BYZANTINE-ROBUST FEDERATED LEARNING ACROSS SELECTION STRATEGIES AND PARTIAL-PARTICIPATION REGIMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) often operates under partial client participation, where only a subset of clients contributes in each communication round. In the presence of Byzantine clients, non-uniform client selection can both bias the observed client population and substantially change the fraction of adversarial clients seen by the server. We propose RAP, a Byzantine-robust federated optimization method for partial participation and general client selection. RAP combines partial-participation bias correction with trial-function-based robust aggregation, validating both fresh updates and the accumulated correction information reused across rounds. The method does not require an honest majority among the clients selected in each round and accommodates stochastic client unavailability. We establish convergence guarantees under heterogeneous client objectives and partial participation. Experiments on heterogeneous CIFAR-10 and real-world ECG classification show that RAP remains stable across uniform and non-uniform client-selection strategies and consistently outperforms Byzantine-robust baselines in challenging adversarial regimes.

1 INTRODUCTION

Federated Learning (FL) enables clients to train a shared model without exchanging raw data (Konečnỳ et al., 2016; Kairouz et al., 2021). In the *Byzantine setting* (Blanchard et al., 2017), an unknown subset of clients may submit arbitrary updates and corrupt optimization (Wang et al., 2023b). Let the M clients be partitioned into an honest set $\mathcal{G}$, with $|\mathcal{G}| = G$, and a malicious set $\mathcal{B}$, with $|\mathcal{B}| = B$, so that $M = G + B$. The robust objective is

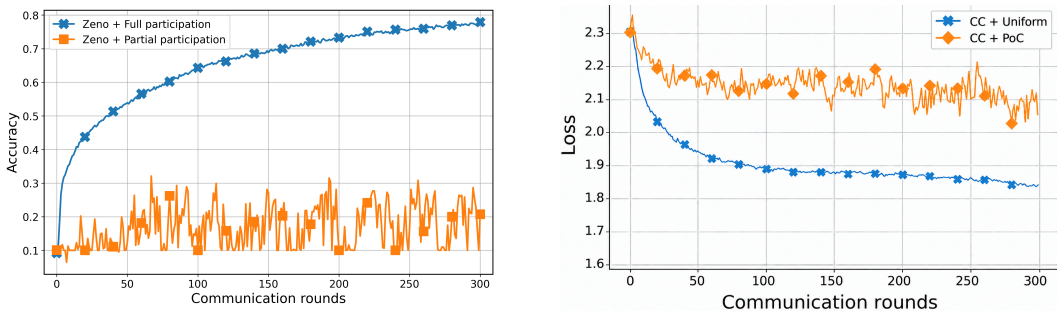
$$\min_{x\in\mathbb{R}^d}\left[f(x) = \frac{1}{G}\sum_{m\in\mathcal{G}}\mathbb{E}_{\xi_m\sim\mathcal{D}_m}\left[f_m(x,\xi_m)\right]\right]. \tag{1}$$

Because the server does not know which clients are honest, it must suppress malicious contributions while approximating the aggregate over $\mathcal{G}$, motivating Byzantine-robust aggregation (Yin et al., 2018; Cao et al., 2020; Karimireddy et al., 2021b).

Realistic FL also relies on *partial participation*: communication, resource, and availability constraints limit each round to a subset of clients (Kairouz et al., 2021; Luo et al., 2022). Under non-IID data, this subset can be statistically unrepresentative and induce selection bias, while intermittent unavailability further distorts participation (Cho et al., 2022; Wang & Ji, 2022). Different client-selection strategies form different participating cohorts and can therefore change both the representation of honest client objectives and the server's exposure to Byzantine clients. Consequently, a selected cohort may be adversary-dominated even when the global malicious fraction is moderate. Figure 1(a) shows that a robust aggregator effective under full participation can degrade sharply under partial participation, while Figure 1(b) shows that non-uniform selection can further alter robust training.

Addressing these effects requires more than robustly aggregating the updates observed in the current round. Correcting the bias introduced by partial participation relies on client information accumulated across rounds; in the Byzantine setting, this delayed correction state becomes an additional

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Zeno: full vs. partial participation

(b) Central Clipping with client selection

Figure 1: CIFAR-10 motivation. (a) Zeno under ALIE with full participation and partial participation with 5 of 25 clients ($\alpha = 0.5$). (b) Central Clipping with Uniform and Power-of-Choice under Label Flip (10 clients, 4 Byzantine clients, $\alpha = 0.1$).

attack surface. A malicious client can therefore influence optimization through both its fresh update and the information carried forward for subsequent rounds. This makes a straightforward combination of client-selection correction and robust aggregation insufficient unless both information paths are protected. RAP is designed around this observation: it robustly validates both fresh client updates and accumulated correction terms while remaining compatible with general client-selection strategies and intermittent client unavailability.

Our contributions.

- We identify accumulated partial-participation correction as an additional Byzantine attack surface: protecting only the fresh updates can leave corrupted delayed information to persist across rounds. We propose RAP, which robustly validates both fresh client updates and accumulated correction terms while supporting general client-selection strategies without requiring an honest majority among the clients selected in each round.
- We establish convergence guarantees under heterogeneous client objectives and partial participation, while accommodating stochastic client unavailability.
- We evaluate RAP on heterogeneous CIFAR-10 and real-world ECG classification with Uniform, OORT, Power-of-Choice, and FEDCBS selection, multiple Byzantine attacks, and the evaluated availability settings. It consistently outperforms established baselines and remains stable in regimes where competing methods substantially degrade.

2 Related Work

Robust Aggregation. Standard distributed and federated methods, including LOCAL SGD, FEDAVG, and SCAFFOLD, aggregate client updates by averaging and are therefore vulnerable to Byzantine submissions (McMahan et al., 2017; Karimireddy et al., 2020; Blanchard et al., 2017). Classical defenses use coordinate-wise robust statistics (Yin et al., 2018), distance-based filtering such as KRUM/MULTI-KRUM (Blanchard et al., 2017), or hybrid procedures such as BULYAN (Guerraoui et al., 2018), but can be circumvented by mean-manipulation attacks including ALIE and INNER PRODUCT MANIPULATION (Baruch et al., 2019a; Xie et al., 2020). Later approaches add validation or computational checks (Alistarh et al., 2018; Allen-Zhu et al., 2020; Gorbunov et al., 2022a), clipping or variance reduction (Karimireddy et al., 2021b; Gorbunov et al., 2022b; Malinovsky et al., 2024), or reliability-based client weighting (Cao et al., 2020; Yan et al., 2023).

A particularly relevant line constructs trust scores with a server-side *trial function* built from a small trusted dataset (Xie et al., 2019; Cao et al., 2020; Park et al., 2021). BANT (Molodtsov et al., 2026) couples trial-function validation with trust-weighted aggregation and removes the standard honest-majority requirement under its assumptions, which is important when a selected cohort may be mostly Byzantine. However, trial-function robustness alone does not resolve the interaction with partial-participation correction. Correcting biased participation introduces accumulated client information that is reused in later rounds, and this delayed state can itself be Byzantine. Consequently,

applying a robust aggregator only to the fresh updates leaves a second adversarial path unprotected. This distinction is central to RAP, which validates both the current updates and the accumulated correction before combining them.

Client selection. Under partial participation, the server must choose a subset of clients in each round. Uniform sampling is the standard baseline (McMahan et al., 2017; Li et al., 2019), but under heterogeneous non-IID data the selected subset may be unrepresentative and distort optimization (Cho et al., 2022; Wang et al., 2023a). Data-dependent strategies therefore score clients using gradient or loss information (Rizk et al., 2021; Nguyen et al., 2020; Chen et al., 2020; Cho et al., 2022), while methods such as FEDCBS and FEDCOR target representative or information-efficient subsets (Zhang et al., 2023; Tang et al., 2022). Theoretical analyses have expanded from uniform sampling to biased and unbiased participation (Cho et al., 2022; Luo et al., 2022; Wang & Ji, 2022); more recently, explicit correction terms have been introduced to compensate for biased client subsampling (Bylinkin et al., 2025).

Byzantine robustness under partial participation. Only a few methods explicitly address both Byzantine robustness and partial participation. BYZ-VR-MARINA-PP (Malinovsky et al., 2024) builds on variance-reduced Byzantine optimization (Gorbunov et al., 2022b), but assumes unbiased sampling. FEDRO (Allouah et al., 2024) analyzes the effects of client subsampling and local updates on federated robust averaging. BANT (Molodtsov et al., 2026) focuses on uniform random selection. The analyses of these methods are tied to particular client-participation or sampling mechanisms and do not establish guarantees for one Byzantine-robust method across different client-selection strategies such as Uniform, OORT, Power-of-Choice, and FEDCBS. Moreover, methods that combine robustness of the current update with delayed or accumulated client information must also control Byzantine contamination of this persistent state. Our experiments in Section 4.3 isolate this failure mode directly. This leaves the joint treatment of general client selection, partial-participation correction, and Byzantine contamination of both fresh and delayed information largely unresolved.

3 METHODOLOGY

3.1 ALGORITHM DESIGN

We present **RAP** (Robustness and Partial participation) in Algorithm 1. The main text shows the server-side orchestration and the two communication time scales. The detailed client-side procedure, including how the epoch-level information is formed and how missing responses are handled, is deferred to Algorithm 2 in Appendix C.

An inner iteration h is one communication round: the selector chooses exactly C of the M clients that may upload a fresh gradient. An epoch k is a block of H^k consecutive rounds. At its boundary, clients return vectors accumulated over the rounds of that epoch. The parameter p controls how often this boundary update occurs: larger p produces shorter epochs and more frequent refreshes, whereas smaller p reduces the refresh frequency but aggregates information over a longer trajectory.

The nonzero coordinates of $\hat{\pi}^{k,h}$ indicate the clients selected in the current round. Any client-selection mechanism can instantiate $\hat{\pi}^{k,h}$, including those in Section 2. Our partial-participation model primarily limits client-to-server participation: only selected and currently available clients upload a fresh gradient, while non-selected clients may still receive the broadcast and maintain their local surrogate state. We separately model stochastic client unavailability through q_m, allowing clients to miss local computations or transmissions altogether. The practical effect of such intermittent availability is evaluated in Section 4.5, Table 5.

Surrogate correction and unavailability. Client subsampling creates a bias relative to full participation, and this bias accumulates over training. To compensate for it, each client forms an accumulated gradient over the current epoch. Under full availability,

$$g_m^{k,h} = \frac{1}{M} \sum_{t=0}^{h-1} \nabla f_m(x^{k,t}, \xi_m^{k,t}), \qquad g_m^{k,0} = 0. \tag{2}$$

Thus, $g_m^{k,h}$ is simply a scaled linear combination of ordinary stochastic gradients evaluated by client m during the epoch. We call it a *surrogate gradient*. At the epoch boundary, the server stores

Algorithm 1 RaP – Server

1: **Input:** $x^{-1,H^{-1}} \in \mathbb{R}^d$, number of epochs K, number of clients M, clients per round C, mean availabilities $\{q_m\}_{m=1}^M$
2: **Parameters:** stepsize $\gamma > 0$, mixing parameter $0 \le \theta \le 1$, epoch-duration parameter $p \in (0, 1]$
3: Initialize $s_m^{-1} \leftarrow 0$ for all $m = 1, \dots, M$
4:
5: **for** $k = 0, \dots, K-1$ **do**
6: $x^{k,0} \leftarrow x^{k-1,H^{k-1}}$
7: Sample $H^k \sim \mathrm{Geom}(p)$
8: **for** $h = 0, \dots, H^k - 1$ **do**
9: Initialize $\hat{\pi}^{k,h} \in \mathbb{R}^M : |\{m : \hat{\pi}_m^{k,h} \neq 0\}| = C$
10: **Broadcast** $(x^{k,h}, \hat{\pi}^{k,h}, \{q_m\}_{m=1}^M)$ to the clients
11: $w^{k,h} \leftarrow \arg\min_{w \in \Delta_1^M} \hat{f}\Big(x^{k,h} - \gamma \sum_{m=1}^M w_m s_m^{k-1}\Big)$ {Validate stored epoch summaries while clients compute.}
12: Initialize $v_m^{k,h} \leftarrow 0$ for all $m = 1, \dots, M$
13: **Receive** timely client responses and set $v_m^{k,h} \leftarrow \nabla f_m(x^{k,h}, \xi_m^{k,h})/q_m$ for each responding client
14: **Update** $\{q_m\}_{m=1}^M$ according to the observed communication outcomes
15: $\tilde{\pi}^{k,h} \leftarrow \arg\min_{w \in \Delta_1^M} \hat{f}\Big(x^{k,h} - \gamma \sum_{m=1}^M w_m v_m^{k,h}\Big)$
16: $x^{k,h+1} \leftarrow x^{k,h} - \gamma \left[(1-\theta) \sum_{m=1}^M \tilde{\pi}_m^{k,h} v_m^{k,h} + \theta \sum_{m=1}^M w_m^{k,h} s_m^{k-1}\right]$
17: **end for**
18: Initialize $s_m^k \leftarrow 0$ for all $m = 1, \dots, M$
19: **Receive** timely epoch-boundary responses and set $s_m^k \leftarrow g_m^{k,H^k}/q_m$ for each responding client
20: **Update** $\{q_m\}_{m=1}^M$ according to the observed communication outcomes
21: **end for**

its corrected version s_m^k and combines it in the next epoch with fresh gradients from the current responders (Line 16). The parameter θ controls the balance between the current and accumulated directions.

For each client, $q_m \in (0, 1]$ denotes its mean availability, i.e., the long-run probability that a requested computation or transmission succeeds. The factors $1/q_m$ provide inverse-probability correction. In the availability-aware client procedure, each successfully computed term in Equation (2) is corrected by $1/q_m$, and the epoch-boundary transmission is handled analogously. The server never waits for all devices: missing fresh or epoch-boundary responses are represented by zero. In practice, q_m can be estimated and periodically updated from communication history.

Byzantine-robust trust weighting. Under partial participation and non-uniform selection, the selected cohort may contain a large Byzantine fraction. RAP therefore validates both update components using a small trusted server dataset. Let $\mathcal{D}_1$ be the trusted reference distribution and define its population objective $f_1(x) = \mathbb{E}_{\xi \sim \mathcal{D}_1}[f_1(x, \xi)]$. f_1 corresponds to the server's dataset. In this setting, we assume that the server participates in training and acts as an honest client during each iteration. Given i.i.d. trusted samples $\xi_1, \dots, \xi_N \sim \mathcal{D}_1$, the trial function is the corresponding empirical objective $\hat{f}(x) = \frac{1}{N} \sum_{i=1}^N f_1(x, \xi_i)$.

Hence $\hat{f}$ is the finite-sample counterpart of the trusted population objective f_1. Technical properties of this construction are deferred to Appendix F.

After receiving fresh gradients, the server solves the trial-function problem in Line 15; updates that produce a favorable trusted-data step receive larger weights. The same validation is necessary for the stored surrogate gradients because Byzantine clients may also corrupt accumulated correction terms. Their weights $w^{k,h}$ are computed in Line 11. This optimization depends only on the current model and information stored before the round, so it can be performed in parallel with current client computation. Appendix D describes the optimization procedure and reports its empirical runtime.

3.2 THEORETICAL ANALYSIS

We state the assumptions used in the convergence analysis.

Assumption 3.1. The function $\hat{f}$ is L-smooth, i.e.,

$$\|\nabla \hat{f}(x) - \nabla \hat{f}(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^d.$$

The smoothness assumption is standard in optimization (Gorbunov et al., 2020; Richtárik et al., 2021).

Assumption 3.2. The function $\hat{f}$ is convex, i.e.,

$$\hat{f}(y) \geq \hat{f}(x) + \langle \nabla \hat{f}(x), y - x \rangle \quad \forall x, y \in \mathbb{R}^d.$$

Our theoretical analysis characterizes the interaction between partial participation, Byzantine robustness, and heterogeneous client objectives in the smooth convex regime. The experiments separately evaluate the resulting method on non-convex deep neural networks.

