Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Quorum-Bounded Asynchronous Federated Learning under Non-IID Data and Adversarial Clients: A Systems Study of Stale-Update Exclusion and Convergence

Md Shahanur Islam Shagor

smshagor.ru@gmail.com

Voronezh State University of Forestry and Technologies https://orcid.org/0009-0003-7730-3202

Research Article

Keywords: federated learning, asynchronous optimization, non-IID data, Dirichlet partitioning, Byzantine clients, stragglers

Posted Date: September 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10975675/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Abstract

Federated learning systems deployed across heterogeneous edge clients must cope simultaneously with non-identically distributed data, variable response times, and malicious participation. Most analyses isolate these effects, while practical coordinators must decide when to advance a global model despite unfinished workers and how to prevent late results from contaminating later rounds. This study evaluates a quorum-bounded asynchronous federated learning design in which persistent workers train independently, a round finalizes after a configurable response quorum, and results arriving after that round are excluded rather than reused as stale updates. Data heterogeneity is controlled with class-wise Dirichlet partitions, and adversarial clients apply scaled sign-flipping to model updates. A five-seed, 25-round controlled study spans IID data, Dirichlet alpha values of 0.5 and 0.2, response quorums of 20/20, 16/20 and 12/20, and malicious fractions of 0 and 0.2. Reducing the quorum to 12/20 lowers simulated cumulative straggler-dominated latency by approximately 72% relative to full synchronization. The final-accuracy difference between 12/20 and 20/20 is not statistically significant in any partition regime. In contrast, malicious participation increasingly damages convergence as heterogeneity strengthens: the final accuracy penalty is about 1.6–2.2 percentage points for IID data, 8.2–8.4 points at alpha = 0.5, and 10.3–11.2 points at alpha = 0.2. The results show that quorum asynchrony can remove substantial waiting time without measurable accuracy loss in this controlled setting, but data heterogeneity amplifies adversarial risk and remains the dominant challenge.

1 Introduction

Federated learning (FL) moves model optimization to the data rather than moving raw data to a central training service. This architectural shift is attractive for privacy-sensitive and edge-distributed applications, but it replaces the relatively stable execution environment of centralized training with a population of clients that differ in data distribution, compute capacity, network quality, and trustworthiness. The original FedAvg formulation demonstrated that local computation can reduce communication while producing competitive models, yet its round-based synchronization assumes that enough selected clients finish before the server proceeds [1]. In large or heterogeneous deployments, the slowest clients can dominate round time and turn synchronization into a systems bottleneck.

Asynchronous federated optimization was introduced to weaken this barrier by allowing the server to update without waiting for every worker [2]. Subsequent work has explored staleness-aware weighting [3], buffered aggregation [4], and system-aware variants that trade model freshness against update throughput. These methods make clear that asynchrony is not simply a scheduling optimization. A client may train on an older global state, and a server that accepts late updates may combine gradients derived from different model versions. The resulting staleness can bias optimization, interact with data heterogeneity, and complicate privacy or robust aggregation mechanisms.

A second difficulty is statistical heterogeneity. Edge-client data are rarely independent and identically distributed. Differences in user behavior, geography, sensing conditions, class prevalence, or acquisition

devices induce local objectives that can point in different directions. FedProx [5], SCAFFOLD [6], FedNova [7] and a large body of subsequent work address client drift and optimization inconsistency, while empirical studies commonly use Dirichlet allocation to produce controlled label skew. Recent generalization analysis further emphasizes that heterogeneity affects not only optimization speed but also the stability and generalization behavior of federated algorithms [8].

A third difficulty is adversarial participation. FL clients can be compromised or deliberately malicious and may submit arbitrary updates. Robust aggregation has therefore been studied using Krum, trimmed mean, median and more recent heterogeneous-data defenses [9–13]. Importantly, heterogeneity and adversarial behavior are not independent. Honest updates naturally become more diverse when client data are skewed, which reduces the statistical separation between benign and malicious behavior. Recent analyses show that client subsampling and local steps can also change the effective Byzantine fraction and the difficulty of robust learning [11]. Semi-asynchronous settings create an additional selection effect because fast responders, rather than a uniform client sample, may disproportionately determine which updates reach the aggregation quorum.

This paper studies these three pressures together from a systems perspective. The focus is a quorum-bounded asynchronous coordinator with persistent workers. Each round broadcasts one global model version to selected workers, workers train independently with heterogeneous compute and network delays, and the coordinator finalizes the round when a minimum number of successful results has arrived. Crucially, a late result from an already finalized round is drained and excluded; it is not reinterpreted as an update for a newer model. This design removes model-version mixing at the price of discarding some completed work. It therefore occupies a specific point in the asynchronous design space: the system tolerates stragglers through early quorum completion while bounding semantic staleness by rejecting obsolete round results.

The study asks three research questions. RQ1 asks whether reducing the completion quorum materially lowers straggler-dominated training latency. RQ2 asks whether that latency reduction degrades convergence under IID and progressively stronger Dirichlet non-IID partitions. RQ3 asks how 20% malicious participation changes the relationship between quorum size, data heterogeneity and final accuracy. We evaluate these questions with five independent seeds, 25 rounds, 20 clients, three partition regimes, three quorum levels, and both benign and adversarial populations. Median aggregation is held fixed in the primary experiment so that the paper isolates asynchronous scheduling and heterogeneity rather than comparing robust aggregators, which is a separate problem.

