Mitigating Data Heterogeneity Effect in Client-Reshuffling-Based Federated Learning

Su Zhang[1], **Peiran Yu**[2], **Heng Huang**[1]
[1]Department of Computer Science, University of Maryland, College Park,
[2]Department of Computer Science and Engineering, University of Texas at Arlington
{suzhang1,heng}@umd.edu, peiran.yu@uta.edu

Abstract

Data heterogeneity and low client participation are two key challenges in federated learning (FL). To improve participation efficiency under low client participation, recent work has introduced client reshuffling, which samples clients without replacement. However, client reshuffling does not address data heterogeneity. Moreover, its without-replacement sampling introduces sampling bias, complicating convergence analysis under heterogeneous data. Existing analyses of client reshuffling either focus on convex objectives or retain heterogeneity error terms in the nonconvex setting. To address this gap, we propose **FedCDR**, a federated learning method that combines client reshuffling with Douglas–Rachford splitting. FedCDR supports inexact local proximal updates, enabling a flexible trade-off between local computation and communication. For smooth nonconvex objectives, we establish an $O(\epsilon^{-1})$ meta-epoch complexity for finding an ϵ-approximate stationary point. Since each meta-epoch consists of n/C communication rounds, this yields an $O(n/(C\epsilon))$ communication-round complexity, where n is the total number of clients and C is the number of participating clients per round. Importantly, our convergence bound does not explicitly depend on client data heterogeneity, avoiding the heterogeneity error terms in existing client-reshuffling analyses. Experiments on synthetic and benchmark datasets show that FedCDR achieves the highest mean test accuracy in nine evaluated settings and requires the fewest communication rounds to reach target accuracy in five of six participation settings.

1 Introduction

Federated learning (FL) enables a large number of clients to collaboratively train a shared model while keeping their raw data local, and has been deployed in real-world applications such as on-device prediction systems [Hard et al., 2018]. However, FL often faces several key challenges in practical applications, such as the data heterogeneity challenge [Li et al., 2020a, Karimireddy et al., 2020] and the low client participation issue [Malinovsky et al., 2023]. Data heterogeneity arises because clients' datasets are typically non-IID, which can lead to client drift as clients optimize different local objectives. Low client participation arises because only a small fraction of clients can participate in each communication round due to high communication costs and system constraints, and naive client sampling strategies lead to inefficient and unfair client participation.

To mitigate the low client participation issue, recent works propose *client reshuffling* [Malinovsky et al., 2023], i.e., random sampling without replacement. Client reshuffling groups multiple communication rounds into a *meta-epoch*: at the beginning of each meta-epoch, the server generates a random permutation of all clients and partitions it into batches, ensuring that each client participates exactly once per meta-epoch. This improves participation efficiency, leads to faster convergence, and matches practical constraints where devices may limit how many times they can participate.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, mitigating data heterogeneity under client-reshuffling-based[1] FL remains challenging. Since client reshuffling samples clients without replacement, the clients selected in each round depend on those selected earlier in the meta-epoch. When client data are heterogeneous, the expected update conditioned on previous selection differs from the expected update of the whole meta-epoch selection, complicating convergence analysis. Recent studies of client reshuffling [Malinovsky et al., 2023, Demidovich et al., 2024] provide convergence guarantees, but their bounds contain heterogeneity error terms or step-size restrictions. This notable gap raises a natural question:

> *Is it possible to design a federated learning algorithm that addresses the data heterogeneity challenge under the client-reshuffling-based FL, without relying on additional restrictive assumptions?*

To address this question, we propose **FedCDR**, a federated learning method that combines client reshuffling with Douglas–Rachford (DR) splitting. Our main contributions are summarized as follows.