Assumption 3.3. For every honest client $m \in \mathcal{G}$,

$$\|\nabla f_m(x) - \nabla f(x)\|^2 \leq \delta_1 \|\nabla f(x)\|^2 + \delta_2 \quad \forall x \in \mathbb{R}^d.$$

The trusted population objective obeys the same relation:

$$\|\nabla f_1(x) - \nabla f(x)\|^2 \leq \delta_1 \|\nabla f(x)\|^2 + \delta_2.$$

This bounded-heterogeneity condition is standard in Byzantine optimization (El-Mhamdi et al., 2021; Gorbunov et al., 2022b; Allouah et al., 2024); without a relation of this type, convergence for heterogeneous convex objectives is impossible in general (Gupta & Vaidya, 2020).

Assumption 3.4. Each honest client $m \in \mathcal{G}$ has access to an independent unbiased stochastic gradient,

$$\mathbb{E}[\nabla f_m(x, \xi_m)] = \nabla f_m(x).$$

Its variance is bounded as

$$\mathbb{E}\|\nabla f_m(x, \xi_m) - \nabla f_m(x)\|^2 \leq \sigma^2.$$

This local stochasticity condition is standard (Gorbunov et al., 2020; Khaled et al., 2020).

3.2.1 CONVERGENCE RESULT.

For the proof, define binary response indicators $\eta_m^{k,h}, \psi_m^k \in \{0, 1\}$. The first records whether client m completes its iteration-level computation and transmission; the second records whether it returns its epoch-level surrogate. We assume

$$\mathbb{E}[\eta_m^{k,h}] = \mathbb{E}[\psi_m^k] = q_m$$

and independence across the events used by the proof. The implemented corrections are then $\eta_m^{k,h} \nabla f_m / q_m$ and $\psi_m^k g_m / q_m$, exactly matching the responder-based pseudocode in Algorithms 1 and 2.

Theorem 3.5. *Suppose Assumptions 3.1, 3.2, 3.3, and 3.4 hold, choose $p = C/M$. Then Algorithm 1 requires*

$$\mathcal{O}\left(\frac{M^2}{C}\left(\frac{\Delta L}{\varepsilon^2} + \frac{\Delta L(\sigma^2 + \delta_2)}{\varepsilon^4 \min_{m \in \mathcal{G}} q_m} + \frac{\Delta \delta_2 L}{\varepsilon^4 \min_{m \in \mathcal{G}} q_m} + \zeta(N)\right)\right)$$

device communications to reach ε-accuracy.

The effective stepsize and mixing parameter must be reduced in proportion to $\min_{m \in \mathcal{G}} q_m$. Accordingly, the variance and heterogeneity terms scale as $\mathcal{O}(1/\min_{m \in \mathcal{G}} q_m)$: infrequent responses amplify the effective gradient noise, while the inverse-probability correction preserves the target direction in expectation. The prefactor $M^2/C = M(M/C)$ reflects two distinct effects in our device-communication accounting. Each epoch incurs $C/p + M$ client communications; with the choice $p = C/M$, this is $O(M)$, while the remaining M/C factor captures the penalty from observing only a C/M fraction of clients in a standard round. This latter factor is analogous to the contraction penalty induced by partial information in compressed or error-feedback methods (Richtárik et al., 2021); consequently, the bound is not directly comparable to rates derived under unbiased full-population sampling.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate RaP on two domains. The first is CIFAR-10 Krizhevsky et al. (2009), a 10-class image classification benchmark with 50,000 training and 10,000 held-out test images. We use RESNET-18 He et al. (2016), cross-entropy loss, and report test loss and accuracy.

Table 1: Federated learning setup used in the experiments.

Parameter	**CIFAR-10**	**ECG**
Total clients	25	100
Clients per round	5	25
Byzantine clients	5	20
Dirichlet parameter	$\alpha = 0.5$	$\alpha = 5$
Trial dataset	CIFAR-10 subset	PTB-XL

The second domain is a private real-world 12-lead ECG dataset collected from hospitals. The task is multi-label classification of six clinically relevant abnormalities: Atrial Fibrillation, Premature Ventricular Complex, Complete Left (Right) Bundle Branch Block, Left Anterior Fascicular Block, and first-degree AV Block. We use RESNET1D-18, an adaptation of ResNet for one-dimensional time-series signals, with binary cross-entropy loss and evaluate the macro-averaged G-mean between specificity and sensitivity.

For server-side validation, CIFAR-10 uses a balanced trusted subset of 500 training samples, while ECG uses a subset of PTB-XL Wagner et al. (2020), a public ECG dataset. Appendix E studies sensitivity to the trusted reference-set size. Models are selected by client-side validation loss, and the held-out test set is used only for final evaluation. Client data are partitioned with a Dirichlet distribution Hsu et al. (2019); smaller concentration values induce stronger heterogeneity. The FL setup is summarized in Table 1, with further implementation details and hyperparameters in Appendix A.

Byzantine attacks. We consider representative Byzantine attacks covering data, direction, random, and coordinated corruptions. Label Flip randomly relabels malicious clients' local samples; Sign Flip multiplies Byzantine gradients by -1; Random Grad replaces the update with a random gradient; and ALIE sends a scaled and shifted average gradient designed to mimic benign updates Baruch et al. (2019b).

Baselines. We compare against FEDAVG McMahan et al. (2017), CENTERED CLIP (CC) Karimireddy et al. (2021b), CENTERED CLIP WITH BUCKETING (CC+Bucketing) Karimireddy et al. (2021a), FLTRUST Cao et al. (2020), RECESS Yan et al. (2023), ZENO Xie et al. (2019), BANT Molodtsov et al. (2026), and BYZ-VR-MARINA-PP Malinovsky et al. (2024). The first group of methods was not designed specifically for partial participation, while BANT and BYZ-VR-MARINA-PP explicitly target this regime. We evaluate all methods in a unified pipeline, including under client selection mechanisms not considered in several original studies. Method-specific hyperparameters are listed in Appendix A, Table 8.

Client selection mechanisms. In addition to uniform sampling, we evaluate three client selection mechanisms. POWER-OF-CHOICE (POC) Cho et al. (2022) prioritizes clients with larger losses and is designed to accelerate convergence under data heterogeneity. FEDCBS Zhang et al. (2023) selects clients so that the induced data distribution is closer to a balanced target distribution. OORT Lai et al. (2021) selects clients according to a utility score that combines statistical utility, measured through the reported loss, and system efficiency, measured through client execution time.

4.2 CLIENT SELECTION UNDER BYZANTINE ATTACKS

We first study Byzantine robustness under uniform and non-uniform client selection. Uniform sampling is the standard partial participation baseline: it does not introduce explicit selection bias, but the server still observes only a small subset of clients, which may contain many Byzantine participants. This isolates the difficulty caused by partial participation itself.

Table 2: CIFAR-10 accuracy across uniform and non-uniform client selection with 25 clients, 5 participating clients per round, 5 Byzantine clients, and Dirichlet heterogeneity $\alpha = 0.5$. We report CC, the strongest baseline on average across the evaluated configurations; BANT, which uses trial-function-based Byzantine validation; and BYZ-VR-MARINA-PP, which explicitly targets Byzantine robustness under partial participation. Entries report mean $\pm$ standard deviation over five runs. Complete results for all methods and corresponding losses are provided in Appendix B.

Client Selection	Method	No attack	Label Flip	Sign Flip	Random Grad	ALIE
Uniform	CC	0.850 ± 0.004	0.747 ± 0.005	0.357 ± 0.008	0.765 ± 0.003	0.729 ± 0.004
	BANT	0.843 ± 0.004	0.814 ± 0.014	0.743 ± 0.005	0.100 ± 0.022	0.100 ± 0.020
	BYZ-VR	0.593 ± 0.017	0.507 ± 0.019	0.465 ± 0.020	0.375 ± 0.022	0.272 ± 0.024
	RaP	$\mathbf{0.895 \pm 0.002}$	$\mathbf{0.847 \pm 0.003}$	$\mathbf{0.819 \pm 0.003}$	$\mathbf{0.847 \pm 0.002}$	$\mathbf{0.844 \pm 0.003}$
Oort	CC	0.858 ± 0.004	0.410 ± 0.018	0.086 ± 0.020	0.554 ± 0.014	0.292 ± 0.021
	BANT	0.843 ± 0.005	0.340 ± 0.018	0.316 ± 0.019	0.100 ± 0.021	0.100 ± 0.022
	BYZ-VR	0.587 ± 0.018	0.540 ± 0.019	0.490 ± 0.020	0.398 ± 0.022	0.290 ± 0.023
	RaP	$\mathbf{0.897 \pm 0.002}$	$\mathbf{0.770 \pm 0.004}$	$\mathbf{0.573 \pm 0.008}$	$\mathbf{0.676 \pm 0.006}$	$\mathbf{0.665 \pm 0.006}$
PoC	CC	0.827 ± 0.005	0.384 ± 0.018	0.239 ± 0.009	0.261 ± 0.018	0.366 ± 0.017
	BANT	0.815 ± 0.006	0.533 ± 0.014	0.468 ± 0.014	0.093 ± 0.021	0.112 ± 0.022
	BYZ-VR	0.568 ± 0.018	0.488 ± 0.020	0.421 ± 0.021	0.341 ± 0.023	0.272 ± 0.024
	RaP	$\mathbf{0.906 \pm 0.003}$	$\mathbf{0.818 \pm 0.005}$	$\mathbf{0.651 \pm 0.006}$	$\mathbf{0.756 \pm 0.006}$	$\mathbf{0.728 \pm 0.005}$
FedCBS	CC	0.852 ± 0.004	0.655 ± 0.005	0.736 ± 0.011	0.782 ± 0.003	0.584 ± 0.005
	BANT	0.856 ± 0.004	0.712 ± 0.005	0.739 ± 0.007	0.111 ± 0.021	0.114 ± 0.022
	BYZ-VR	0.582 ± 0.018	0.508 ± 0.020	0.459 ± 0.021	0.369 ± 0.023	0.293 ± 0.024
	RaP	$\mathbf{0.914 \pm 0.002}$	$\mathbf{0.856 \pm 0.003}$	$\mathbf{0.831 \pm 0.004}$	$\mathbf{0.833 \pm 0.002}$	$\mathbf{0.858 \pm 0.003}$

We then evaluate structured client selection mechanisms, which are commonly used to improve communication efficiency or reduce statistical bias under heterogeneous data. Since they change which clients contribute to the global update, they can also substantially change the behavior of Byzantine-robust aggregation.

Table 2 compares RAP with three representative baselines. Centered Clip (CC) is the strongest baseline on average across the 20 client-selection/attack configurations in our study. BANT is particularly relevant because it also uses a trusted server-side objective for Byzantine validation, while BYZ-VR-MARINA-PP is designed specifically for Byzantine robustness under partial participation. Results for all remaining baselines, together with the corresponding loss values, are reported in Appendix B.

Under Uniform selection, the baselines exhibit complementary strengths: BANT remains competitive under Label Flip and Sign Flip, whereas CC is considerably stronger under Random Grad and ALIE. RAP remains consistently strong across all attacks rather than specializing to a particular corruption pattern. The distinction becomes more pronounced under non-uniform selection. Both OORT and Power-of-Choice alter the composition of the participating cohort, and the performance of the competing defenses varies substantially across attacks. In contrast, RAP retains stable accuracy across both selectors.

FEDCBS provides a more favorable selection regime for several robust baselines by constructing more representative client subsets, but it does not itself provide Byzantine protection. Even in this regime, RAP achieves the highest accuracy in every evaluated attack configuration. Overall, the results indicate that robust aggregation and client selection cannot be treated independently: a defense that performs well under one selector or attack may degrade sharply under another, whereas RAP remains stable across the considered combinations.

4.3 WHY THE SURROGATE CORRECTION MUST BE ROBUST

We isolate whether Byzantine protection is required for the delayed partial-participation correction itself. We compare three variants of RAP. *Fresh-only* sets $\theta = 0$, removing the surrogate term while retaining trial-function validation of the fresh updates. *Unvalidated surrogate* additionally restores the accumulated correction, but applies Byzantine validation only to the fresh updates; the surrogate gradients are combined with uniform weights $w_m = 1/M$. This corresponds to a straightforward composition of a partial-participation correction mechanism with a robust aggregator: the currently

Table 3: Ablation of Byzantine validation for the accumulated surrogate correction on CIFAR-10. Accuracy is reported under two attacks and two client-selection strategies. *Unvalidated surrogate* protects the fresh updates but leaves the delayed correction term unprotected.

	Random Grad		**ALIE**	
Variant	**Uniform**	**OORT**	**Uniform**	**OORT**
Fresh-only ($\theta = 0$)	0.804 ± 0.005	0.520 ± 0.012	0.792 ± 0.006	0.486 ± 0.013
Unvalidated surrogate	0.612 ± 0.018	0.298 ± 0.021	0.548 ± 0.020	0.241 ± 0.019
Full RAP	$\mathbf{0.847 \pm 0.002}$	$\mathbf{0.676 \pm 0.006}$	$\mathbf{0.844 \pm 0.003}$	$\mathbf{0.665 \pm 0.006}$

observed gradients are protected, while the delayed client information introduced by bias correction is not. Full RAP instead validates both paths.

We evaluate the three variants under RANDOM GRAD and ALIE with both Uniform and OORT selection. Uniform isolates the effect of partial participation, while OORT additionally changes the participating cohort through non-uniform selection.

The gap between Fresh-only and Unvalidated surrogate is particularly informative. Fresh-only discards the accumulated correction but retains robustly validated current updates, whereas the unvalidated variant allows corrupted historical information to persist and interfere with the robust fresh direction. Full RAP avoids this failure mode by validating both paths and consistently performs best across the considered settings. The surrogate-side validation depends only on previously stored information and can therefore be computed in parallel with client-side training; Appendix D reports the implementation details and wall-clock measurements.

4.4 REAL-WORLD TIME-SERIES EXPERIMENT

We further evaluate RAP in a less controlled, deployment-oriented setting: private hospital ECG data distributed across 100 clients. The trial function is built from public PTB-XL dataset, while training data remain private and distributed. We also use the availability-aware procedure from Appendix C, Algorithm 2, with $q_m = 0.3$ for all clients, so clients may become unavailable during training. We test Random Grad and ALIE, the strongest adversarial regimes from our CIFAR-10 study.

Table 4: Real-world ECG performance under Byzantine attacks with 100 clients, 25 participating clients per round, 20 Byzantine clients, client availability $q_m = 0.3$ for all clients, and Dirichlet heterogeneity $\alpha = 5$. G-mean is macro-averaged across pathologies. Statistics are obtained over 3 runs.

Method	**Random Grad**	**ALIE**
FedAvg	0.6183±0.0214	0.4971±0.0268
CC	0.9060±0.0067	0.7935±0.0149
CC+Buck.	0.7653±0.0178	0.6928±0.0216
FLTrust	0.5417±0.0249	0.4382±0.0291
Recess	0.7340±0.0193	0.6528±0.0232
Zeno	0.8220±0.0136	0.7306±0.0194
BANT	0.7920±0.0158	0.6991±0.0206
BYZ-VR	0.5864±0.0227	0.4619±0.0275
RaP	$\mathbf{0.9188 \pm 0.0041}$	$\mathbf{0.8578 \pm 0.0063}$

Table 4 shows that RAP achieves the best G-mean under both attacks. Competing methods are less consistent: each either degrades substantially or remains competitive only under one attack. Thus, RAP remains stable in a setting combining real clinical time-series data, more clients, partial participation, intermittent availability, and an external public trial dataset.

4.5 Ablation and Efficiency Analysis

Client availability. For RAP, we set $q_m = r$ for all clients and vary $r \in \{1.0, 0.9, 0.8, 0.6, 0.4, 0.2\}$, where $r = 1$ corresponds to full availability and smaller values model increasingly severe stochastic unavailability. The baselines in Table 5 are reported only under full availability, so the sweep is intended to isolate the sensitivity of RAP to missing client responses rather than compare methods at $r < 1$. Performance remains stable under moderate unavailability and degrades only when missing responses become severe. This behavior is qualitatively consistent with the theoretical dependence on q_m: inverse-probability correction preserves the target direction in expectation, while reduced availability increases the effective stochasticity of the update.