The principal finding is a separation between systems cost and statistical risk. A 12-of-20 quorum reduces simulated cumulative latency by roughly 72% while showing no statistically significant final-accuracy difference from full synchronization in any tested partition regime. However, adversarial damage increases sharply with heterogeneity. Under IID data the malicious penalty remains around two percentage points; under Dirichlet alpha = 0.2 it grows to roughly eleven points. The implication is that a coordinator can aggressively mitigate stragglers without necessarily paying an accuracy penalty,

provided that stale updates are excluded, but the same system still requires explicit controls for the interaction between non-IID drift and malicious participation.

2 Related work

Synchronous FL and statistical heterogeneity. FedAvg remains the canonical baseline for communication-efficient federated optimization [1]. Li et al. introduced FedProx to stabilize learning under systems and statistical heterogeneity by adding a proximal term to local objectives [5]. SCAFFOLD uses control variates to correct client drift [6], while FedNova normalizes local updates to address objective inconsistency created by variable local computation [7]. Surveys by Kairouz et al. and Li et al. organize the broader FL design space and identify non-IID data, partial participation, systems heterogeneity and security as central unresolved concerns [14, 15]. Recent stability-based analysis confirms that heterogeneity materially affects generalization behavior even when optimization algorithms converge [8].

Asynchronous FL. Xie et al. proposed asynchronous federated optimization and analyzed convergence in the presence of stale client updates [2]. FedSA subsequently introduced explicit staleness-aware weighting for non-IID clients [3]. FedBuff buffers asynchronous updates before aggregation, showing that buffered asynchrony can provide substantial efficiency gains while preserving compatibility with secure aggregation and differential privacy [4]. More recent work formulates the design as an explicit trade-off between model-parameter staleness and update frequency [16]. Application-specific asynchronous systems likewise report advantages under heterogeneous delays [17]. The present study differs by choosing a strict stale-exclusion policy: updates are accepted only for their originating round and are discarded after the round finalizes. This eliminates cross-version mixing but introduces a response-selection effect governed by the quorum.

Adversarial and Byzantine FL. Krum established a distance-based route to Byzantine-tolerant distributed gradient descent [9]. Coordinate-wise robust statistics and later FL-specific defenses extended this line of work. Allouah et al. showed that client subsampling and local updates materially affect Byzantine-robust FL and that careless subsampling can increase the effective adversarial fraction [11]. BOBA addresses label skew explicitly [12], while self-driven entropy aggregation targets Byzantine robustness under heterogeneous private data [13]. Recent work on asynchronous Byzantine FL demonstrates that the timing model itself must be considered in the fault-tolerance design [18], and PoiSAFL shows that semi-asynchronous defenses can be targeted by scalable poisoning strategies [19]. These results motivate treating scheduling, heterogeneity and adversarial participation as a coupled systems problem rather than three independent layers.

Research gap. Existing studies provide sophisticated algorithms for stale-update weighting, drift correction, or Byzantine filtering, but comparatively less attention has been paid to a simpler coordinator policy that advances on quorum and refuses to reuse obsolete results. Such a policy is attractive because its semantics are easy to audit: an update belongs to exactly one model version. Its cost is

wasted work and potentially biased participation. This paper quantifies the latency-accuracy trade-off of that policy under controlled Dirichlet heterogeneity and malicious participation.

3 System model and problem formulation

3.1 Federated objective and heterogeneous clients

Let there be N clients. Client i owns local data D_i and minimizes a local empirical objective F_i(w). The global objective is F(w) = the weighted sum over clients of p_i F_i(w), where p_i is proportional to the client sample count in the fully weighted case. Under IID allocation, local empirical distributions approximate the population distribution. Under a Dirichlet label split with concentration alpha, smaller alpha concentrates classes among fewer clients and increases the divergence among local objectives. In the experiments we use IID allocation and two non-IID levels, alpha = 0.5 and alpha = 0.2.

3.2 Quorum-bounded asynchronous rounds

At round t, the coordinator broadcasts the current parameter vector w(t) to a selected set S(t). Each client independently performs local optimization and returns an update u_i(t) together with its completion event. Let a_i(t) denote the simulated response time, including a heterogeneous compute component and a network component. A quorum q <= |S(t)| defines the finalization boundary. The coordinator sorts observed successful completions by arrival time and aggregates the first q updates. The simulated round latency is therefore the q-th order statistic of {a_i(t)}. When q=|S(t)| the protocol reduces to full synchronization; smaller q values allow the round to advance before the slowest workers finish.

3.3 Staleness semantics

The system uses version-strict rather than staleness-weighted semantics. Every training command carries a round identifier. A returned update is accepted only when its identifier matches the currently collecting round and its worker belongs to the selected set. Once the round quorum finalizes, any later result for that round is drained as an obsolete completion and is not applied to the next model version. Hence the system bounds semantic staleness to zero among applied updates: accepted updates are all derived from the same w(t). Physical lateness still exists, but it is converted into excluded work rather than cross-version aggregation. We call the number |S(t)|-q the stale-exclusion capacity of a round.