- **New Method.** We develop FedCDR (**Fed**erated **C**lient Reshuffling **D**ouglas–**R**achford Method; Algorithm 1) for smooth nonconvex federated learning. FedCDR integrates client reshuffling with DR splitting and maintains client-specific states across communication rounds. Moreover, FedCDR supports *inexact* local proximal updates, avoiding costly exact proximal solves and enabling a flexible trade-off between local computation and communication.
- **Convergence Analysis.** We establish an $O(\epsilon^{-1})$ meta-epoch complexity for finding an ϵ-approximate stationary point (i.e., $\mathbb{E}\|\nabla f(\tilde{x})\|^2 \leq \epsilon$) under smooth nonconvex objectives. Since each meta-epoch consists of n/C communication rounds, this yields an $O(n/(C\epsilon))$ communication-round complexity. Importantly, the leading constant in our convergence bound does not depend on client data heterogeneity. Consequently, the bound avoids the heterogeneity-dependent error terms present in existing client-reshuffling analyses. The key technical idea is to analyze FedCDR over an entire meta-epoch rather than individual communication rounds. By exploiting the fact that each client participates exactly once per meta-epoch, we can bound the cumulative effect of client reshuffling on convergence.
- **Experimental Validation.** We evaluate FedCDR on synthetic and benchmark datasets under heterogeneous data distributions. FedCDR achieves higher final test accuracy than competitive baselines and requires fewer communication rounds to reach target accuracy in most evaluated settings. We further investigate the communication–computation trade-off, the effect of client participation, and the scalability of FedCDR.

2 Related Work

Federated Optimization under Data Heterogeneity. Federated Averaging (FedAvg) [McMahan et al., 2017] performs local stochastic gradient updates followed by server aggregation. Under heterogeneous client data, local updates may drift toward different client objectives, degrading convergence [Li et al., 2020b, Karimireddy et al., 2020]. Several methods have been developed to mitigate this issue. FedProx [Li et al., 2020a] introduces a proximal term to stabilize local optimization, SCAFFOLD [Karimireddy et al., 2020] uses control variates to correct client drift, and FedDC [Gao et al., 2022] employs drift-correction variables. FedDR [Tran-Dinh et al., 2021] applies randomized Douglas–Rachford splitting to nonconvex federated composite optimization and supports inexact proximal updates. Our focus is to establish convergence without an explicit heterogeneity-dependent error term under client reshuffling. A complementary line of research, including IFCA [Ghosh et al., 2020] and CLoVE [Bhatia et al., 2025], addresses data heterogeneity by learning cluster-specific models, whereas we consider a single global objective.

Client Participation and Reshuffling. Under conventional partial participation, client cohorts are typically sampled independently across communication rounds, which does not guarantee that every client is selected within a fixed number of rounds. Alternative participation mechanisms include cyclic client scheduling [Cho et al., 2023] and general client participation patterns [Wang and Ji,

[1] By *client-reshuffling-based FL*, we mean federated learning algorithms that adopt client reshuffling as the client sampling strategies.

2022]. FedShuffle [Horváth et al., 2022] incorporates random reshuffling of local training examples together with client sampling, which differs from reshuffling the client order. In contrast, regularized client participation [Malinovsky et al., 2023] partitions the full client set into disjoint cohorts within each meta-epoch, ensuring that every client participates exactly once. This improves client coverage but introduces dependence among the client cohorts selected in successive rounds. Further discussion of client coverage is provided in Appendix D.

Convergence Analysis under Client Reshuffling. RR–CLI [Malinovsky et al., 2023] combines client reshuffling with local data reshuffling and establishes convergence guarantees for convex and strongly convex objectives. Its bounds involve heterogeneity-related quantities evaluated at an optimum, such as $\tilde{\sigma}_*^2 = n^{-1}\sum_{i=1}^n \|\nabla f_i(x^*)\|^2$. Clipped RR–CLI [Demidovich et al., 2024] extends client reshuffling to nonconvex objectives under generalized (L_0, L_1)-smoothness, with heterogeneity-related quantities appearing in its convergence bounds and sufficient stepsize conditions. These analyses consider different objective assumptions and convergence criteria. For L-smooth nonconvex objectives, FedCDR combines client reshuffling with Douglas–Rachford splitting and establishes an $O(1/T)$ bound on $\mathbb{E}\|\nabla f(\tilde{x})\|^2$, where T denotes the number of meta-epochs. The bound contains no explicit data-heterogeneity error term and applies to both exact proximal updates and inexact updates satisfying the stated error conditions. A detailed comparison of assumptions, convergence criteria, and communication complexity is provided in Appendix I.

3 Preliminaries

We denote $\mathbb{R}^n$ as the n-dimensional Euclidean space with inner product $\langle\cdot,\cdot\rangle$ and Euclidean norm $\|\cdot\|$. An extended-real-valued function $f : \mathbb{R}^n \to [-\infty, \infty]$ is said to be proper if $\operatorname{dom} f := \{x \in \mathbb{R}^n : f(x) < \infty\}$ is not empty and f never equals $-\infty$. We say a proper function f is closed if it is lower semicontinuous.