Table 5: Effect of random client availability on CIFAR-10 under uniform client selection. Baselines are reported only in the fully available setting. RAP is evaluated with client availability probability $q_m = r$ for all clients. Accuracy statistics obtained over 5 runs.

Method	**Availability** r	**No attack**	**Random Grad**	**ALIE**
CC	1.0	0.850±0.004	0.765±0.003	0.729±0.004
Zeno	1.0	0.833±0.004	0.236±0.020	0.178±0.021
BANT	1.0	0.843±0.004	0.100±0.022	0.100±0.020
RaP	1.0	**0.895±0.002**	**0.847±0.002**	**0.844±0.003**
RaP	0.9	0.894±0.003	0.873±0.003	0.859±0.004
RaP	0.8	0.891±0.004	0.869±0.004	0.855±0.005
RaP	0.6	0.883±0.006	0.857±0.007	0.842±0.008
RaP	0.4	0.851±0.010	0.815±0.012	0.798±0.014
RaP	0.2	0.786±0.018	0.731±0.021	0.711±0.024

Effect of the Mixing Parameter. The parameter θ controls the relative contribution of fresh client updates and the accumulated surrogate correction, with $\theta = 0$ removing the surrogate term entirely. We evaluate $\theta \in \{0, 0.15, 0.5, 0.9\}$ on CIFAR-10 under the RANDOM GRAD attack with Uniform, FEDCBS, and POWER-OF-CHOICE selection. Table 6 shows that moderate non-zero values improve over $\theta = 0$ across the considered selectors, whereas $\theta = 0.9$ substantially degrades accuracy. Thus, the accumulated correction is useful when it complements rather than dominates the fresh updates.

Table 6: Effect of the mixing parameter θ on CIFAR-10 accuracy.

θ	Uniform	FEDCBS	POWER-OF-CHOICE
0	0.804 ± 0.005	0.790 ± 0.002	0.753 ± 0.004
0.15	0.847 ± 0.002	$\mathbf{0.833 \pm 0.002}$	$\mathbf{0.756 \pm 0.006}$
0.5	$\mathbf{0.852 \pm 0.004}$	0.822 ± 0.007	0.750 ± 0.002
0.9	0.532 ± 0.006	0.494 ± 0.008	0.510 ± 0.009

5 Conclusion and Future Work

We proposed RAP for Byzantine-robust federated learning under partial participation and general client selection. RAP validates both fresh client updates and accumulated correction information, preventing corrupted delayed state from undermining an otherwise robust current update. Across heterogeneous CIFAR-10 and real-world ECG experiments, RAP remains stable under multiple Byzantine attacks, uniform and non-uniform client selection, and intermittent client availability. The ablations support the role of surrogate validation, while our analysis establishes convergence in the smooth convex regime. Future work includes developing trust-estimation mechanisms that do not require external reference data.

AI USE STATEMENT

Generative AI tools were used solely for formatting purposes and did not impact the core methodology, scientific rigor, or originality of the research. All AI-assisted changes were reviewed by the authors, who take responsibility for the final content of the work.

REPRODUCIBILITY STATEMENT

The anonymized source code is available at https://anonymous.4open.science/r/RaPFL-ICLR.
Implementation and experimental details are provided in the appendices.

REFERENCES

Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. *Advances in neural information processing systems*, 31, 2018.

Zeyuan Allen-Zhu. Katyusha X: Practical momentum method for stochastic sum-of-nonconvex optimization. *arXiv preprint arXiv:1802.03866*, 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, Rafael Pinot, Geovani Rizk, and Sasha Voitovych. Byzantine-robust federated learning: Impact of client subsampling and local updates. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1078–1114. PMLR, 2024. URL `https://proceedings.mlr.press/v235/allouah24a.html`.

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

Moran Baruch, Gilad Baruch, and Yoav Goldberg. *A little is enough: circumventing defenses for distributed learning*. Curran Associates Inc., Red Hook, NY, USA, 2019b.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

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.

Dmitry Bylinkin, Daniil Medyakov, Vladimir Aletov, Nail Bashirov, Michael Diskin, and Aleksandr Beznosikov. A robust training method for federated learning with partial participation. 2025.

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*, 2020.

Wenlin Chen, Samuel Horvath, and Peter Richtarik. Optimal client sampling for federated learning. *arXiv preprint arXiv:2010.13723*, 2020.

Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Towards understanding biased client selection in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 10351–10375. PMLR, 2022.

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). *Advances in neural information processing systems*, 34:25044–25057, 2021.

Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pp. 680–690. PMLR, 2020.

Eduard Gorbunov, Alexander Borzunov, Michael Diskin, and Max Ryabinin. Secure distributed training at scale. In *International Conference on Machine Learning*, pp. 7679–7739. PMLR, 2022a.

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. *arXiv preprint arXiv:2206.00529*, 2022b.

Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International conference on machine learning*, pp. 3521–3530. PMLR, 2018.

Nirupam Gupta and Nitin H Vaidya. Resilience in collaborative optimization: Redundant and independent cost functions. *arXiv preprint arXiv:2003.09675*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

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

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. In *International Conference on Learning Representations*, 2021a.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *International conference on machine learning*, pp. 5311–5319. PMLR, 2021b.

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

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. Oort: Efficient federated learning via guided participant selection. In *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*, pp. 19–35, 2021.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Bing Luo, Wenli Xiao, Shiqiang Wang, Jianwei Huang, and Leandros Tassiulas. Tackling system and statistical heterogeneity for federated learning with adaptive client sampling. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, pp. 1739–1748. IEEE, 2022.

Grigory Malinovsky, Peter Richtárik, Samuel Horváth, and Eduard Gorbunov. Byzantine robustness and partial participation can be achieved at once: Just clip gradient differences. *Advances in Neural Information Processing Systems*, 37:34900–34979, 2024.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Gleb Molodtsov, Daniil Medyakov, Sergey Skorik, Nikolas Khachaturov, Shahane Tigranyan, Vladimir Aletov, Aram Avetisyan, Martin Takáč, and Aleksandr Beznosikov. Bant: Byzantine antidote via trial function and trust scores. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24431–24440, 2026.

Arkadiĭ Semenovich Nemirovskiĭ and David Berkovich Yudin. *Problem complexity and method efficiency in optimization*. John Wiley & Sons, Chichester, 1983. ISBN 978-0471103479.

Hung T Nguyen, Vikash Sehwag, Seyyedali Hosseinalipour, Christopher G Brinton, Mung Chiang, and H Vincent Poor. Fast-convergent federated learning. *IEEE Journal on Selected Areas in Communications*, 39(1):201–218, 2020.

Jungwuk Park, Dong-Jun Han, Minseok Choi, and Jaekyun Moon. Sageflow: Robust federated learning against both stragglers and adversaries. *Advances in neural information processing systems*, 34:840–851, 2021.

Hamza Reguieg, Mohammed El Hanjri, Mohamed El Kamili, and Abdellatif Kobbane. A comparative evaluation of fedavg and per-fedavg algorithms for dirichlet distributed heterogeneous data. In *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1–6. IEEE, 2023.

Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. Ef21: A new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, 34: 4384–4396, 2021.

Elsa Rizk, Stefan Vlaski, and Ali H Sayed. Optimal importance sampling for federated learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3095–3099. IEEE, 2021.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT*, volume 2, pp. 5, 2009.

Minxue Tang, Xuefei Ning, Yitu Wang, Jingwei Sun, Yu Wang, Hai Li, and Yiran Chen. Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10102–10111, 2022.

Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):154, 2020.

Lin Wang, YongXin Guo, Tao Lin, and Xiaoying Tang. Delta: Diverse client sampling for fasting federated learning. *Advances in Neural Information Processing Systems*, 36:47626–47668, 2023a.

Shiqiang Wang and Mingyue Ji. A unified analysis of federated learning with arbitrary client participation. *Advances in neural information processing systems*, 35:19124–19137, 2022.

Yulong Wang, Tong Sun, Shenghong Li, Xin Yuan, Wei Ni, Ekram Hossain, and H Vincent Poor. Adversarial attacks and defenses in machine learning-empowered communication systems and networks: A contemporary survey. *IEEE Communications Surveys & Tutorials*, 25(4):2245–2298, 2023b.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International conference on machine learning*, pp. 6893–6901. PMLR, 2019.

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

Haonan Yan, Wenjing Zhang, Qian Chen, Xiaoguang Li, Wenhai Sun, Hui Li, and Xiaodong Lin. Recess vaccine for federated learning: Proactive defense against model poisoning attacks. *Advances in Neural Information Processing Systems*, 36:8702–8713, 2023.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pp. 5650–5659. Pmlr, 2018.

Jianyi Zhang, Ang Li, Minxue Tang, Jingwei Sun, Xiang Chen, Fan Zhang, Changyou Chen, Yiran Chen, and Hai Li. Fed-cbs: A heterogeneity-aware client sampling mechanism for federated learning via class-imbalance reduction. In *International Conference on Machine Learning*, pp. 41354–41381. PMLR, 2023.

A TECHNICAL DETAILS

This section provides additional details on the technical setup underlying our experiments. We describe the computational resources and software environment used to simulate the federated learning setting, followed by the implementation details related to client-server orchestration, data partitioning, validation, and hyperparameter configurations. Our codebase is available in supplements.

Computational resources. Our implementation is written in Python 3.10 and relies on PyTorch 2.6.0 with CUDA 12.8 for GPU acceleration on Debian GNU/Linux 12 (*bookworm*). All experiments simulate a federated setup on a single server equipped with an AMD EPYC 7513 32-core CPU and NVIDIA A100 GPUs with 80 GB of memory. The same computational setup is used for all methods to ensure a fair comparison.

Simulation framework. The code simulates federated training by maintaining a pool of client workers and a central server. Each client is assigned a local dataset partition and performs local computation independently when selected by the server. The server orchestrates communication rounds, applies the corresponding client selection strategy, receives client updates, and performs aggregation. Since all clients are simulated under the same computational environment, the experiments isolate the statistical and adversarial aspects of partial participation rather than system-level hardware heterogeneity.

Data partitioning and validation. For each dataset, client data are partitioned using a Dirichlet distribution to emulate statistical heterogeneity across clients, following a standard protocol in federated learning Hsu et al. (2019); Zhang et al. (2023); Reguieg et al. (2023). Specifically, for each class, we sample a vector of client proportions from $\mathrm{Dir}(\alpha)$ and allocate samples of this class to clients according to the sampled proportions. The concentration parameter α controls the degree of non-IIDness: smaller values of α produce more skewed client label distributions, where each client tends to contain data from only a few classes, while larger values make client datasets closer to the global distribution. Within each client, the local dataset is split into training and validation subsets. The validation subset is used only for model selection and not for training. Test metrics are computed on a held-out test set after selecting the best communication round according to validation loss.

Hyperparameters. To ensure fair comparison, all methods are evaluated under the same federated learning protocol within each dataset. Table 7 summarizes the main dataset-level hyperparameters. Table 8 presents the hyperparameters of the federated learning methods used in the experiments.

Table 7: Dataset-level hyperparameters used in the experiments.

Parameter	**CIFAR-10**	**ECG**
Input type	RGB image	12-lead signal
Number of classes	10	6
Task type	Multiclass	Multi-label
Model	ResNet-18	ResNet1D-18
Loss function	CrossEntropyLoss	BCEWithLogitsLoss
Optimizer	SGD	SGD
Batch size	64	128
Local epochs	1	1
Preprocessing	Standard normalization	Signal normalization
Augmentations	RandomCrop, HorizontalFlip	None
Trust dataset size	500	1000

B EXPERIMENTS WITH DIFFERENT CLIENT SELECTION STRATEGIES

This section provides the loss and accuracy results for the CIFAR-10 experiments summarized in Section 4.2. The main text reports a compact accuracy table comparing RAP with three represen-

Table 8: Method-specific hyperparameters. Parameters not listed use the common protocol in Table 7.

Method	Hyperparameters
CC	$\beta_{\text{momentum}} = 0.9$, $\tau_{\text{clip}} = 0.1$, 1 clipping iteration
Recess	baseline-decreased score $= 0.5$
Zeno	$\rho = 5 \times 10^{-4}$, $b = 0.2$
BANT	momentum $\beta = 0.5$
Bucketing	$\beta = 0.9$, bucket size $s = 2$
RaP	$\theta = 0.15$, $\gamma = 0.5$, $\gamma_{\text{mirror}} = 1$, $p = C/M$ (0.2 for CIFAR-10, 0.25 for ECG), 10 optimization epochs

Table 9: CIFAR-10 accuracy under uniform client selection with 25 clients, 5 participating clients per round, 5 Byzantine clients, and Dirichlet heterogeneity $\alpha = 0.5$. Entries report mean $\pm$ standard deviation over five runs.

Method	No attack	Label Flip	Sign Flip	Random Grad	ALIE
FedAvg	0.831±0.004	0.755±0.005	0.705±0.006	0.134±0.018	0.110±0.021
CC	0.850±0.004	0.747±0.005	0.357±0.008	0.765±0.003	0.729±0.004
CC+Buck.	0.748±0.009	0.638±0.014	0.417±0.016	0.506±0.013	0.103±0.020
FLTrust	0.641±0.016	0.574±0.018	0.492±0.019	0.438±0.021	0.318±0.023
Recess	0.793±0.007	0.710±0.009	0.540±0.013	0.229±0.019	0.119±0.018
Zeno	0.833±0.004	0.770±0.004	0.763±0.005	0.236±0.020	0.178±0.021
BANT	0.843±0.004	0.814±0.014	0.743±0.005	0.100±0.022	0.100±0.020
BYZ-VR-MARINA-PP	0.593±0.017	0.507±0.019	0.465±0.020	0.375±0.022	0.272±0.024
RaP (ours)	**0.895±0.002**	**0.847±0.003**	**0.819±0.003**	**0.847±0.002**	**0.844±0.003**

tative baselines under each client selection mechanism. Here, we present the complete per-method results for uniform sampling, OORT, POWER-OF-CHOICE, and FEDCBS. These tables complement the main results by showing both optimization quality, through loss, and predictive performance, through accuracy.

Uniform sampling. This setting does not introduce an explicit selection bias and therefore serves as the cleanest test of Byzantine robustness under client subsampling. Several baselines are competitive under individual attacks but are not uniformly reliable. For instance, CENTERED CLIP performs well under Random Grad and ALIE but degrades under Sign Flip, while ZENO and BANT remain strong in some scenarios but collapse in others. RAP obtains the best accuracy in all reported attacks and keeps the loss consistently low. Tables 9 and 10 report accuracy and loss, respectively.

OORT. OORT is particularly challenging in the Byzantine setting because its utility score depends on quantities that malicious clients can influence. Once attacks are introduced, most baselines experience a sharp drop relative to their no-attack performance. RAP also becomes more challenged than under uniform sampling, but it remains the best-performing method in accuracy under every attack scenario. Tables 11 and 12 report accuracy and loss, respectively.

POWER-OF-CHOICE. POC favors high-loss clients. While this can accelerate learning in benign heterogeneous settings, it becomes risky when Byzantine clients are present. Several baselines retain reasonable performance under one attack but degrade under another. RAP maintains the best accuracy across all attack scenarios, indicating that the proposed method tolerates the stronger selection bias induced by POC. Tables 13 and 14 report accuracy and loss, respectively.

FEDCBS. FEDCBS selects client subsets whose data distribution is closer to balanced. This can improve update quality when the aggregation rule suppresses Byzantine contributions. Several baselines perform better than under OORT or POC, but the improvement is not uniform. RAP achieves the best accuracy across all attacks. Tables 15 and 16 report accuracy and loss, respectively.

Training dynamics. For the training curves, we repeated each experiment five times and plot one representative run selected at random. We also account for the additional communication required by RAP at the end of each epoch. Specifically, when RAP performs a full-client synchronization step to

Table 10: CIFAR-10 loss under uniform client selection with 25 clients, 5 participating clients per round, 5 Byzantine clients, and Dirichlet heterogeneity $\alpha = 0.5$. Entries report mean $\pm$ standard deviation over five runs.