3.4 Adversarial participation

A malicious client can modify its local model delta before submission. The primary adversarial experiment uses scaled sign flipping, u_i(t) <- -gamma u_i(t) with gamma = 5. This attack reverses the descent direction and magnifies its norm. Twenty percent of the 20-client population is malicious. The malicious identities are sampled once per run and remain fixed, while their presence inside each quorum depends on response ordering. This creates a realistic systems interaction: a lower quorum may include fewer malicious clients on average, but it could also concentrate them if adversaries are systematically

faster. Our delay process is independent of malicious status, so the study does not model timing-manipulation attacks.

3.5 Robust aggregation control

The primary study uses coordinate-wise median aggregation over the q accepted model deltas. Median is held fixed to avoid conflating the present question with a comparison of robust aggregation algorithms. For each model coordinate j, the server takes the median of {u_i,j(t) : i in Q(t)}, where Q(t) is the set of first q successful responders. This choice supplies a basic Byzantine-resistant control while keeping the experimental focus on non-IID drift and quorum scheduling.

4 Experimental methodology

4.1 Controlled data and model

The controlled benchmark uses a four-class synthetic classification problem with 20 input features. For each seed, a latent linear weight matrix is sampled and used to generate 3,000 training examples and 1,000 independent test examples from Gaussian features with additive logit noise. This construction provides a known, reproducible decision structure without external dataset acquisition or preprocessing variability. Local optimization uses a linear softmax classifier initialized at zero. Each client performs one full-batch gradient step per federated round with learning rate 0.35. The deliberately compact model makes it feasible to repeat the full experimental matrix across multiple seeds while isolating scheduling and distribution effects.

4.2 Partition regimes

Twenty clients participate in every experimental run. The IID condition randomly permutes samples and divides them into equal partitions. Non-IID conditions use an unbalanced class-wise symmetric Dirichlet allocation. For each class, client proportions are sampled from Dirichlet(alpha), then class examples are assigned multinomially according to those proportions. A minimum of five samples per client is enforced. We quantify realized label skew using one minus normalized class entropy, averaged across clients; a value near zero indicates balanced class composition and larger values indicate stronger concentration.

4.3 Delay and quorum model

Each client receives a stochastic response delay independently at every round. The delay is drawn from a right-skewed log-normal distribution and includes a small sample-count-dependent term to reflect additional local work for larger partitions. The same delay model is used for benign and malicious clients. We evaluate q in {20,16,12}. With 20 selected clients, these settings respectively exclude at most 0%, 20% and 40% of late results from the finalized round. Cumulative latency is the sum of per-round q-th response times over 25 rounds; it is a simulated systems metric rather than measured wall-clock execution time on a particular machine.

4.4 Experimental matrix and replication

Table 1
Experimental design.

Factor	Setting	Notes
Clients	20	Fixed population per run
Rounds	25	All conditions
Seeds	5	11, 29, 47, 65, 83
Partitions	3	IID, Dirichlet 0.5, Dirichlet 0.2
Quorums	3	20/20, 16/20, 12/20
Malicious fractions	2	0%, 20%
Attack	Sign flip	Scale gamma = 5
Aggregation	Median	Coordinate-wise
Primary outcomes	Accuracy, loss, latency	Final and per-round

The factorial design contains 18 conditions per seed and 90 complete runs, for 2,250 federated rounds. Random seeds determine the generated classification problem, partitions, malicious identities and delay draws. Final accuracy is summarized as mean and sample standard deviation over five seeds. Paired t-tests compare q = 12 with q = 20 within the same seed and partition/adversarial condition. Because only five seeds are used and the benchmark is synthetic, p-values are interpreted as descriptive evidence rather than definitive population-level inference.

4.5 Implementation-validation evidence

Separate functional validation of the multi-process coordinator verifies three properties relevant to the study design: persistent worker processes complete repeated rounds without deadlock; a quorum smaller than the selected population permits the coordinator to complete while marking a slow worker as a straggler; and obsolete or mismatched-round results are ignored rather than merged into a newer round. Dataset-partition validation additionally checks exact non-overlapping coverage and confirms strong class skew for small Dirichlet concentration. These tests establish the coordinator semantics evaluated here, while the controlled benchmark supplies the quantitative convergence and latency results.

4.6 Use of generative language technology

A generative-language system was used for manuscript organization and language editing. It did not generate experimental observations autonomously. Experimental outputs were produced by executable numerical procedures, and all numerical claims, tables and figures were checked against the resulting data before inclusion.

5 Results

5.1 Realized statistical heterogeneity

The mean normalized class-skew score is 0.020 for IID partitions, 0.356 for Dirichlet alpha = 0.5, and 0.567 for alpha = 0.2. The monotonic increase confirms that the experimental partition regimes create progressively stronger label concentration. This matters because robust aggregation must distinguish adversarial deviations from the legitimate variation produced by client-specific objectives.

5.2 Quorum completion and latency

Reducing the response quorum produces the expected systems benefit. Across partition regimes and malicious conditions, q = 16 reduces cumulative simulated latency by approximately 56.5–57.9% relative to q = 20. The q = 12 configuration reduces latency by approximately 71.6–72.4%. Because the delay model is independent of client trust status, the latency reductions are nearly identical in benign and adversarial runs. These gains arise from avoiding the heavy tail of the response-time distribution rather than from faster local optimization.

