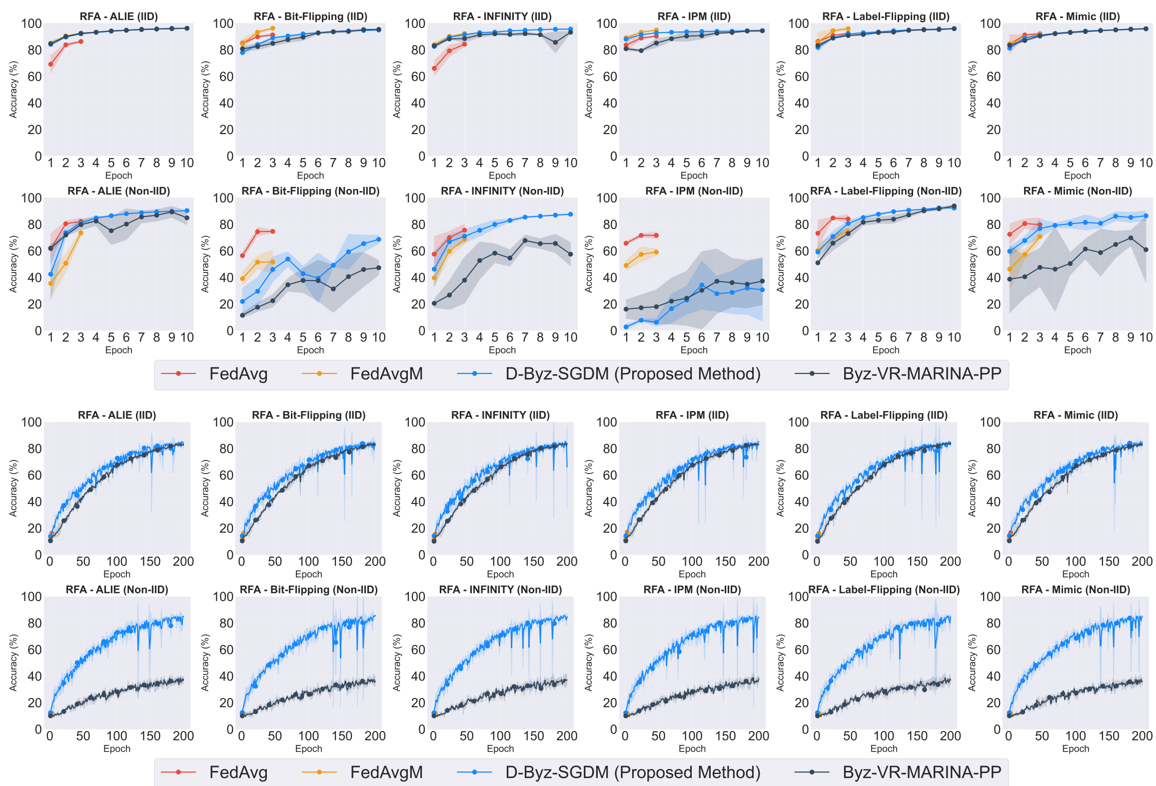

Figure 8: `rfa` (Robust Federated Averaging) under Byzantine attacks with partial participation. Top: MNIST. Bottom: CIFAR-10.