Method	No attack	Label Flip	Sign Flip	Random Grad	ALIE
FedAvg	0.425±0.006	0.491±0.008	0.623±0.011	1.842±0.045	2.012±0.052
CC	0.382±0.005	0.485±0.007	0.719±0.012	0.424±0.005	0.461±0.006
CC+Buck.	0.649±0.014	0.959±0.021	1.044±0.025	0.866±0.019	1.699±0.041
FLTrust	1.118±0.031	1.372±0.036	1.694±0.044	1.931±0.047	2.184±0.054
Recess	0.573±0.012	0.674±0.015	0.890±0.021	1.407±0.038	2.135±0.056
Zeno	0.422±0.006	0.477±0.007	0.521±0.009	1.424±0.037	1.760±0.044
BANT	0.586±0.011	0.869±0.019	0.709±0.014	1.672±0.043	1.914±0.050
BYZ-VR-MARINA-PP	1.246±0.032	1.528±0.039	1.771±0.045	2.063±0.052	2.211±0.057
RaP (ours)	**0.379±0.003**	**0.466±0.004**	**0.495±0.005**	**0.414±0.004**	**0.456±0.004**

Table 11: CIFAR-10 accuracy under OORT client selection with 25 clients, 5 participating clients per round, 5 Byzantine clients, and Dirichlet heterogeneity $\alpha = 0.5$. Entries report mean $\pm$ standard deviation over five runs.

Method	No attack	Label Flip	Sign Flip	Random Grad	ALIE
FedAvg	0.838±0.004	0.262±0.021	0.094±0.019	0.096±0.022	0.108±0.023
CC	0.858±0.004	0.410±0.018	0.086±0.020	0.554±0.014	0.292±0.021
CC+Buck.	0.761±0.009	0.177±0.017	0.071±0.018	0.084±0.017	0.115±0.015
FLTrust	0.617±0.017	0.511±0.020	0.437±0.021	0.361±0.023	0.279±0.024
Recess	0.795±0.007	0.302±0.021	0.092±0.022	0.121±0.020	0.120±0.019
Zeno	0.845±0.005	0.174±0.018	0.101±0.017	0.093±0.016	0.096±0.021
BANT	0.843±0.005	0.340±0.018	0.316±0.019	0.100±0.021	0.100±0.022
BYZ-VR	0.587±0.018	0.540±0.019	0.490±0.020	0.398±0.022	0.290±0.023
RaP	**0.897±0.002**	**0.770±0.004**	**0.573±0.008**	**0.676±0.006**	**0.665±0.006**

collect or distribute epoch-level information, we add the equivalent number of partial-participation communication rounds to the horizontal axis. For example, if the system contains M clients and each standard round selects C clients, one full-client synchronization contributes M/C communication rounds. This makes the comparison with the baselines fair in terms of client–server communication. The figures are truncated to a fixed communication budget for readability; final reported metrics are selected according to client-side validation, as described in Appendix A.

Figures 2, 3, 4, 5, and 6 show the training dynamics in the absence of attacks and under the Random Gradient, Label Flip, Sign Flip, and ALIE attacks, respectively. Each figure reports test accuracy as a function of communication rounds for uniform sampling, FEDCBS, POWER-OF-CHOICE, and OORT. Across the considered attack models and client-selection strategies, RAP exhibits stable convergence behavior, whereas competing methods often display high variance, stagnation, or collapse.

C CLIENT UNAVAILABILITY

Algorithm 2 gives the detailed client routine corresponding to Algorithm 1. It is written directly in terms of successful computation and communication events: an unavailable client skips the operation, and the server proceeds with the responses that arrive.

The $1/q_m$ factors compensate for the fact that client m contributes only on a fraction q_m of requested events. In the theoretical model, the iteration-level and epoch-level response indicators have expectation q_m, so the corresponding fresh-gradient and surrogate estimators are unbiased with respect to availability. If q_m is unknown, it can be replaced in the implementation by an empirical estimate from recent communication history.

Empirical effect of client unavailability. We additionally evaluate how random client unavailability affects RAP in the CIFAR-10 setup. We vary the client availability probability $q_m = r$ for

Table 12: CIFAR-10 loss under OORT client selection with 25 clients, 5 participating clients per round, 5 Byzantine clients, and Dirichlet heterogeneity $\alpha = 0.5$. Entries report mean $\pm$ standard deviation over five runs.

Method	No attack	Label Flip	Sign Flip	Random Grad	ALIE
FedAvg	0.425±0.006	2.106±0.053	1.876±0.047	2.012±0.051	2.145±0.056
CC	**0.362±0.005**	1.793±0.045	1.956±0.049	0.998±0.024	2.243±0.058
CC+Buck.	0.649±0.014	2.262±0.057	2.345±0.060	2.364±0.059	2.296±0.061
FLTrust	1.203±0.032	1.534±0.039	1.801±0.046	2.046±0.052	2.284±0.058
Recess	0.573±0.012	2.010±0.051	2.105±0.054	2.077±0.053	2.303±0.060
Zeno	0.464±0.007	2.338±0.059	2.456±0.063	2.319±0.060	1.760±0.044
BANT	0.621±0.012	1.659±0.043	1.715±0.045	2.071±0.053	2.214±0.057
BYZ-VR	1.187±0.033	1.356±0.037	1.623±0.042	1.879±0.049	2.045±0.052
RaP	0.422±0.004	**0.487±0.005**	**0.967±0.011**	**0.820±0.009**	**0.781±0.008**

Table 13: CIFAR-10 accuracy under POWER-OF-CHOICE client selection with 25 clients, 5 participating clients per round, 5 Byzantine clients, and Dirichlet heterogeneity $\alpha = 0.5$. Entries report mean $\pm$ standard deviation over five runs.

Method	No attack	Label Flip	Sign Flip	Random Grad	ALIE
FedAvg	0.855±0.006	0.268±0.019	0.187±0.022	0.171±0.021	0.169±0.019
CC	0.827±0.005	0.384±0.018	0.239±0.009	0.261±0.018	0.366±0.017
CC+Buck.	0.663±0.011	0.310±0.016	0.134±0.018	0.171±0.020	0.502±0.008
FLTrust	0.576±0.018	0.489±0.020	0.412±0.021	0.348±0.023	0.287±0.024
Recess	0.776±0.008	0.420±0.015	0.140±0.020	0.124±0.018	0.143±0.017
Zeno	0.804±0.006	0.247±0.016	0.364±0.018	0.105±0.021	0.394±0.006
BANT	0.815±0.006	0.533±0.014	0.468±0.014	0.093±0.021	0.112±0.022
BYZ-VR	0.568±0.018	0.488±0.020	0.421±0.021	0.341±0.023	0.272±0.024
RaP	**0.906±0.003**	**0.818±0.005**	**0.651±0.006**	**0.756±0.006**	**0.728±0.005**

all clients m, with $r \in \{0.2, 0.4, 0.6, 0.8, 0.9, 1.0\}$. Here, $r = 1.0$ corresponds to full availability, while smaller values represent increasingly severe client unavailability. For reference, we report the strongest baselines only in the fully available setting, since the purpose of this experiment is to isolate the effect of varying availability on RAP.

The results are shown in Table 5 of the main paper. RAP remains stable for moderate availability levels and starts to degrade only under severe unavailability. This behavior is consistent with the theoretical dependence on q_m: lower availability increases the variance and slows down learning, but does not invalidate the update rule.

D COMPUTATIONAL COST OF SURROGATE WEIGHT OPTIMIZATION

RAP requires solving two trial-function-based optimization problems at each server iteration: one for the fresh gradients received from selected clients and one for the surrogate gradients stored from the previous epoch. In this section, we focus on the surrogate-weight computation in Line 11, since it is the part that can be performed while the server is waiting for the selected clients to complete their local computations.

For a fixed iteration (k, h), denote the stored, availability-corrected surrogate gradients by

$$a_m = s_m^{k-1}, \qquad m = 1, \dots, M.$$

The server computes the surrogate weights by approximately solving

$$w^{k,h} \approx \arg\min_{w \in \Delta_1^M} \hat{f}\left(x^{k,h} - \gamma \sum_{m=1}^{M} w_m a_m\right).$$

The key point is that the optimization variable is the weight vector $w \in \mathbb{R}^M$, not the model parameters. Therefore, the dimensionality of this optimization problem is the number of clients rather than the number of neural-network parameters.

Table 14: CIFAR-10 loss under POWER-OF-CHOICE client selection with 25 clients, 5 participating clients per round, 5 Byzantine clients, and Dirichlet heterogeneity $\alpha = 0.5$. Entries report mean $\pm$ standard deviation over five runs.

Method	No attack	Label Flip	Sign Flip	Random Grad	ALIE
FedAvg	0.570±0.012	1.570±0.039	2.031±0.052	2.156±0.055	2.289±0.059
CC	0.435±0.007	1.416±0.036	0.880±0.019	1.284±0.033	1.334±0.034
CC+Buck.	0.955±0.021	1.354±0.035	2.303±0.060	1.943±0.050	0.918±0.018
FLTrust	1.462±0.038	1.776±0.046	2.031±0.052	2.258±0.058	2.417±0.063
Recess	0.613±0.013	1.165±0.029	1.697±0.044	2.229±0.058	2.513±0.066
Zeno	0.625±0.014	1.290±0.033	1.376±0.036	2.010±0.052	0.764±0.012
BANT	1.179±0.030	1.540±0.040	1.323±0.033	2.082±0.054	2.197±0.057
BYZ-VR	1.533±0.040	1.842±0.048	2.116±0.055	2.301±0.060	2.459±0.064
RaP	**0.419±0.004**	**0.953±0.010**	**0.844±0.009**	**0.839±0.009**	**0.744±0.007**

Table 15: CIFAR-10 accuracy under FEDCBS client selection with 25 clients, 5 participating clients per round, 5 Byzantine clients, and Dirichlet heterogeneity $\alpha = 0.5$. Entries report mean $\pm$ standard deviation over five runs.

Method	No attack	Label Flip	Sign Flip	Random Grad	ALIE
FedAvg	0.837±0.004	0.782±0.006	0.669±0.010	0.153±0.014	0.125±0.015
CC	0.852±0.004	0.655±0.005	0.736±0.011	0.782±0.003	0.584±0.005
CC+Buck.	0.771±0.008	0.622±0.009	0.591±0.012	0.360±0.019	0.177±0.016
FLTrust	0.621±0.017	0.548±0.019	0.461±0.021	0.372±0.023	0.309±0.024
Recess	0.805±0.007	0.671±0.010	0.657±0.012	0.320±0.018	0.143±0.017
Zeno	0.857±0.004	0.790±0.004	0.804±0.006	0.392±0.017	0.280±0.019
BANT	0.856±0.004	0.712±0.005	0.739±0.007	0.111±0.021	0.114±0.022
BYZ-VR	0.582±0.018	0.508±0.020	0.459±0.021	0.369±0.023	0.293±0.024
RaP	**0.914±0.002**	**0.856±0.003**	**0.831±0.004**	**0.833±0.002**	**0.858±0.003**

We solve this problem using Mirror Descent on the simplex Nemirovskiĭ & Yudin (1983); Beck & Teboulle (2003). Let

$$\Phi^{k,h}(w) = \hat{f}\left(x^{k,h} - \gamma \sum_{m=1}^{M} w_m a_m\right).$$

At an inner iteration s, mirror descent performs

$$w^{s+1} = \arg\min_{w \in \Delta_1^M} \left\{ \left\langle \nabla_w \Phi^{k,h}(w^s), w \right\rangle + \frac{1}{\lambda} KL(w \,\|\, w^s) \right\},$$

where $\lambda > 0$ is the mirror-descent stepsize. The gradient is taken only with respect to the aggregation weights. By the chain rule, its j-th coordinate is

$$\left[\nabla_w \Phi^{k,h}(w^s)\right]_j = -\gamma \left\langle \nabla \hat{f}\left(x^{k,h} - \gamma \sum_{m=1}^{M} w_m^s a_m\right), a_j \right\rangle.$$

Using the entropy geometry, the update has the closed form

$$w_j^{s+1} = \frac{w_j^s \exp\left(-\lambda \left[\nabla_w \Phi^{k,h}(w^s)\right]_j\right)}{\sum_{i=1}^{M} w_i^s \exp(-\lambda \left[\nabla_w \Phi^{k,h}(w^s)\right]_i)}.$$

Thus, each optimization step only requires gradients with respect to M scalar weights. In contrast, a client update optimizes over the full model parameter space and processes the local training data. This makes the surrogate-weight optimization substantially cheaper than a standard local training step.

In our implementation, the optimization is performed as follows. First, the server aggregates the stored surrogate gradients according to the current weights and forms one candidate update direction.

Table 16: CIFAR-10 loss under FEDCBS client selection with 25 clients, 5 participating clients per round, 5 Byzantine clients, and Dirichlet heterogeneity $\alpha = 0.5$. Entries report mean $\pm$ standard deviation over five runs.

Method	No attack	Label Flip	Sign Flip	Random Grad	ALIE
FedAvg	0.620±0.013	0.579±0.012	0.755±0.016	2.497±0.064	2.336±0.060
CC	0.389±0.005	0.688±0.007	1.001±0.024	0.597±0.005	0.684±0.006
CC+Buck.	0.590±0.012	0.610±0.013	1.340±0.035	1.683±0.044	5.119±0.091
FLTrust	1.287±0.034	1.496±0.039	1.763±0.046	2.071±0.054	2.238±0.058
Recess	0.562±0.011	0.741±0.016	1.224±0.031	1.251±0.032	3.776±0.078
Zeno	0.669±0.014	0.581±0.012	0.963±0.023	1.251±0.032	1.367±0.035
BANT	0.763±0.016	0.594±0.012	0.896±0.020	1.742±0.045	1.926±0.050
BYZ-VR	1.401±0.036	1.653±0.043	1.884±0.049	2.143±0.055	2.317±0.060
RaP	**0.375±0.003**	**0.558±0.005**	**0.584±0.006**	**0.554±0.005**	**0.542±0.005**

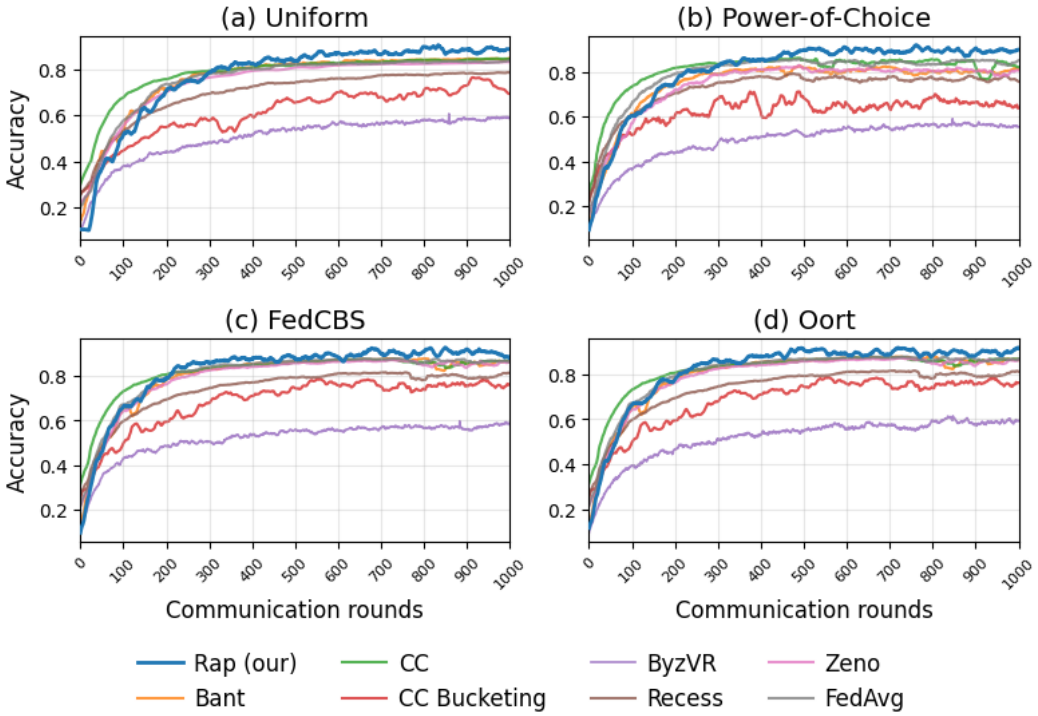


Figure 2: Performance comparison across client-selection strategies in the absence of attacks.