5.3 Accuracy under benign participation

With no malicious clients, final accuracy after 25 rounds is stable across quorum sizes within each data regime. For IID data, mean final accuracy ranges from 91.68% at q = 20 to 91.98% at q = 16 and 91.88% at q = 12. At alpha = 0.5 the corresponding values are 88.74%, 88.42% and 88.32%; at alpha = 0.2 they are 82.60%, 83.56% and 83.20%. Thus, lowering the quorum does not produce a monotonic loss in accuracy. The paired q = 12 versus q = 20 differences are not statistically significant for IID (mean + 0.20 pp, p = 0.50), alpha = 0.5 (-0.42 pp, p = 0.58), or alpha = 0.2 (+ 0.60 pp, p = 0.56). In this controlled setting, stale-update exclusion converts stragglers into discarded work without measurable degradation of the final model.

5.4 Malicious participation and the heterogeneity interaction

The effect of 20% malicious participation is strongly conditioned on data heterogeneity. Under IID data, q = 20 falls from 91.68% to 90.10%, a 1.58-point penalty. At q = 12 the penalty is 2.10 points. Under alpha = 0.5, the penalty is approximately 8.36 points for q = 20 and 8.22 points for q = 12. Under alpha = 0.2, it increases to 11.22 points for q = 20 and 10.30 points for q = 12. Paired tests between benign and malicious runs are significant at all quorum levels for each partition regime, with the largest absolute effects occurring under the strongest non-IID split.

The observed pattern supports the hypothesis that heterogeneity amplifies adversarial damage even when a robust coordinate-wise statistic is used. Honest client deltas become less mutually consistent as local class distributions diverge. A fixed sign-flipping attack therefore operates against a wider benign update distribution, and the median has less concentrated honest structure from which to recover a

stable direction. This is consistent with recent analyses of Byzantine FL under label skew and client subsampling [11–13].

5.5 Does a lower quorum worsen adversarial convergence?

Within the tested delay model, reducing q from 20 to 12 does not significantly change final accuracy under attack. The paired differences are − 0.32 percentage points for IID data (p = 0.44), -0.28 points at alpha = 0.5 (p = 0.86), and + 1.52 points at alpha = 0.2 (p = 0.56). This result should not be interpreted as a general guarantee that smaller quorums are safe. The delay process in this experiment is independent of malicious identity, so the expected malicious fraction among early responders remains close to the population fraction. Across rounds, the 12-client quorum contains approximately 2.4–2.6 malicious updates on average, compared with four in the full 20-client set. An attacker able to manipulate response timing could violate this assumption and increase its representation among early responders.

Table 2
Final accuracy and simulated cumulative latency over 25 rounds (mean over five seeds).

Partition	Malicious	Quorum	Final accuracy (%)	Cum. latency	Reduction
IID	0%	20	91.68 +/- 0.79	16.80	0.0%
IID	0%	16	91.98 +/- 0.74	7.12	57.6%
IID	0%	12	91.88 +/- 0.65	4.64	72.4%
IID	20%	20	90.10 +/- 0.88	16.38	0.0%
IID	20%	16	89.82 +/- 0.58	7.10	56.6%
IID	20%	12	89.78 +/- 0.50	4.61	71.8%
Dirichlet 0.5	0%	20	88.74 +/- 1.70	16.81	0.0%
Dirichlet 0.5	0%	16	88.42 +/- 1.72	7.08	57.9%
Dirichlet 0.5	0%	12	88.32 +/- 2.09	4.65	72.4%
Dirichlet 0.5	20%	20	80.38 +/- 4.92	16.40	0.0%
Dirichlet 0.5	20%	16	80.08 +/- 6.28	7.13	56.5%
Dirichlet 0.5	20%	12	80.10 +/- 5.38	4.65	71.7%
Dirichlet 0.2	0%	20	82.60 +/- 2.38	16.87	0.0%
Dirichlet 0.2	0%	16	83.56 +/- 2.38	7.12	57.8%
Dirichlet 0.2	0%	12	83.20 +/- 2.83	4.69	72.2%
Dirichlet 0.2	20%	20	71.38 +/- 6.47	16.46	0.0%
Dirichlet 0.2	20%	16	72.96 +/- 7.55	7.09	56.9%
Dirichlet 0.2	20%	12	72.90 +/- 8.39	4.67	71.6%

6 Discussion

6.1 Quorum asynchrony is a systems optimization with semantic consequences

The results demonstrate why asynchronous design should be described in terms of explicit update semantics rather than only worker concurrency. A coordinator that accepts stale gradients from older model versions faces one optimization problem; a coordinator that rejects obsolete results faces another. The latter design studied here produces large reductions in waiting time while keeping all applied updates version-consistent. Its hidden cost is wasted local computation. When q = 12, eight of twenty results can arrive too late to affect the round, so up to 40% of dispatched work may be excluded.

This can be acceptable in simulation or resource-rich edge settings, but it may be undesirable where client energy, bandwidth or privacy budget is scarce.

6.2 Non-IID data dominates the tested accuracy trade-off

The absence of a significant q = 12 versus q = 20 accuracy difference is notable because the latency gain is large. In these experiments, the dominant determinant of final accuracy is not the completion quorum but the client-data distribution. Moving from IID to alpha = 0.2 lowers benign accuracy by roughly nine points and makes the same malicious fraction far more damaging. This suggests that engineering work on asynchronous scheduling cannot substitute for algorithms that address client drift or heterogeneous robust aggregation. A production system may therefore combine quorum completion with methods such as proximal local objectives, control variates, normalized updates, or heterogeneity-aware robust aggregation.