In this paper, we consider the following nonconvex optimization problem:

$$\min_{x\in\mathbb{R}^d}\left[f(x) := \frac{1}{n}\sum_{i=1}^n f_i(x)\right], \tag{1}$$

where f_i is differentiable. In federated learning, n is the total number of clients. $x \in \mathbb{R}^d$ corresponds to the parameters of the model. $f_i : \mathbb{R}^d \to \mathbb{R}$ is empirical loss of the model x defined by the local training data stored by the i_{th} client.

To analyze the optimization problem (1), we make the following assumption.

Assumption 1. (Well-defined Optimization Problem)
$\operatorname{dom}(f) \neq \emptyset$ and $f^* := \inf_{x\in\mathbb{R}^d} f(x) > -\infty$

All of our theoretical results rely on the smoothness property of f_i.

Definition 1. (L-smoothness). Function $h : \mathbb{R}^d \to \mathbb{R}$ is called L-smooth if it is continuously differentiable and its gradient is L-Lipschitz continuous for some $L > 0$

$$\|\nabla h(x) - \nabla h(y)\| \leq L\|x - y\| \ \ \forall x, y \in \mathbb{R}^d. \tag{2}$$

For (1), we made the following assumption.

Assumption 2. The $f_1, f_2, .., f_n$ in (1) are L-smooth functions.

Assumptions 1 and 2 are widely used in the study of non-convex optimization problems.

Then, we define the ϵ-stationary point for (1).

Definition 2. (ϵ-approximate stationary point) Consider (1). If $\tilde{x} \in \operatorname{dom}(f)$ satisfies $\mathbb{E}[\|\nabla f(\tilde{x})\|^2] \leq \epsilon$, then $\tilde{x}$ is called ϵ-approximate stationary point, where we use expectation to eliminate the randomness of the algorithm.

Our method uses the proximal operator in DR iterations. Therefore, we provide its definition below.

Definition 3. (Proximal Operator). For a proper function $f : \mathbb{R}^n \to [-\infty, \infty]$, we denote the proximal operator of f as

$$\mathrm{prox}_{\beta f}(x) := \arg\min_{z \in \mathbb{R}^n} \left\{ f(z) + \frac{1}{2\beta}\|z - x\|^2 \right\}.$$

Even if f is nonconvex, under Assumption 2, choosing $0 < \beta < \frac{1}{L}$ ensures that the proximal operator $\mathrm{prox}_{\beta f}$ is well-defined and single-valued. Moreover, evaluating $\mathrm{prox}_{\beta f}$ amounts to solving a strongly convex optimization problem.

However, in some cases, $\mathrm{prox}_{\beta f}$ can only be computed approximately up to an accuracy $\epsilon > 0$. That is, if $x^* := \mathrm{prox}_{\beta f}(x)$ denotes the exact solution, then an approximate solution $\tilde{x} \approx \mathrm{prox}_{\beta f}(x)$ satisfies $\|\tilde{x} - x^*\| \leq \epsilon$. Approximating $\mathrm{prox}_{\beta f}$ can be efficiently performed using various existing methods, including stochastic gradient descent (SGD) and accelerated gradient-based algorithms.

4 Algorithm

We now introduce **FedCDR**, which combines Douglas–Rachford (DR) splitting with client reshuffling to mitigate the effects of data heterogeneity. Under heterogeneous data, local updates may drift toward different client objectives. DR helps address this issue by adjusting each client's local update based on the difference between its local model and the current global model [Tran-Dinh et al., 2021]. Existing DR-based FL methods, such as FedDR, use with-replacement client sampling, under which clients may be selected repeatedly across communication rounds. In contrast, client reshuffling samples without replacement, ensuring that each client participates once per meta-epoch and improving client participation efficiency. Algorithm 1 summarizes the method, and Appendix A provides its derivation.

FedCDR runs for T meta-epochs, indexed by $t = 0, \ldots, T-1$, each consisting of $R := n/C$ communication rounds indexed by $r = 0, \ldots, R-1$, where C is the number of participating clients per round (assuming C divides n). At the beginning of each meta-epoch t, the server generates a random permutation $\pi^t = (\pi_1^t, \ldots, \pi_n^t)$ of all clients and partitions it into R disjoint batches:

$$S_r^t := \{\pi_{rC+1}^t, \ldots, \pi_{(r+1)C}^t\}, \quad r = 0, \ldots, R-1.$$

Only clients in S_r^t participate in round r. The server maintains the global model x_r^t, while each client i maintains its local DR states $(y_{i,r}^t, x_{i,r}^t, \hat{x}_{i,r}^t)$. Inactive clients keep their states unchanged, and all states are carried over to the next meta-epoch.