Then it evaluates a single model on the trusted dataset, computes the trial loss, and backpropagates only through the aggregation weights. After applying the mirror-descent update above, the server re-aggregates the surrogate gradients with the updated weights and repeats the procedure for a small number of inner iterations. Therefore, the memory footprint is also modest: the server does not need to instantiate or train M independent models. It evaluates one aggregated model at a time, and the optimized variable is only the vector of weights.

Because Line 11 depends only on $x^{k,h}$ and surrogate gradients stored from the previous epoch, all information required for this optimization is available immediately after the server broadcasts the current model. The surrogate-weight computation can therefore be overlapped with client-side training.

Empirical timing comparison. We measure this overlap over 200 CIFAR-10 rounds. Each selected client performs one local epoch, while RAP uses 10 inner optimization epochs for the sur-

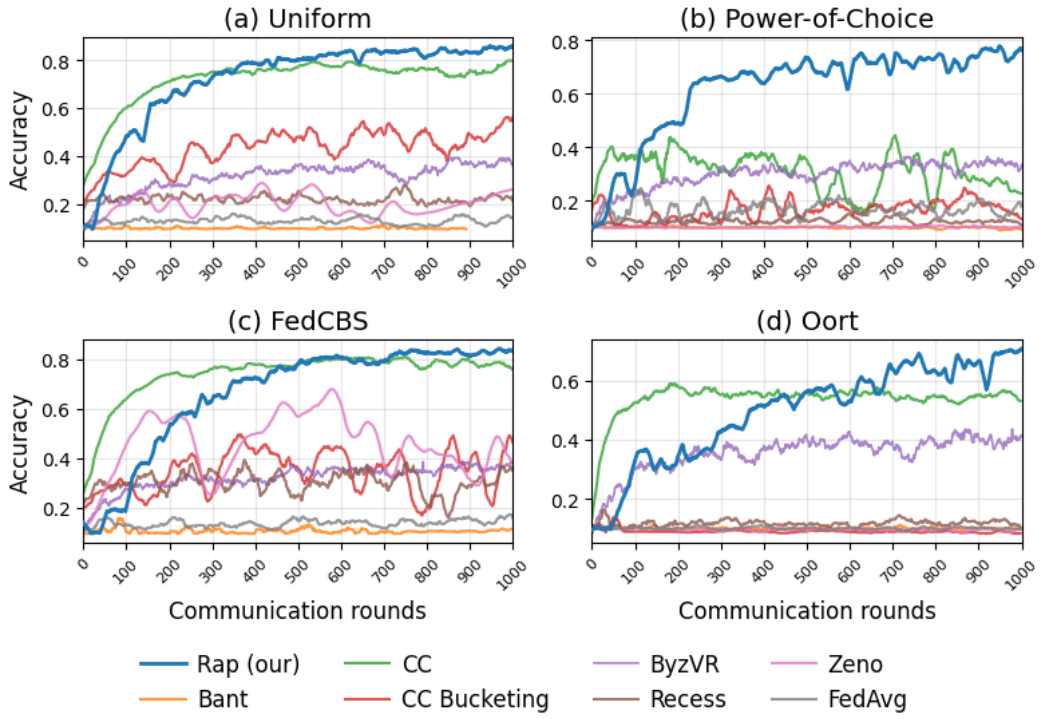


Figure 3: Performance comparison across client-selection strategies under the Random Gradient attack.

Algorithm 2 Availability-aware RaP – Client m

for $k = 0, \ldots, K-1$ **do**
  $g_m^{k,0} \leftarrow 0$
  for $h = 0, \ldots, H^k - 1$ **do**
    if client m receives the round-h broadcast and is available to compute **then**
      Receive $(x^{k,h}, \hat{\pi}^{k,h}, q_m)$ from the server
      $g_m^{k,h+1} \leftarrow g_m^{k,h} + \frac{1}{Mq_m} \nabla f_m(x^{k,h}, \xi_m^{k,h})$
      if $\hat{\pi}_m^{k,h} \neq 0$ **then**
        Send $\nabla f_m(x^{k,h}, \xi_m^{k,h})$ to the server
      end if
    else
      $g_m^{k,h+1} \leftarrow g_m^{k,h}$
    end if
  end for
  if client m is available for the epoch-boundary transmission **then**
    Send g_m^{k,H^k} to the server
  end if
end for

rogate weights, see Figure 7. The server and clients are simulated on the same machine using the multiprocessing setup from Appendix A. The surrogate-weight optimization takes approximately 4.3 seconds per round on average, compared with 7.6 seconds for the slowest selected client, and remains faster in every measured round. Thus, the additional surrogate-side validation fits within the client-computation window in this setup.

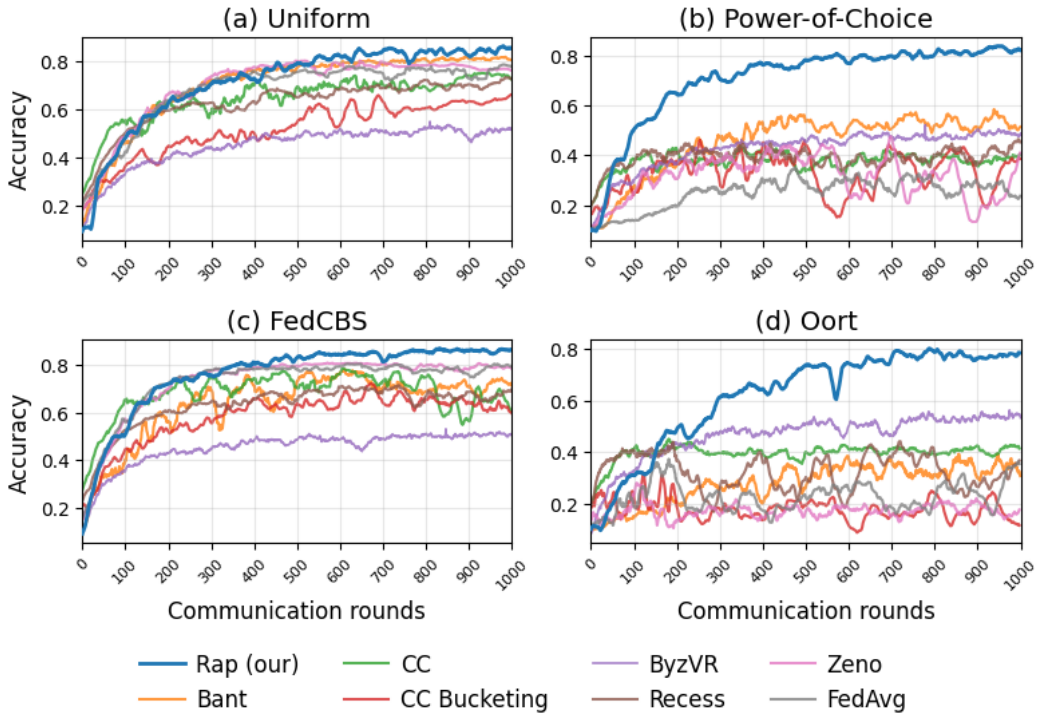


Figure 4: Performance comparison across client-selection strategies under the Label Flip attack.

E Sensitivity to the Trusted Reference-Set Size

RAP uses a finite trusted reference set to construct the trial function for Byzantine validation. We study the sensitivity to its size by varying $N \in \{50, 100, 250, 500\}$ on CIFAR-10 while keeping the remaining training configuration fixed. We use FEDCBS client selection and the SIGN FLIP attack, providing a non-uniform client-selection setting with Byzantine corruption.

Figure 8 shows the resulting test-accuracy trajectories. For $N \in \{100, 250, 500\}$, the curves remain close after the initial training phase and reach similar accuracy levels. Reducing the trusted set from 500 to 100 samples therefore has only a modest empirical effect in this setting. In contrast, $N = 50$ produces substantially larger round-to-round fluctuations and more frequent accuracy drops.

Overall, the ablation suggests that RAP does not require a large trusted server dataset to obtain stable training, while very small reference sets can make the empirical trial objective less reliable.

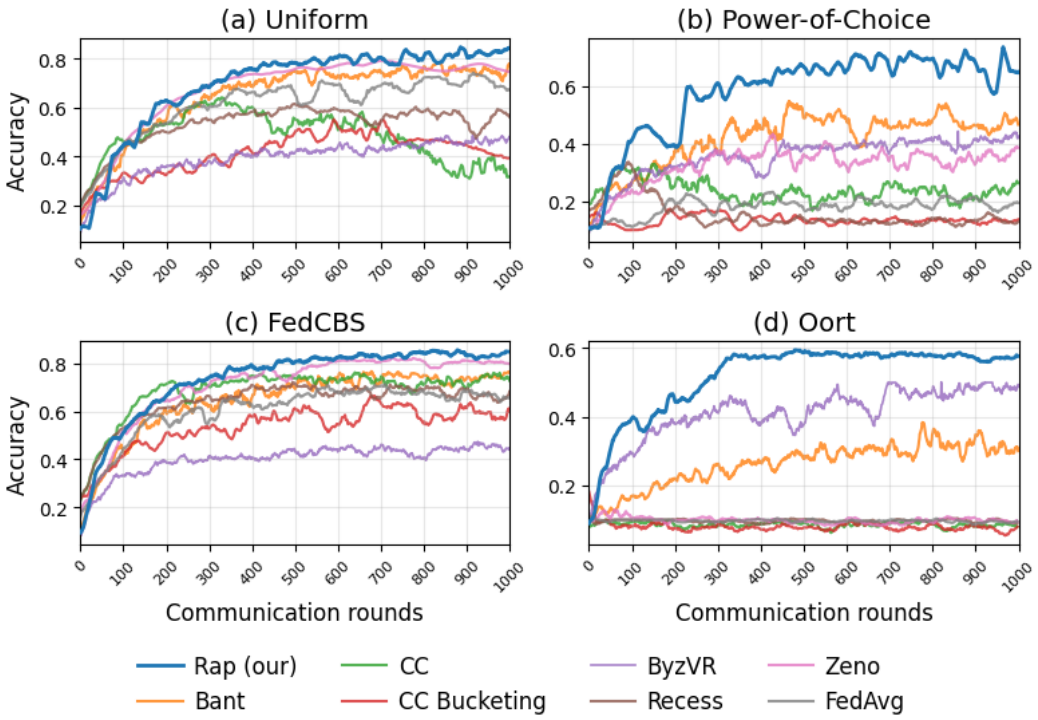


Figure 5: Performance comparison across client-selection strategies under the Sign Flip attack.

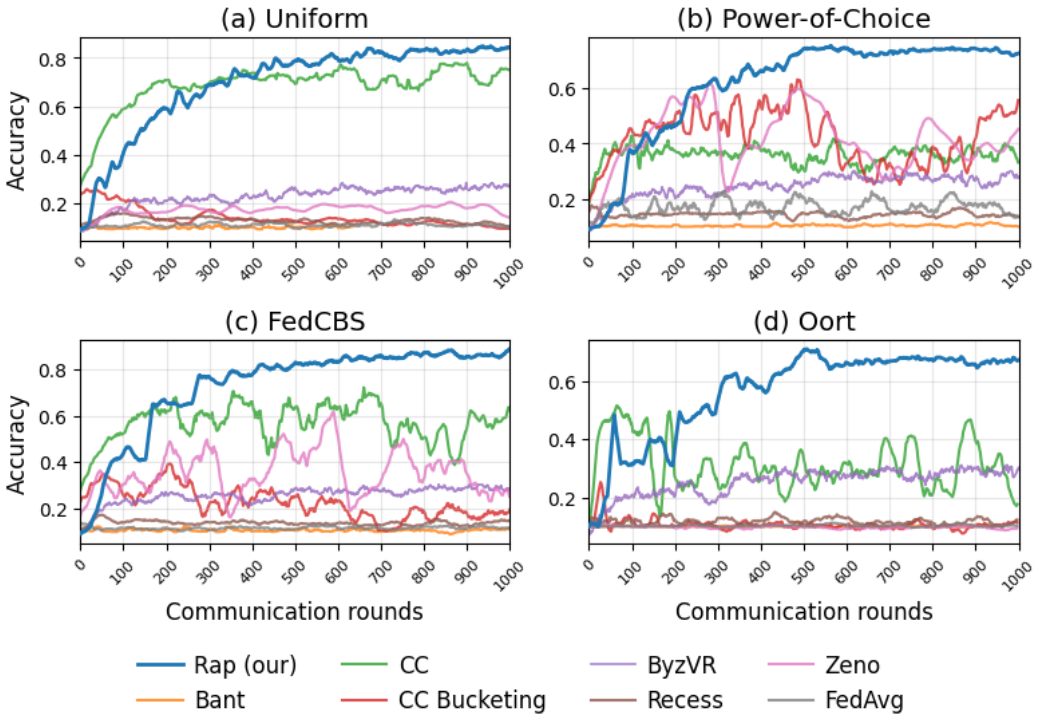


Figure 6: Performance comparison across client-selection strategies under the ALIE attack.

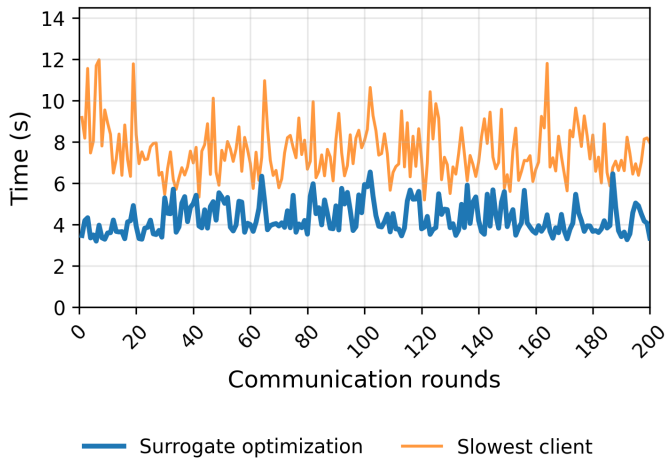


Figure 7: Per-round timing on CIFAR-10. Surrogate-weight optimization is consistently faster than the slowest selected client and can be overlapped with client computation.

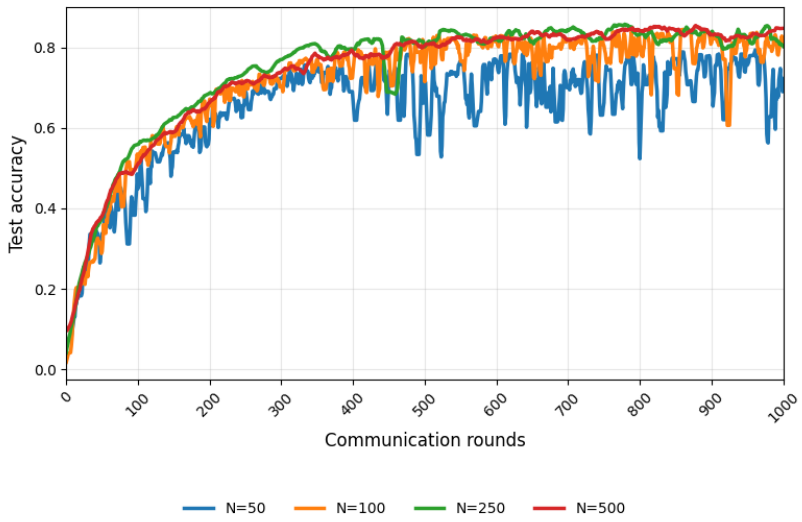


Figure 8: Sensitivity of RAP to the trusted reference-set size N on CIFAR-10 under FEDCBS client selection and the SIGN FLIP attack. We vary $N \in \{50, 100, 250, 500\}$ while keeping all other training parameters fixed. The trajectories for $N \geq 100$ remain close, whereas $N = 50$ exhibits noticeably higher variability and less stable convergence.