6.3 Adversaries and response selection

Quorum protocols introduce a response-selection channel. If latency is independent of client type, a smaller quorum is approximately a random subset biased only by benign systems characteristics. If malicious clients can deliberately return early, however, they can raise their effective fraction among accepted updates without increasing their population fraction. This attack surface is conceptually related to the client-subsampling effect identified in Byzantine-robust FL [11] and to recent poisoning research against semi-asynchronous defenses [19]. A robust coordinator should therefore monitor not only model-update statistics but also persistent timing patterns and the relationship between identity, selection frequency and accepted-round participation.

6.4 Relationship to staleness-weighted asynchronous FL

The design should not be conflated with FedAsync, FedSA or FedBuff. Those methods are intended to exploit asynchronous updates that may have different ages, often by weighting, buffering or otherwise compensating for staleness [2–4]. Here, model-version staleness is not numerically corrected because stale updates are not admitted. This simplifies correctness reasoning and prevents old gradients from perturbing new models, but it sacrifices information and can bias participation toward consistently fast clients. The result is best characterized as semi-asynchronous or quorum-bounded asynchronous FL with strict stale exclusion.

6.5 Implications for ML systems

For ML systems designers, the principal practical lesson is that quorum size can be tuned as a service-level parameter. Lower q reduces tail latency; higher q uses more of the dispatched work and can improve representativeness. The present data suggest a broad operating region in which q can be reduced substantially without a measurable accuracy penalty, but only under the tested independent-delay model and robust aggregation rule. A deployment should therefore select q jointly with observability for response-time distributions, accepted-client diversity, per-client participation rates, and

model-quality metrics. The same telemetry can reveal whether a low-quorum policy is systematically excluding slow demographic or device groups, a fairness concern not measured here.

6.6 Limitations and threats to validity

First, the quantitative study uses a synthetic linear classification task rather than image, language or sensor datasets. This choice isolates causal factors and enables repeated factorial experiments, but it does not establish state-of-the-art predictive performance on real workloads.

Second, response latency is simulated with an independent log-normal process plus a small data-size term. Real clients exhibit temporal correlation, device-specific speed, network outages and potentially adversarial timing. The measured latency reduction is therefore a relative systems result for the stated model, not a universal wall-clock speedup.

Third, the malicious study uses fixed 20% scaled sign flipping. The implementation supports additional poisoning families, but adaptive, colluding, label-flipping and timing-aware adversaries are outside the main matrix. A lower quorum could be less safe under strategic early-return attacks.

Fourth, coordinate-wise median is held fixed. Results may differ under FedAvg, Krum, trimmed mean or newer heterogeneity-aware robust aggregators. Holding the aggregator constant is intentional because this paper targets asynchronous systems behavior rather than ranking aggregation defenses.

Fifth, only five seeds are used. Paired tests provide useful descriptive evidence but limited statistical power. Larger-scale replication across real datasets and device traces is needed before making deployment-level guarantees.

Finally, strict stale exclusion prevents model-version mixing but can waste computation and may create participation bias. The study does not quantify energy consumption, communication cost, privacy accounting or fairness effects of discarded updates.

7 Conclusion

This study examined quorum-bounded asynchronous federated learning under the coupled pressures of non-IID data, straggling workers and malicious participation. The coordinator advances after a configurable response quorum and excludes late results from obsolete rounds rather than applying staleness weights. Across five seeds and 90 runs, a 12-of-20 quorum reduced simulated cumulative latency by about 72% relative to full synchronization, with no statistically significant final-accuracy difference within any tested partition regime. The stronger effect came from the interaction between data heterogeneity and attack behavior: a 20% sign-flipping population caused only a small penalty under IID data but an approximately 10–11 point penalty under the strongest Dirichlet split. The results indicate that strict stale exclusion can be an effective systems mechanism for controlling stragglers, but it does not solve the statistical-security problem created by heterogeneous malicious clients. Future work should combine quorum scheduling with real device traces, timing-aware adversaries, adaptive robust

aggregation, client-drift correction, larger datasets and explicit accounting for discarded computation and privacy cost.

Declarations

Competing interests

The author declares no competing financial or non-financial interests.

Ethics approval

Not applicable. The study used synthetic data and did not involve human participants, human tissue, personal data, or animals.

Consent to participate

Not applicable.

Consent for publication

Not applicable.

Funding

No external funding was received for this study.

Author contributions

M.S.I.S. conceived and designed the study, developed the system and experimental methodology, performed the experiments and analysis, prepared the figures and tables, and wrote and revised the manuscript. M.S.I.S. approved the final manuscript and accepts responsibility for the work.

Data availability

The data supporting the findings of this study were generated during the reported experiments. No external or proprietary datasets were used in the primary evaluation. The numerical result tables underlying the figures and statistical comparisons are available from the corresponding author upon reasonable request.