F THEORETICAL ANALYSIS.

This appendix introduces the notation and provides the convergence proof for RAP.

F.1 NOTATION.

To aid in understanding the proofs presented in the appendix and to simplify engagement with all formulas throughout this paper, we provide a comprehensive list of the notation used in this study in the form of the following table.

Table 17: Notation Reference.

Symbol	Description	Symbol	Description
d	Dimensionality of the parameter space $\mathbb{R}^d$	$x^\star$	Optimal solution of the objective function
M	Total number of clients	C	Number of clients participating in one round
G, B	Numbers of honest and Byzantine clients	$\mathcal{G}, \mathcal{B}$	Sets of honest and Byzantine clients
f	Global objective over honest clients	f_m	Local objective function of client m
$\mathcal{D}_m$	Data distribution of client m	ξ_m	Data sample of client m
$\widehat{f}$	Trial function on the server	f_1	Server-side objective used to construct $\widehat{f}$
N	Number of samples in the trial function	$\zeta(N)$	Trial-function approximation error
Δ	Initial trial-function gap	$w^{k,h}$	Weights assigned to surrogate gradients
Δ_M^1	M-dimensional probability simplex	$\|\cdot\|$	Euclidean norm unless specified otherwise
k, h	Current epoch and iteration, respectively	H^k	Number of iterations in epoch k
K	Total number of training epochs	p	Parameter of the geometric distribution
$\mathcal{G}(k,h)$	Set of honest clients selected at iteration (k,h)	$G(k,h), \widetilde{G}$	Number of selected honest clients and its minimum
$\widehat{\pi}^{k,h}, \widetilde{\pi}^{k,h}$	Client-selection vector and fresh-gradient weights	$g_m^{k,h}$	Surrogate gradient of client m
q_m	Availability probability of client m	$\eta_m^{k,h}, \psi_m^k$	Iteration-level and epoch-level availability indicators
$\nabla f_m(x^{k,h}, \xi_m^{k,h})$	Stochastic gradient of client m	$\nabla f_m(x^{k,h})$	Expected gradient of client m
γ	Learning rate (stepsize)	θ	Mixing parameter
L	Smoothness constant of $\widehat{f}$	σ^2	Variance bound for stochastic gradients
δ_1, δ_2	Multiplicative and additive heterogeneity parameters	ε^2	Average stationarity measure

F.2 General Inequalities and Lemmas.

General inequalities. Suppose $x, y, \{a_i\}_{i=1}^n \in \mathbb{R}^d$, $w \in \{w \in \mathbb{R}_+^n : \sum_{i=1}^n w_i = 1\}$, p - positive scalar, $f(\cdot)$ satisfies the Assumption 3.1, $\phi(\cdot)$ is under Assumption 3.2. Then,

$$f(x) \le f(y) + \langle \nabla f(y), x - y\rangle + \frac{L}{2}\|x-y\|^2, \tag{Lip}$$

$$|\langle x, y\rangle| \le \frac{\|x\|^2}{2p} + \frac{p\|y\|^2}{2}, \tag{Young}$$

$$\|x+y\|^2 = \|x\|^2 + \|y\|^2 + 2\langle x, y\rangle, \tag{Norm}$$

$$\left\|\sum_{i=1}^n x_i\right\|^2 \le n\sum_{i=1}^n \|x_i\|^2, \tag{CS}$$

$$\phi\left(\frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}\right) \le \frac{\sum_{i=1}^n w_i \phi(x_i)}{\sum_{i=1}^n w_i}. \tag{Jen}$$

Lemmas. One of the core components of our algorithm is the trial function. Recall that it is constructed from a subset of the data stored on the server. To enable the main theoretical analysis,

we prove a lemma that bounds the deviation between the trial function $\hat{f}(x)$ and the server objective $f_1(x)$.

Proposition F.1. *Suppose Assumption 3.1 holds. Then for all $x \in \mathcal{X} \subset \mathbb{R}^d$ with probability at least $1 - \tilde{\delta}$ over a sample of size N, the following estimate linking the trial function gradient to the objective function gradient on the server is valid:*

$$\|\nabla f_1(x) - \nabla \hat{f}(x)\|_2^2 \leq \zeta(N) = \tilde{O}\left(\frac{1}{N}\right).$$

Proof. By the norm equivalence $\|\cdot\|_2 \leq \sqrt{d}\|\cdot\|_\infty$, we obtain the chain:

$$\|\nabla f_1(x) - \nabla \hat{f}(x)\|_2^2 \leq d\|\nabla f_1(x) - \nabla \hat{f}(x)\|_\infty^2 \leq d \cdot \tilde{O}\left(\frac{1}{N}\right) = \tilde{O}\left(\frac{1}{N}\right),$$

where the second inequality follows from uniform convergence via Theorem 5 Shalev-Shwartz et al. (2009), leveraging the l_∞-covering number bound $N(\varepsilon, \mathcal{X}, d_2) = O\left(d^2 \left(\frac{\varepsilon}{\delta}\right)^d\right)$ for $\mathcal{X}$ within an l_2-sphere of radius S and the L-smoothness property $\|\nabla f_1(x_i;\cdot) - \nabla f_1(x_j;\cdot)\|_\infty \leq L\|x_i - x_j\|_2$. Substituting the derived $\|\nabla f_1(x) - \nabla \hat{f}(x)\|_\infty = \tilde{O}\left(\frac{1}{\sqrt{N}}\right)$ into the norm equivalence and defining $\zeta(N) \overset{def}{=} \tilde{O}\left(\frac{1}{N}\right)$ completes the proof. □

We now prove an auxiliary statement that is standard in convergence analyses within the near-stationary point finding paradigm.

Proposition F.2. *Suppose Assumptions 3.3 and 3.1 hold. Then for all $x \in \mathbb{R}^d$ the following estimate is valid:*

$$\begin{aligned}
&-\gamma\left\langle \nabla \hat{f}(x^{k,h}), \frac{1}{G(k,h)} \sum_{m \in \mathcal{G}(k,h)} \nabla f_m(x^{k,h}) \right\rangle \\
&\leq -\frac{\gamma}{2}\left\|\nabla f(x^{k,h})\right\|^2 + \gamma\zeta(N) \\
&\quad + \frac{3\gamma}{2}\left(\delta_1\left\|\nabla f(x^{k,h})\right\|^2 + \delta_2\right),
\end{aligned}$$

here $\zeta(N) = \mathcal{O}\left(\frac{1}{N}\right)$, $\quad N$ – amount of samples in $\hat{f}$.

Proof. We begin with following estimate:

$$\begin{aligned}
-\gamma\left\langle \nabla \hat{f}(x), \frac{1}{G(k,h)} \sum_{i \in \mathcal{G}(k,h)} \nabla f_i(x) \right\rangle &= \gamma\left\langle \nabla f(x) - \nabla \hat{f}(x), \frac{1}{G(k,h)} \sum_{i \in \mathcal{G}(k,h)} \nabla f_i(x) \right\rangle \\
&\quad - \gamma\left\langle \nabla f(x), \frac{1}{G(k,h)} \sum_{i \in \mathcal{G}(k,h)} \nabla f_i(x) \right\rangle. \qquad (1)
\end{aligned}$$

Next, we continue with further manipulations on the first term:

$$\begin{aligned}
&\gamma\left\langle \nabla f(x) - \nabla \hat{f}(x), \frac{1}{G(k,h)} \sum_{i \in \mathcal{G}(k,h)} \nabla f_i(x) \right\rangle \\
&\overset{\text{(Young)}}{\leq} \frac{\gamma}{2}\left\|\nabla f(x) - \nabla \hat{f}(x)\right\|^2 + \frac{\gamma}{2}\left\|\frac{1}{G(k,h)} \sum_{i \in \mathcal{G}(k,h)} \nabla f_1(x)\right\|^2 \\
&\overset{\text{(Young)}}{\leq} \frac{\gamma}{2}\left(\|\nabla f(x) - \nabla f_1(x)\|^2 \left\|\nabla f_1(x) - \nabla \hat{f}(x)\right\|^2\right)
\end{aligned}$$

$$+\frac{\gamma}{2}\left\|\frac{1}{G(k,h)}\sum_{i\in\mathcal{G}(k,h)}\nabla f_i(x)\right\|^2$$
$$\overset{\text{(Lemma } F.1)}{\leq} \gamma\left(\zeta(N)+\delta_1\|\nabla f(x)\|^2+\delta_2\right)$$
$$+\frac{\gamma}{2}\left\|\frac{1}{G(k,h)}\sum_{i\in\mathcal{G}(k,h)}\nabla f_i(x)\right\|^2. \tag{2}$$

Now we evaluate the second term:

$$-\gamma\left\langle\nabla f(x),\frac{1}{G(k,h)}\sum_{i\in\mathcal{G}(k,h)}\nabla f_i(x)\right\rangle \overset{\text{(Norm)}}{=} -\frac{\gamma}{2}\|\nabla f(x)\|^2-\frac{\gamma}{2}\left\|\frac{1}{G(k,h)}\sum_{i\in\mathcal{G}(k,h)}\nabla f_i(x)\right\|^2$$
$$+\frac{\gamma}{2}\left\|\frac{1}{G(k,h)}\sum_{i\in\mathcal{G}(k,h)}(\nabla f_i(x)-\nabla f(x))\right\|^2$$
$$\overset{\text{(CS)}}{\leq} -\frac{\gamma}{2}\|\nabla f(x)\|^2-\frac{\gamma}{2}\left\|\frac{1}{G(k,h)}\sum_{i\in\mathcal{G}(k,h)}\nabla f_i(x)\right\|^2$$
$$+\frac{\gamma}{2G(k,h)}\sum_{i\in\mathcal{G}(k,h)}\|(\nabla f_i(x)-\nabla f(x))\|^2$$
$$\overset{\text{(Ass. 3.3)}}{\leq} -\frac{\gamma}{2}\|\nabla f(x)\|^2-\frac{\gamma}{2}\left\|\frac{1}{G(k,h)}\sum_{i\in\mathcal{G}(k,h)}\nabla f_i(x)\right\|^2$$
$$+\frac{\gamma}{2}\left(\delta_1\|\nabla f(x)\|^2+\delta_2\right). \tag{3}$$

Substituting (2) and (3) into (1) completes the proof:

$$-\gamma\left\langle\nabla\hat{f}(x),\frac{1}{G(k,h)}\sum_{i\in\mathcal{G}(k,h)}\nabla f_i(x)\right\rangle$$
$$\leq -\frac{\gamma}{2}\left\|\nabla f(x^{k,h})\right\|^2+\gamma\zeta(N)$$
$$+\frac{3\gamma}{2}\left(\delta_1\left\|\nabla f(x^{k,h})\right\|^2+\delta_2\right).$$

□

Since, algorithmically, we model the epoch length as a random variable drawn from a geometric distribution, we state the following auxiliary result.

Proposition F.3. *[(Allen-Zhu, 2018)] Given sequence $D_0, D_1, \dots D_n \in \mathbb{R}$, where $N \sim Geom(p)$. Then, $\mathbb{E}_N[D_{N-1}] = pD_0 + (1-p)\mathbb{E}_N[D_N]$.*

F.3 Proof of convergence of Algorithm 1

Proposition F.4. *Under Assumptions 3.3 and 3.4, the availability-aware client procedure in Algorithm 2 satisfies*

$$\mathbb{E}_{H^k}\mathbb{E}_{\psi_m^k}\mathbb{E}_{\xi_m^{k,0}}\mathbb{E}_{\eta_m^{k,0}}\dots\mathbb{E}_{\xi_m^{k,H^k-1}}\mathbb{E}_{\eta_m^{k,H^k-1}}\left\|g^{k,H^k}\right\|^2 \leq \frac{504(1+\delta_1)}{p^2\min_{m\in\mathcal{G}}^2 q_m}\mathbb{E}_{H^k}\left\|\nabla f(x^{k,H^k})\right\|^2$$
$$+\frac{1008(\sigma^2+\delta_2)}{p^2\min_{m\in\mathcal{G}}^2 q_m},$$

where $g^{k,h}=\sum_{m\in\mathcal{G}(k,h)}\frac{\psi_m^k}{q_m}g_m^{k,h}$.

Proof. Let us start with following estimate:

$$\begin{aligned}\left\|g^{k,h+1}\right\|^2 &= \left\|g^{k,h} + \sum_{m\in\mathcal{G}(k,h)} \frac{\psi_m^k}{q_m}\frac{\eta_m^{k,h}}{q_m}\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2 \\ &\overset{}{\leq} (Young)\,(1+c)\left\|g^{k,h}\right\|^2 \\ &\quad + \left(1+\frac{1}{c}\right)\frac{1}{M^2}\left\|\sum_{m\in G}\frac{\psi_m^k}{q_m}\frac{\eta_m^{k,h}}{q_m}\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2. \end{aligned} \tag{1}$$

Let's estimate the second term:

$$\begin{aligned}&\left\|\sum_{m\in\mathcal{G}(k,h)}\frac{\psi_m^k}{q_m}\frac{\eta_m^{k,h}}{q_m}\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2 \\ &\overset{(CS)}{\leq} 4\left\|\sum_{m\in\mathcal{G}(k,h)}\left(\frac{\eta_m^{k,h}}{q_m}-1\right)\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2 \\ &\quad + 4\left\|\sum_{m\in\mathcal{G}(k,h)}\left(\frac{\psi_m^k}{q_m}-1\right)\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2 \\ &\quad + 4\left\|\sum_{m\in\mathcal{G}(k,h)}\left(\frac{\psi_m^k}{q_m}-1\right)\left(\frac{\eta_m^{k,h}}{q_m}-1\right)\right. \\ &\qquad\qquad \left.\cdot\,\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2 \\ &\quad + 16\left\|\sum_{m\in\mathcal{G}(k,h)}\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2. \end{aligned} \tag{2}$$

Evaluating the terms one at a time. Let's begin with the last one.

$$\begin{aligned}\left\|\sum_{m\in\mathcal{G}(k,h)}\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2 &\overset{(CS)}{\leq} 3\left\|\sum_{m\in\mathcal{G}(k,h)}\nabla f(x^{k,h})\right\|^2 \\ &\quad + 3\left\|\sum_{m\in\mathcal{G}(k,h)}\left(\nabla f_m(x^{k,h})-\nabla f(x^{k,h})\right)\right\|^2 \\ &\quad + 3\left\|\sum_{m\in\mathcal{G}(k,h)}\left(\nabla f_m(x^{k,h})-\nabla f_m(x^{k,h},\xi_m^{k,h})\right)\right\|^2. \end{aligned}$$

Taking expectation over $\xi_m^{k,h}$, we have:

$$\mathbb{E}_{\xi_m^{k,h}}\left\|\sum_{m\in\mathcal{G}(k,h)}\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2 \leq 3M^2\left(\sigma^2+\delta_2+(1+\delta_1)\left\|\nabla f(x^{k,h})\right\|^2\right). \tag{3}$$

For the first and the third terms we note that $\eta_m^{k,h}\sim \text{Bernoulli}(q_m)$ and

$$\mathbb{E}_{\eta_m^{k,h}}\left(\frac{\eta_m^{k,h}}{q_m}-1\right)^2 = \frac{\mathbb{E}_{\eta_m^{k,h}}\left(\eta_m^{k,h}-q_m\right)^2}{q_m^2} \leq \frac{\sigma_{\eta_m}^2}{q_m^2} = \frac{1-q_m}{q_m} \leq \frac{1}{q_m}.$$

In that way, denoting $A = \sum_{m\in\mathcal{G}(k,h)} \left(\frac{\eta_m^{k,h}}{q_m} - 1\right) \nabla f_m(x^{k,h}, \xi_m^{k,h})$, we have

$$\begin{aligned}\mathbb{E}_{\eta_m^{k,h}} \|A\|^2 &\leq \sum_{m\in\mathcal{G}(k,h)} \mathbb{E}_{\eta_m^{k,h}} \left(\frac{\eta_m^{k,h}}{q_m} - 1\right)^2 \sum_{m\in\mathcal{G}(k,h)} \left\|\nabla f_m(x^{k,h}, \xi_m^{k,h})\right\|^2 \\ &\leq \frac{M}{\min_{m\in\mathcal{G}} q_m} \sum_{m\in\mathcal{G}(k,h)} \left\|\nabla f_m(x^{k,h}, \xi_m^{k,h})\right\|^2.\end{aligned}$$

Taking expectation over $\xi_m^{k,h}$, for the first term we have:

$$\mathbb{E}_{\xi_m^{k,h}} \mathbb{E}_{\eta_m^{k,h}} \|A\|^2 \leq \frac{3M^2(\delta_1+1)\left\|\nabla f\left(x^{k,h}\right)\right\|^2}{\min_{m\in\mathcal{G}} q_m} + \frac{3M^2(\sigma^2+\delta_2)}{\min_{m\in\mathcal{G}} q_m}. \tag{4}$$

For the second term we have:

$$\begin{aligned}&\mathbb{E}_{\xi_m^{k,h}} \mathbb{E}_{\eta_m^{k,h}} \left\| \sum_{m\in\mathcal{G}(k,h)} \left(\frac{\psi_m^k}{q_m} - 1\right) \nabla f_m(x^{k,h}, \xi_m^{k,h}) \right\|^2 \\ &\leq 3M \left(\sigma^2 + \delta_2 + (1+\delta_1)\left\|\nabla f(x^{k,h})\right\|^2\right) \sum_{m\in\mathcal{G}(k,h)} \left(\frac{\psi_m^k}{q_m} - 1\right)^2.\end{aligned} \tag{5}$$

And for the third term:

$$\begin{aligned}&\mathbb{E}_{\xi_m^{k,h}} \mathbb{E}_{\eta_m^{k,h}} \left\| \sum_{m\in\mathcal{G}(k,h)} \left(\frac{\psi_m^k}{q_m} - 1\right)\left(\frac{\eta_m^{k,h}}{q_m} - 1\right) \nabla f_m(x^{k,h}, \xi_m^{k,h}) \right\|^2 \\ &\leq \left(\frac{3M(\delta_1+1)\left\|\nabla f\left(x^{k,h}\right)\right\|^2}{\min_{m\in\mathcal{G}} q_m} + \frac{3M(\sigma^2+\delta_2)}{\min_{m\in\mathcal{G}} q_m}\right) \sum_{m\in\mathcal{G}(k,h)} \left(\frac{\psi_m^k}{q_m} - 1\right)^2.\end{aligned} \tag{6}$$

Substituting (3), (4), (5) and (6) into (2):

$$\begin{aligned}&\mathbb{E}_{\xi_m^{k,h}} \mathbb{E}_{\eta_m^{k,h}} \left\| \sum_{m\in\mathcal{G}(k,h)} \frac{\eta_m^{k,h}}{q_m} \nabla f_m(x^{k,h}, \xi_m^{k,h}) \right\|^2 \\ &\leq \frac{12(\delta_1+1)M}{\min_{m\in\mathcal{G}} q_m} \left(5M + 2 \sum_{m\in\mathcal{G}(k,h)} \left(\frac{\psi_m^k}{q_m} - 1\right)^2\right) \\ &\quad \cdot \left\|\nabla f(x^{k,h})\right\|^2 \\ &\quad + \frac{12(\sigma^2+\delta_2)M}{\min_{m\in\mathcal{G}} q_m} \left(5M + 2 \sum_{m\in\mathcal{G}(k,h)} \left(\frac{\psi_m^k}{q_m} - 1\right)^2\right).\end{aligned} \tag{7}$$

Let‘s substitute (7) into (1):

$$\begin{aligned}\mathbb{E}_{\xi_m^{k,h}} \mathbb{E}_{\eta_m^{k,h}} \left\|g^{k,h+1}\right\|^2 &\leq (1+c)\left\|g^{k,h}\right\| \\ &+ \left(1 + \frac{1}{c}\right) \frac{12(\delta_1+1)}{M \min_{m\in\mathcal{G}} q_m} \left(5M + 2 \sum_{m\in\mathcal{G}(k,h)} \left(\frac{\psi_m^k}{q_m} - 1\right)^2\right) \\ &\quad \cdot \left\|\nabla f(x^{k,h})\right\|^2\end{aligned}$$

$$+\left(1+\frac{1}{c}\right)\frac{12(\sigma^2+\delta_2)}{M\min_{m\in\mathcal{G}}q_m}\left(5M+2\sum_{m\in\mathcal{G}(k,h)}\left(\frac{\psi_m^k}{q_m}-1\right)^2\right).$$

Enrolling a recursion, we get:

$$\begin{aligned}
&\mathbb{E}_{\xi_m^{k,0}}\mathbb{E}_{\eta_m^{k,0}}\ldots\mathbb{E}_{\xi_m^{k,h}}\mathbb{E}_{\eta_m^{k,h}}\left\|g^{k,h+1}\right\|^2\\
&\leq\left(1+\frac{1}{c}\right)\frac{12(\delta_1+1)}{M\min_{m\in\mathcal{G}}q_m}\left(5M+2\sum_{m\in\mathcal{G}(k,h)}\left(\frac{\psi_m^k}{q_m}-1\right)^2\right)\\
&\quad\cdot\sum_{i=0}^{h-1}(1+c)^{h-i-1}\left\|\nabla f(x^{k,i})\right\|^2\\
&\quad+\left(1+\frac{1}{c}\right)\frac{12(\sigma^2+\delta_2)}{M\min_{m\in\mathcal{G}}q_m}\left(5M+2\sum_{m\in\mathcal{G}(k,h)}\left(\frac{\psi_m^k}{q_m}-1\right)^2\right)\\
&\quad\cdot\sum_{i=0}^{h-1}(1+c)^{h-i-1}.
\end{aligned}$$

Now we use that $H^k\sim Geom(p)$ and $\psi_m^k\sim\text{Bernoulli}(q_m)$:

$$\begin{aligned}
&\mathbb{E}_{H^k}\mathbb{E}_{\psi_m^k}\mathbb{E}_{\xi_m^{k,0}}\mathbb{E}_{\eta_m^{k,0}}\ldots\mathbb{E}_{\xi_m^{k,H^k-1}}\mathbb{E}_{\eta_m^{k,H^k-1}}\left\|g^{k,H^k}\right\|^2\\
&=\sum_{j\geq1}p(1-p)^j\mathbb{E}_{\psi_m^k}\mathbb{E}_{\xi_m^{k,0}}\mathbb{E}_{\eta_m^{k,0}}\ldots\\
&\qquad\mathbb{E}_{\xi_m^{k,H^k-1}}\mathbb{E}_{\eta_m^{k,H^k-1}}\left\|g^{k,j}\right\|^2\\
&\leq\left(1+\frac{1}{c}\right)\frac{84(\delta_1+1)}{\min_{m\in\mathcal{G}}^2q_m}\\
&\qquad\cdot\sum_{j\geq1}p(1-p)^j\sum_{i=0}^{j-1}(1+c)^{j-i-1}\left\|\nabla f(x^{k,i})\right\|^2\\
&\quad+\left(1+\frac{1}{c}\right)\frac{84(\delta_2+\sigma^2)}{\min_{m\in\mathcal{G}}^2q_m}\\
&\qquad\cdot\sum_{j\geq1}p(1-p)^j\sum_{i=0}^{j-1}(1+c)^{j-i-1}.
\end{aligned}\tag{6}$$

It remains to evaluate the two geometric sums in (6). Set $c=p/2$. Reordering the first double sum, we obtain

$$\begin{aligned}
&\sum_{j\geq1}p(1-p)^j\sum_{i=0}^{j-1}(1+c)^{j-i-1}\left\|\nabla f(x^{k,i})\right\|^2\\
&\leq\sum_{t\geq0}(1-p)^t\left(1+\frac{p}{2}\right)^t\sum_{j\geq0}p(1-p)^{j+1}\left\|\nabla f(x^{k,j})\right\|^2\\
&\leq\frac{1}{1-(1-p)(1+p/2)}\sum_{j\geq0}p(1-p)^j\left\|\nabla f(x^{k,j})\right\|^2\\
&=\frac{2}{p(1+p)}\mathbb{E}_{H^k}\left\|\nabla f(x^{k,H^k})\right\|^2\leq\frac{2}{p}\mathbb{E}_{H^k}\left\|\nabla f(x^{k,H^k})\right\|^2.
\end{aligned}\tag{7}$$

Similarly,

$$\sum_{j\geq1}p(1-p)^j\sum_{i=0}^{j-1}(1+c)^{j-i-1}\leq p\sum_{j\geq0}j\left(1-\frac{p}{2}\right)^j\leq\frac{4}{p}.\tag{8}$$

Substituting (7) and (8) into (6), gives

$$\mathbb{E}_{H^k}\mathbb{E}_{\psi_m^k}\mathbb{E}_{\xi_m^{k,0}}\mathbb{E}_{\eta_m^{k,0}}\ldots\mathbb{E}_{\xi_m^{k,H^{k-1}}}\mathbb{E}_{\eta_m^{k,H^{k-1}}}\left\|g^{k,H^k}\right\|^2$$
$$\leq \frac{504(1+\delta_1)}{p^2\min_{m\in\mathcal{G}}^2 q_m}\mathbb{E}_{H^k}\left\|\nabla f(x^{k,H^k})\right\|^2 + \frac{1008(\sigma^2+\delta_2)}{p^2\min_{m\in\mathcal{G}}^2 q_m}.$$

□

Theorem F.5. *Suppose Assumptions 3.3, 3.2, 3.1 and 3.4 hold. Then for Algorithm 1 with $\delta_1 \leq \frac{1}{6}$, $\gamma \leq \frac{p\min_{m\in\mathcal{G}} q_m}{40L}$ and $\theta \leq \frac{\gamma L p^2 \min_{m\in\mathcal{G}} q_m}{598}$ it implies that*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla f(x^{k,0})\right\|^2 \leq \frac{8\left(\hat{f}(x^{0,0})-\hat{f}(x^*)\right)}{\gamma K} + \frac{32L\gamma(\sigma^2+\delta_2)}{p\min_{m\in\mathcal{G}} q_m} + \frac{14(\sigma^2+\delta_2)\gamma L}{\widetilde{G}^2 p\min_{m\in\mathcal{G}} q_m}$$
$$+ \frac{\delta_2\gamma L}{75p} + \frac{8(1-\theta)\zeta(N)}{p}.$$

Proof. We begin with following estimate:

$$\begin{aligned}
\hat{f}(x^{k,h+1}) &= \hat{f}\Bigg(x^{k,h} - \gamma(1-\theta)\sum_{m=1}^{M}\tilde{\pi}_m^k \mathbb{I}_{\hat{\pi}_m^{k,h}\neq 0}\frac{\eta_m^{k,h}}{q_m}\nabla f_m(x^{k,h},\xi_m^{k,h}) \\
&\quad - \gamma\theta\sum_{m=1}^{M} w_m^{k,h}\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\Bigg) \\
&\overset{(Jen)}{\leq} (1-\theta)\hat{f}\Bigg(x^{k,h} - \gamma\sum_{m=1}^{M}\tilde{\pi}_m^k\mathbb{I}_{\hat{\pi}_m^{k,h}\neq 0}\frac{\eta_m^{k,h}}{q_m} \\
&\qquad\qquad \cdot\nabla f_m(x^{k,h},\xi_m^{k,h})\Bigg) \\
&\quad + \theta\hat{f}\Bigg(x^{k,h} - \gamma\sum_{m=1}^{M} w_m^{k,h}\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\Bigg) \\
&\overset{(i)}{\leq} (1-\theta)\hat{f}\Bigg(x^{k,h} - \frac{\gamma}{G(k,h)}\sum_{m\in\mathcal{G}(k,h)}\frac{\eta_m^{k,h}}{q_m} \\
&\qquad\qquad \cdot\nabla f_m(x^{k,h},\xi_m^{k,h})\Bigg) \\
&\quad + \theta\hat{f}\Bigg(x^{k,h} - \frac{\gamma}{G(k-1,H^{k-1})}\sum_{m\in\mathcal{G}(k-1,H^{k-1})} \\
&\qquad\qquad \cdot\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\Bigg).
\end{aligned} \tag{1}$$

Step (i) is valid because $w^{k,h}$ and $\tilde{\pi}^{k,h}$ are the minimizers of their respective expressions. Moreover, $\mathcal{G}(k,h)$ denotes the set of good workers selected by the client selection strategy at iteration (k,h), and thus $\tilde{\pi}_m^{k,h} \neq 0$ for all $m \in \mathcal{G}(k,h)$.

Let's evaluate the terms one at a time. For the first term we use (Lip) inequation.

$$\hat{f}\left(x^{k,h} - \frac{\gamma}{G(k,h)}\sum_{m\in\mathcal{G}(k,h)}\frac{\eta_m^{k,h}}{q_m}\nabla f(x^{k,h},\xi_m^{k,h})\right)$$
$$\overset{(Lip)}{\leq} \hat{f}(x^{k,h})$$

$$+\left\langle \nabla\hat{f}(x^{k,h}), -\frac{\gamma}{G(k,h)}\sum_{m\in\mathcal{G}(k,h)}\frac{\eta_m^{k,h}}{q_m}\nabla f_m(x^{k,h},\xi_m^{k,h})\right\rangle$$
$$+\frac{L\gamma^2}{2}\left\|\frac{1}{G(k,h)}\sum_{m\in\mathcal{G}(k,h)}\frac{\eta_m^{k,h}}{q_m}\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2.$$

Taking expectation over $\xi_m^{k,h}$ and $\eta_m^{k,h}$ and using $\mathbb{E}[\eta_m^{k,h}/q_m] = 1$ and the unbiasedness of the stochastic gradients, the linear term becomes

$$\mathbb{E}_{\xi_m^{k,h}}\mathbb{E}_{\eta_m^{k,h}}\left\langle \nabla\hat{f}(x^{k,h}), -\frac{\gamma}{G(k,h)}\sum_{m\in\mathcal{G}(k,h)}\frac{\eta_m^{k,h}}{q_m}\nabla f_m(x^{k,h},\xi_m^{k,h})\right\rangle$$
$$\overset{\text{Ass. 3.4}}{=}\left\langle \nabla\hat{f}(x^{k,h}), -\frac{\gamma}{G(k,h)}\sum_{m\in\mathcal{G}(k,h)}\nabla f_m(x^{k,h})\right\rangle$$
$$\overset{\text{Lemma } F.2}{\leq} -\frac{\gamma}{2}\left\|\nabla f(x^{k,h})\right\|^2+\gamma\zeta(N)+\frac{3\gamma}{2}\left(\delta_1\left\|\nabla f(x^{k,h})\right\|^2+\delta_2\right).$$

And for the third term we have:

$$\mathbb{E}_{\xi_m^{k,h}}\mathbb{E}_{\eta_m^{k,h}}\left\|\frac{1}{G(k,h)}\sum_{m\in\mathcal{G}(k,h)}\frac{\eta_m^{k,h}}{q_m}\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2$$
$$\overset{\text{CS}}{\leq} 2\mathbb{E}_{\xi_m^{k,h}}\mathbb{E}_{\eta_m^{k,h}}\frac{1}{G^2(k,h)}\sum_{m\in\mathcal{G}(k,h)}\left(\frac{\eta_m^{k,h}}{q_m}-1\right)^2\sum_{m\in\mathcal{G}(k,h)}\left\|\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2$$
$$+2\mathbb{E}_{\xi_m^{k,h}}\frac{1}{G^2(k,h)}\left\|\sum_{m\in\mathcal{G}(k,h)}\nabla f_m(x^{k,h},\xi_m^{k,h})\right\|^2$$
$$\overset{CS,Ass.3.3,Ass.3.4}{\leq}\frac{16(\delta_1+1)}{\min_{m\in\mathcal{G}}q_m}\left\|\nabla f(x^{k,h})\right\|^2+\frac{8(\sigma^2+\delta_2)}{\min_{m\in\mathcal{G}}q_m}.$$