References

1. McMahan HB, Moore E, Ramage D, Hampson S, y, Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. Proceedings of AISTATS. PMLR 54:1273–1282
2. Xie C, Koyejo S, Gupta I (2019) Asynchronous federated optimization. arXiv:1903.03934
3. Chen M, Mao B, Ma T (2021) FedSA: A staleness-aware asynchronous federated learning algorithm with non-IID data. Future Generation Comput Syst 120:1–12
4. Nguyen J, Malik K, Zhan H, Yousefpour A, Rabbat M, Malek M, Huba D (2022) Federated learning with buffered asynchronous aggregation. Proceedings of AISTATS. PMLR 151:3581–3607
5. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V (2020) Federated optimization in heterogeneous networks. Proceedings of MLSys
6. Karimireddy SP, Kale S, Mohri M, Reddi S, Stich S, Suresh AT (2020) SCAFFOLD: Stochastic controlled averaging for federated learning. Proceedings of ICML. PMLR 119:5132–5143
7. Wang J, Liu Q, Liang H, Joshi G, Poor HV (2020) Tackling the objective inconsistency problem in heterogeneous federated optimization. Adv Neural Inf Process Syst 33:7611–7623
8. Sun Z, Niu X, Wei E (2024) Understanding generalization of federated learning via stability: Heterogeneity matters. Proceedings of AISTATS. PMLR 238:676–684
9. Blanchard P, El Mhamdi EM, Guerraoui R, Stainer J (2017) Machine learning with adversaries: Byzantine tolerant gradient descent. Adv Neural Inf Process Syst. ;30
10. Yin D, Chen Y, Kannan R, Bartlett P (2018) Byzantine-robust distributed learning: Towards optimal statistical rates. Proceedings of ICML. PMLR 80:5650–5659
11. Allouah Y, Farhadkhani S, Guerraoui R, Gupta N, Pinot R, Rizk G, Voitovych S (2024) Byzantine-robust federated learning: Impact of client subsampling and local updates. Proceedings of ICML. PMLR 235:1078–1114
12. Bao W, Wu J, He J, BOBA (2024) Byzantine-robust federated learning with label skewness. Proceedings of AISTATS. PMLR 238:892–900
13. Huang W, Shi Z, Ye M, Li H, Du B (2024) Self-driven entropy aggregation for Byzantine-robust heterogeneous federated learning. Proceedings of ICML. PMLR 235:20096–20110
14. Kairouz P, McMahan HB, Avent B et al (2021) Advances and open problems in federated learning. Found Trends Mach Learn 14(1–2):1–210. 10.1561/2200000083
15. Li Q, Wen Z, Wu Z, Hu S, Wang N, Li Y, Liu X, He B (2023) A survey on federated learning systems: Vision, hype and reality for data privacy and protection. IEEE Trans Knowl Data Eng 35(4):3347–3366. 10.1109/TKDE.2021.3124599
16. Alahyane A, Comte C, Jonckheere M, Moulines E (2025) Optimizing asynchronous federated learning: A delicate trade-off between model-parameter staleness and update frequency. arXiv:2502.08206
17. Jha AK, Mahendran A, Ghafir I et al (2026) Asynchronous federated reinforcement learning for scalable load balancing in software-defined networks. Discover Comput 29:320

18. Cox B, Malan A, Chen LY, Decouchant J (2024) Asynchronous Byzantine federated learning. arXiv:2406.01438
19. Pang X, Zhao C, Wang Z, Hu J, Wang Y, Wang L, Wei T, Ren K, Chen C (2025) PoiSAFL: Scalable poisoning attack framework to Byzantine-resilient semi-asynchronous federated learning. 34th USENIX Security Symposium. :6461–6479
20. Hsu TMH, Qi H, Brown M (2019) Measuring the effects of non-identical data distribution for federated visual classification. arXiv:1909.06335
21. Li X, Huang K, Yang W, Wang S, Zhang Z (2020) On the convergence of FedAvg on non-IID data. International Conference on Learning Representations
22. Bonawitz K, Eichner H, Grieskamp W et al (2019) Towards federated learning at scale: System design. Proceedings of MLSys
23. Wang H, Kaplan Z, Niu D, Li B (2020) Optimizing federated learning on non-IID data with reinforcement learning. IEEE INFOCOM ; :1698–1707
24. Yang C, Ghaderi J (2024) Byzantine-robust decentralized learning via remove-then-clip aggregation. Proceedings of the AAAI Conference on Artificial Intelligence. ;38(19):21735–21743. 10.1609/aaai.v38i19.30173