In this way,

$$\mathbb{E}_{\xi_m^{k,h}}\mathbb{E}_{\eta_m^{k,h}}\hat{f}\left(x^{k,h}-\frac{\gamma}{G(k,h)}\sum_{m\in\mathcal{G}(k,h)}\frac{\eta_m^{k,h}}{q_m}\nabla f(x^{k,h})\right)$$
$$\leq\hat{f}(x^{k,h})$$
$$-\left(\frac{\gamma}{2}-\frac{3\gamma\delta_1}{2}-\frac{8L\gamma^2(\delta_1+1)}{\min_{m\in\mathcal{G}}q_m}\right)\left\|\nabla f(x^{k,h})\right\|^2$$
$$+\gamma\zeta(N)+\frac{4L\gamma^2(\sigma^2+\delta_2)}{\min_{m\in\mathcal{G}}q_m}+\frac{3\gamma\delta_2}{2}. \tag{2}$$

Evaluating the second term:

$$\hat{f}\left(x^{k,h}-\frac{\gamma}{G(k-1,H^{k-1})}\sum_{m\in\mathcal{G}(k-1,H^{k-1})}\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\right)$$
$$\overset{Lip}{\leq}\hat{f}(x^{k,h})-\gamma\Bigg\langle\nabla\hat{f}(x^{k,h}),\frac{1}{G(k-1,H^{k-1})}\sum_{m\in\mathcal{G}(k-1,H^{k-1})}$$
$$\cdot\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\Bigg\rangle$$

$$
\begin{aligned}
&+ \frac{L\gamma^2}{G^2(k-1,H^{k-1})}\left\|\sum_{m\in\mathcal{G}(k-1,H^{k-1})}\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\right\|^2\\
&\overset{(Young),\,\text{Ass.3.3},\,CS}{\leq} \hat{f}(x^{k,h}) + \frac{2\gamma(\delta_1+1)\|\nabla f(x^{k,h})\|^2 + 2\delta_2\gamma}{2}\\
&+ \frac{(L\gamma+1)\gamma}{2\widetilde{G}^2}\left\|\sum_{m\in\mathcal{G}(k-1,H^{k-1})}\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\right\|^2\\
&\leq \hat{f}(x^{k,h}) + \gamma(\delta_1+1)\|\nabla f(x^{k,h})\|^2 + \gamma\delta_2 \qquad (3)\\
&+ \frac{\gamma}{\widetilde{G}^2}\left\|\sum_{m\in\mathcal{G}(k-1,H^{k-1})}\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\right\|^2.
\end{aligned}
$$

Let's substitute 2 and 3 into 1:

$$
\begin{aligned}
\mathbb{E}_{\xi_m^{k,h}}\mathbb{E}_{\eta_m^{k,h}}\hat{f}(x^{k,h+1}) \leq\ & \hat{f}(x^{k,h})\\
&- \frac{\gamma}{2}\left(1-3\theta-3\delta_1-\frac{8L\gamma^2(\delta_1+1)}{\min_{m\in\mathcal{G}}q_m}\right)\left\|\nabla f(x^{k,h})\right\|^2\\
&+ \frac{\theta\gamma}{\widetilde{G}^2}\left\|\sum_{m\in\mathcal{G}(k-1,H^{k-1})}\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\right\|^2\\
&+ (1-\theta)\left(\gamma\zeta(N)+\frac{4L\gamma^2(\sigma^2+\delta_2)}{\min_{m\in\mathcal{G}}q_m}+\frac{3\gamma\delta_2}{2}\right)+\theta\gamma\delta_2.
\end{aligned}
$$

We put $h = H^k - 1$ and take additional expectations:

$$
\begin{aligned}
\mathbb{E}_{\xi_m^{k-1,0}}\mathbb{E}_{\eta_m^{k-1,0}}\ldots\mathbb{E}_{\xi_m^{k,H^{k-1}}}\mathbb{E}_{\eta_m^{k,H^{k-1}}}\hat{f}(x^{k,H^k}) \leq\ & \hat{f}(x^{k,H^{k-1}})\\
&- \frac{\gamma}{2}\left(1-3\theta-3\delta_1-\frac{8L\gamma^2(\delta_1+1)}{\min_{m\in\mathcal{G}}q_m}\right)\left\|\nabla f(x^{k,H^{k-1}})\right\|^2\\
&+ \frac{\theta\gamma}{\widetilde{G}^2}\left\|\sum_{m\in\mathcal{G}(k-1,H^{k-1})}\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\right\|^2\\
&+ (1-\theta)\left(\gamma\zeta(N)+\frac{4L\gamma^2(\sigma^2+\delta_2)}{\min_{m\in\mathcal{G}}q_m}+\frac{3\gamma\delta_2}{2}\right)+\theta\gamma\delta_2.
\end{aligned}
$$

Taking expectation over H^{k-1} and H^k, using the geometric identity $\mathbb{E}[D_{H^k-1}] = pD_0 + (1-p)\mathbb{E}[D_{H^k}]$, and the independence of H^k and H^{k-1}, we obtain:

$$
\begin{aligned}
&\mathbb{E}_{H^{k-1}}\mathbb{E}_{H^k}\mathbb{E}_{\xi_m^{k-1,0}}\mathbb{E}_{\eta_m^{k-1,0}}\ldots\mathbb{E}_{\xi_m^{k,H^{k-1}}}\mathbb{E}_{\eta_m^{k,H^{k-1}}}\hat{f}\left(x^{k,H^k}\right) \leq p\hat{f}(x^{k,0}) + (1-p)\mathbb{E}_{H^k}\hat{f}\left(x^{k,H^k}\right)\\
&- p\left(\frac{\gamma}{2}-\frac{3\gamma\delta_1}{2}-\frac{3\gamma\theta}{2}-\frac{8L\gamma^2(\delta_1+1)}{\min_{m\in\mathcal{G}}q_m}\right)\left\|\nabla f(x^{k,0})\right\|^2\\
&- (1-p)\left(\frac{\gamma}{2}-\frac{3\gamma\delta_1}{2}-\frac{3\gamma\theta}{2}-\frac{8L\gamma^2(\delta_1+1)}{\min_{m\in\mathcal{G}}q_m}\right)\mathbb{E}_{H^k}\left\|\nabla f(x^{k,H^k})\right\|^2\\
&+ \frac{\theta\gamma}{\widetilde{G}^2}\mathbb{E}_{H^{k-1}}\left\|\sum_{m\in\mathcal{G}(k-1,H^{k-1})}\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\right\|^2\\
&+ (1-\theta)\left(\gamma\zeta(N)+\frac{4L\gamma^2(\sigma^2+\delta_2)}{\min_{m\in\mathcal{G}}q_m}+\frac{3\gamma\delta_2}{2}\right)+\theta\gamma\delta_2. \qquad (4)
\end{aligned}
$$

We now apply Proposition F.4 to the stored availability-corrected surrogate term:

$$\begin{aligned}
&\mathbb{E}_{H^{k-1}}\mathbb{E}_{\xi_m^{k-1,0}}\mathbb{E}_{\eta_m^{k-1,0}}\ldots\mathbb{E}_{\xi_m^{k-1,H^{k-1}-1}}\mathbb{E}_{\eta_m^{k-1,H^{k-1}-1}}\left\|\sum_{m\in\mathcal{G}(k-1,H^{k-1})}\frac{\psi_m^{k-1}}{q_m}g_m^{k-1,H^{k-1}}\right\|^2\\
&\quad\leq\frac{504(1+\delta_1)}{p^2\min_{m\in\mathcal{G}}^2 q_m}\mathbb{E}_{H^{k-1}}\left\|\nabla f(x^{k-1,H^{k-1}})\right\|^2+\frac{1008(\sigma^2+\delta_2)}{p^2\min_{m\in\mathcal{G}}^2 q_m}\\
&\quad=\frac{504(1+\delta_1)}{p^2\min_{m\in\mathcal{G}}^2 q_m}\left\|\nabla f(x^{k,0})\right\|^2+\frac{1008(\sigma^2+\delta_2)}{p^2\min_{m\in\mathcal{G}}^2 q_m}. \qquad (5)
\end{aligned}$$

Substituting (5) into (4) and taking full expectation, we get:

$$\begin{aligned}
\mathbb{E}\hat{f}(x^{k,H^k})&\leq p\hat{f}(x^{k,0})+(1-p)\mathbb{E}\hat{f}(x^{k,H^k})\\
&\quad-p\left(\frac{\gamma}{2}-\frac{3\gamma\delta_1}{2}-\frac{3\gamma\theta}{2}-\frac{8L\gamma^2(\delta_1+1)}{\min_{m\in\mathcal{G}}q_m}-\frac{504\gamma\theta(1+\delta)}{\widetilde{G}^2p^3\min_{m\in\mathcal{G}}^2 q_m}\right)\mathbb{E}\left\|\nabla f(x^{k,0})\right\|^2\\
&\quad-(1-p)\left(\frac{\gamma}{2}-\frac{3\gamma\delta_1}{2}-\frac{3\gamma\theta}{2}-\frac{8L\gamma^2(\delta_1+1)}{\min_{m\in\mathcal{G}}q_m}\right)\mathbb{E}\left\|\nabla f(x^{k,H^k})\right\|^2\\
&\quad+\frac{1008\theta\gamma(\sigma^2+\delta_2)}{\widetilde{G}^2p^2\min_{m\in\mathcal{G}}^2 q_m}+(1-\theta)\left(\gamma\zeta(N)+\frac{4L\gamma^2(\sigma^2+\delta_2)}{\min_{m\in\mathcal{G}}q_m}+\frac{3\gamma\delta_2}{2}\right)+\theta\gamma\delta_2.
\end{aligned}$$

We put $\delta_1\leq\frac{1}{6}$, $\gamma\leq\frac{p\min_{m\in\mathcal{G}}q_m}{40L}$ and $\theta\leq\frac{\gamma Lp^2\min_{m\in\mathcal{G}}q_m}{598}$:

$$\begin{aligned}
\frac{p\gamma}{8}\mathbb{E}\left\|\nabla f(x^{k,0})\right\|^2&\leq p\left(\hat{f}(x^{k,0})-\hat{f}(x^{k,H^k})\right)+\frac{1008\theta\gamma(\sigma^2+\delta_2)}{\widetilde{G}^2p^2\min_{m\in\mathcal{G}}^2 q_m}+\\
&\quad(1-\theta)\left(\gamma\zeta(N)+\frac{4L\gamma^2(\sigma^2+\delta_2)}{\min_{m\in\mathcal{G}}q_m}+\frac{3\gamma\delta_2}{2}\right)+\theta\gamma\delta_2.
\end{aligned}$$

Averaging over all epochs, we obtain the result of the theorem:

$$\begin{aligned}
\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla f(x^{k,0})\right\|^2&\leq\frac{8\left(\hat{f}(x^{0,0})-\hat{f}(x^{K,H^K})\right)}{\gamma K}+\frac{8064\theta(\sigma^2+\delta_2)}{\widetilde{G}^2p^3\min_{m\in\mathcal{G}}^2 q_m}\\
&\quad+\frac{8(1-\theta)\zeta(N)}{p}+\frac{32(1-\theta)L\gamma(\sigma^2+\delta_2)}{p\min_{m\in\mathcal{G}}q_m}+\frac{8\theta\delta_2}{p}\\
&\leq\frac{8\left(\hat{f}(x^{0,0})-\hat{f}(x^*)\right)}{\gamma K}+\frac{32L\gamma(\sigma^2+\delta_2)}{p\min_{m\in\mathcal{G}}q_m}+\frac{14(\sigma^2+\delta_2)\gamma L}{\widetilde{G}^2p\min_{m\in\mathcal{G}}q_m}\\
&\quad+\frac{\delta_2\gamma L}{75p}+\frac{8(1-\theta)\zeta(N)}{p}.
\end{aligned}$$

□

Corollary F.6 (Theorem 3.5)**.** *Under conditions of Theorem F.5 Algorithm 1 needs*

$$\mathcal{O}\left(\frac{M}{C}\left(\frac{\Delta L}{\varepsilon^2}+\frac{\Delta L(\sigma^2+\delta_2)}{\varepsilon^4\min_{m\in\mathcal{G}}q_m}+\frac{\Delta\delta_2 L}{\varepsilon^4\min_{m\in\mathcal{G}}q_m}+\zeta(N)\right)\right)\quad\textit{epochs and}$$

$$\mathcal{O}\left(\frac{M^2}{C}\left(\frac{\Delta L}{\varepsilon^2}+\frac{\Delta L(\sigma^2+\delta_2)}{\varepsilon^4\min_{m\in\mathcal{G}}q_m}\right.\right.$$
$$\left.\left.+\frac{\Delta\delta_2 L}{\varepsilon^4\min_{m\in\mathcal{G}}q_m}+\zeta(N)\right)\right)$$ *number of devices communications*

to reach ϵ-accuracy, where $\epsilon^2=\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla f(x^{k,0})\|^2$, $\Delta=\hat{f}(x^{0,0})-\hat{f}(x^*)$ and C is the number of devices participating in each round.

Proof. Using the result of Theorem F.5, we choose

$$\gamma\leq\min\left\{\frac{p\min_{m\in\mathcal{G}}q_m}{40L},\frac{\sqrt{(\hat{f}(x^{0,0})-\hat{f}(x^*))p\min_{m\in\mathcal{G}}q_m}}{\sqrt{4LK(\sigma^2+\delta_2)}},\right.$$
$$\left.\frac{G\sqrt{4(\hat{f}(x^{0,0})-\hat{f}(x^*))p\min_{m\in\mathcal{G}}q_m}}{\sqrt{7KL(\sigma^2+\delta_2)}},\frac{\sqrt{600(\hat{f}(x^{0,0})-\hat{f}(x^*))p}}{\sqrt{K\delta_2 L}}\right\}.$$

Thus, we need

$$O\left(\frac{(\hat{f}(x^{0,0})-\hat{f}(x^*))L}{p\varepsilon^2}+\frac{(\hat{f}(x^{0,0})-\hat{f}(x^*))L(\sigma^2+\delta_2)}{p\varepsilon^4\min_{m\in\mathcal{G}}q_m}\right.$$
$$\left.+\frac{(\hat{f}(x^{0,0})-\hat{f}(x^*))\delta_2 L}{p\varepsilon^4\min_{m\in\mathcal{G}}q_m}+\frac{\zeta(N)}{p}\right)$$

epochs to reach ε-accuracy, where $\varepsilon^2=\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f(x^{k,0})\right\|^2$. The length of the epoch $H\in\text{Geom}(p)$, Algorithm 1 requires

$$O\left(\frac{(\hat{f}(x^{0,0})-\hat{f}(x^*))L}{p^2\varepsilon^2}+\frac{(\hat{f}(x^{0,0})-\hat{f}(x^*))L(\sigma^2+\delta_2)}{p^2\varepsilon^4\min_{m\in\mathcal{G}}q_m}\right.$$
$$\left.+\frac{(\hat{f}(x^{0,0})-\hat{f}(x^*))\delta_2 L}{p^2\varepsilon^4\min_{m\in\mathcal{G}}q_m}+\frac{\zeta(N)}{p^2}\right)$$

communication rounds.

Next we mention that at each communication round we communicate with C devices, thus, number of communications is

$$O\left(C\frac{(\hat{f}(x^{0,0})-\hat{f}(x^*))L}{p^2\varepsilon^2}+C\frac{(\hat{f}(x^{0,0})-\hat{f}(x^*))L(\sigma^2+\delta_2)}{p^2\varepsilon^4\min_{m\in\mathcal{G}}q_m}\right.$$
$$\left.+C\frac{(\hat{f}(x^{0,0})-\hat{f}(x^*))\delta_2 L}{p^2\varepsilon^4\min_{m\in\mathcal{G}}q_m}+C\frac{\zeta(N)}{p^2}\right).$$

Taking $p=\frac{C}{M}$, we have the result of the corollary. The choice of p is motivated by the fact that we perform $\frac{1}{p}C+M$ communications per-epoch, and established p is the minimal, which delivers $\mathcal{O}(M)$ communications at each epoch. This is also the reason for the additional factor M in the estimate on communications. □