Figures

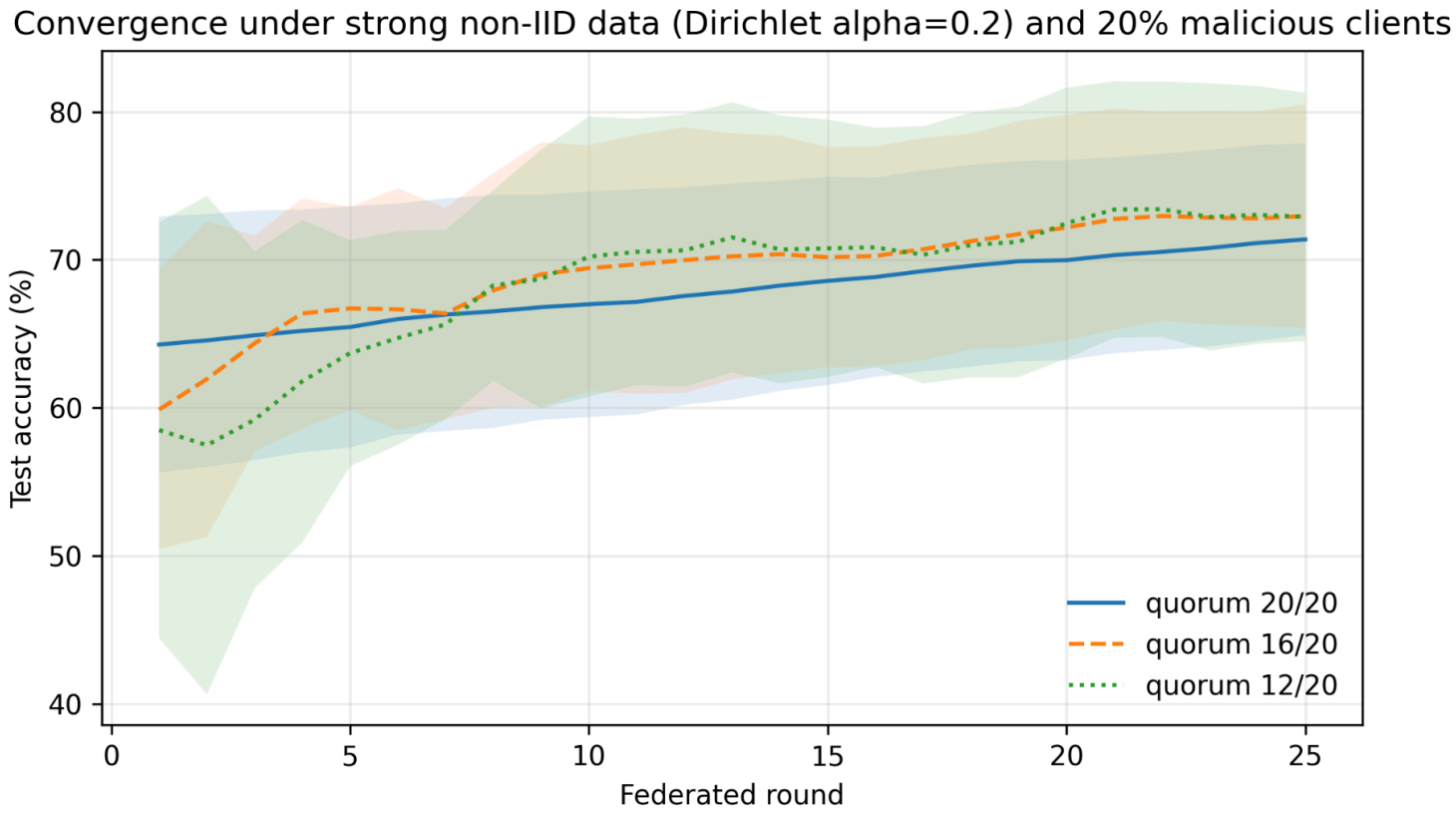


Figure 1

Mean convergence across five seeds for Dirichlet alpha=0.2 with 20% malicious clients. Shaded bands denote one standard deviation.

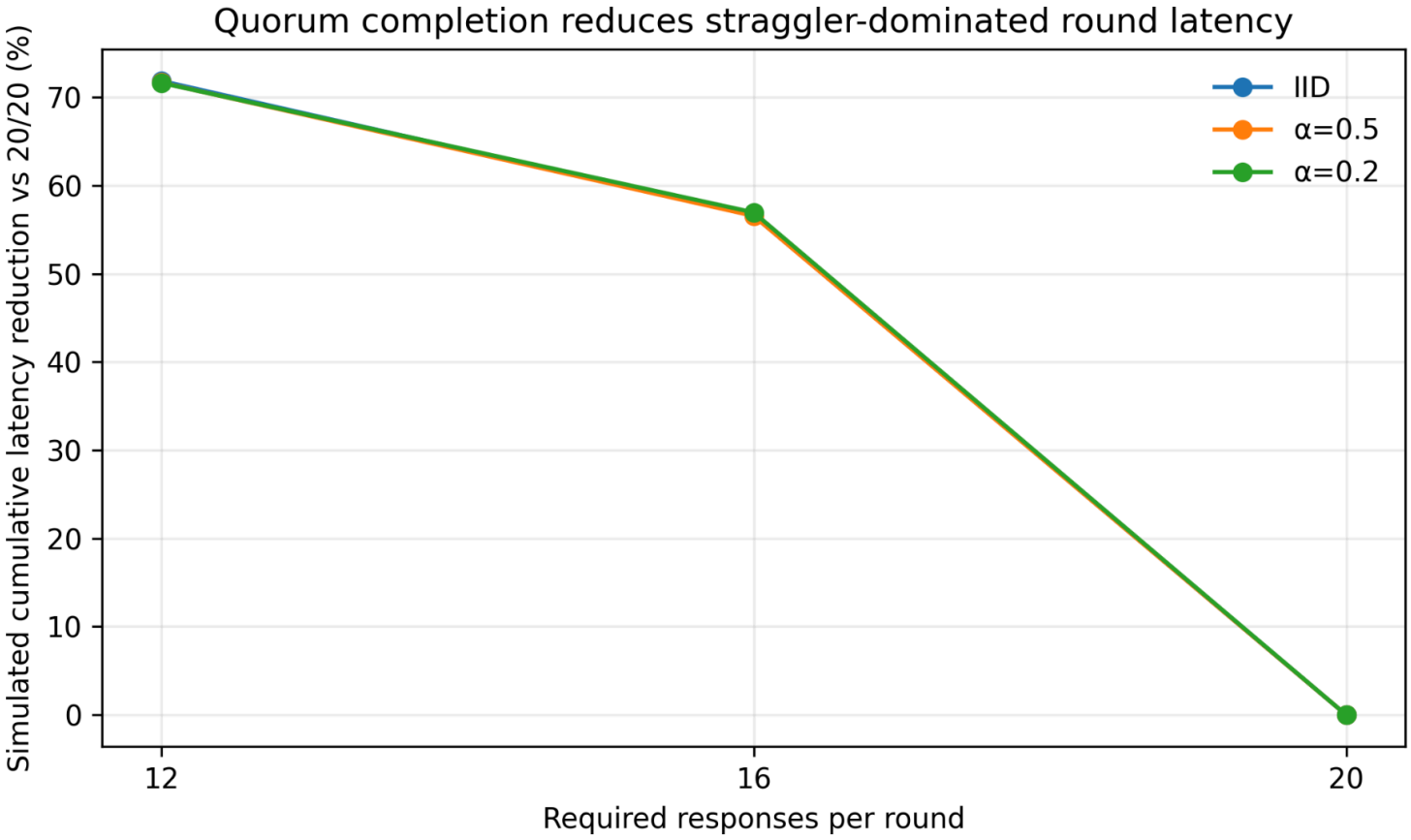


Figure 2

Simulated cumulative latency reduction relative to full synchronization under 20% malicious participation.

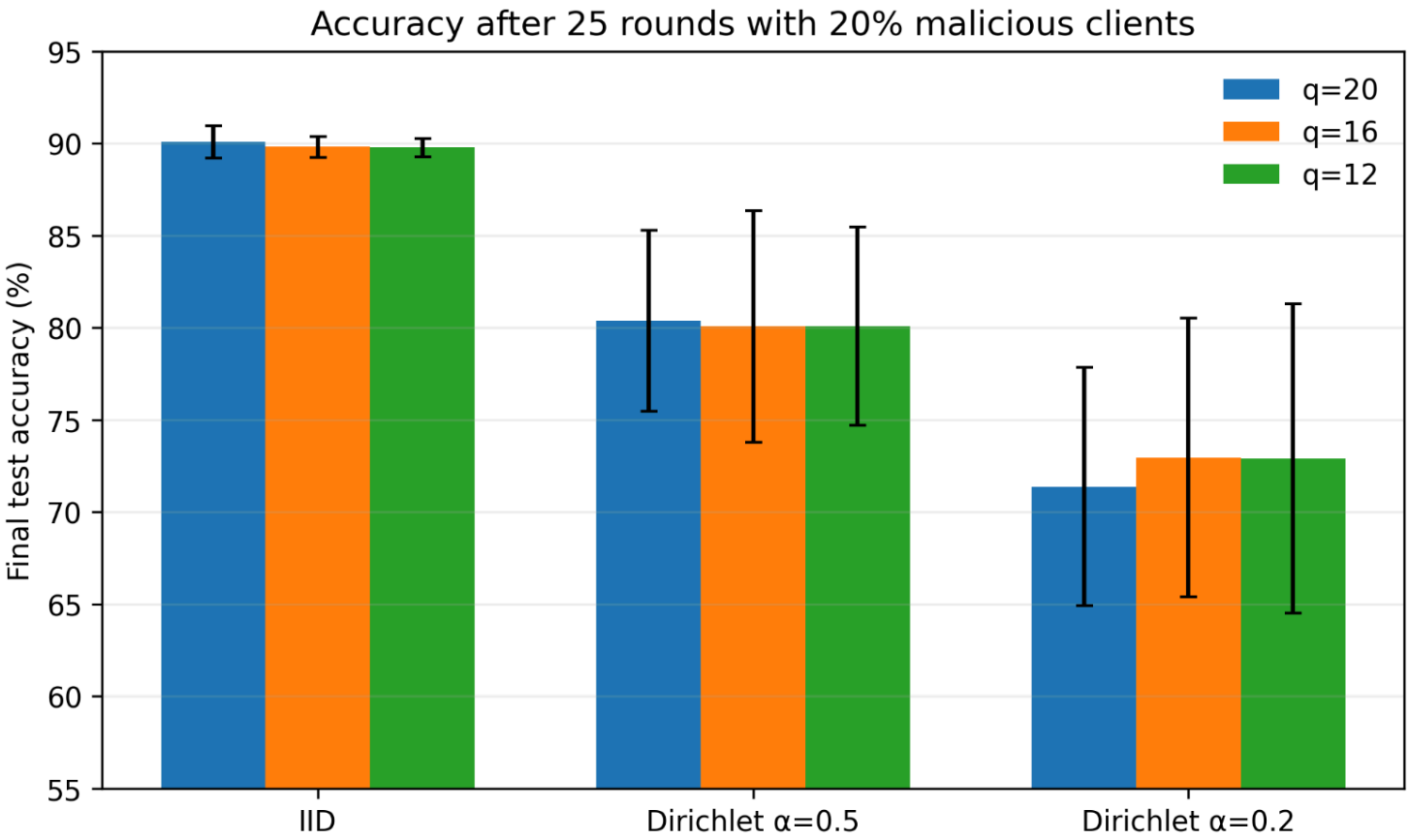


Figure 3

Final accuracy after 25 rounds with 20% malicious clients. Error bars show one standard deviation across five seeds.