Let $u := x^0$. We initialize the server model and each client's DR states as

$$\begin{aligned} x_{i,0}^0 &= u, \\ y_{i,0}^0 &= u + \eta \nabla f_i(u), \\ \hat{x}_{i,0}^0 &= u - \eta \nabla f_i(u), \\ x_0^0 &= \frac{1}{n}\sum_{i=1}^n \hat{x}_{i,0}^0 = u - \eta \nabla f(u). \end{aligned} \tag{3}$$

At communication round r of meta-epoch t, the server broadcasts the current global model x_r^t to the participating clients $i \in S_r^t$. Each participating client performs the following DR updates:

$$\begin{aligned} y_{i,r+1}^t &= y_{i,r}^t + \alpha(x_r^t - x_{i,r}^t), \\ x_{i,r+1}^t &\approx \mathrm{prox}_{\eta f_i}(y_{i,r+1}^t), \\ \hat{x}_{i,r+1}^t &= 2x_{i,r+1}^t - y_{i,r+1}^t, \end{aligned} \tag{4}$$

where $\eta > 0$ is the proximal coefficient and $\alpha \in (0, 2]$ is the relaxation coefficient. Each participating client then sends the increment of its reflected state, $g_{i,r}^t := \hat{x}_{i,r+1}^t - \hat{x}_{i,r}^t$, to the server. The server aggregates these increments to update the global model:

$$x_{r+1}^t = x_r^t + \frac{1}{n}\sum_{i \in S_r^t} g_{i,r}^t. \tag{5}$$

Algorithm 1 FedCDR: Client Reshuffling with Douglas–Rachford Splitting

Input: Initial point $x^0 \in \mathrm{dom}(f)$, proximal coefficient $\eta > 0$, relaxation coefficient $\alpha \in (0, 2]$, number of meta-epochs T, and clients per round C.
Set $u \leftarrow x^0$ and $R \leftarrow n/C$.
Initialize: At each client $i \in [n]$, set
$x_{i,0}^0 \leftarrow u$, $y_{i,0}^0 \leftarrow u + \eta \nabla f_i(u)$.
$\hat{x}_{i,0}^0 \leftarrow u - \eta \nabla f_i(u)$.
Server sets $x_0^0 \leftarrow \frac{1}{n} \sum_{i=1}^n \hat{x}_{i,0}^0$.
for $t = 0, \ldots, T-1$ **do**
  Client reshuffling: Generate a random permutation of $[n]$ and partition it into batches $S_0^t, \ldots, S_{R-1}^t$ of size C.
  for $r = 0, \ldots, R-1$ **do**
    Server broadcasts x_r^t to clients in S_r^t.
    for each client $i \in S_r^t$ **in parallel do**
      $y_{i,r+1}^t \leftarrow y_{i,r}^t + \alpha(x_r^t - x_{i,r}^t)$
      $x_{i,r+1}^t \approx \mathrm{prox}_{\eta f_i}(y_{i,r+1}^t)$
      $\hat{x}_{i,r+1}^t \leftarrow 2x_{i,r+1}^t - y_{i,r+1}^t$
      $g_{i,r}^t \leftarrow \hat{x}_{i,r+1}^t - \hat{x}_{i,r}^t$
      Send $g_{i,r}^t$ to the server.
    end for
    Inactive clients retain their previous DR states.
    Server update: $x_{r+1}^t \leftarrow x_r^t + \frac{1}{n} \sum_{i \in S_r^t} g_{i,r}^t$.
  end for
  $x_0^{t+1} \leftarrow x_R^t$.
  Carry over each client's DR states to the next meta-epoch.
end for
Output: Final global model x_R^{T-1} (used in practice).
For convergence analysis: $\tilde{x}$ selected uniformly from $\{x_r^t : 0 \le t < T,\ 0 \le r < R\}$.

The aggregation uses $1/n$ rather than $1/C$, since inactive clients retain their states and contribute zero increments.

To avoid the computational cost of solving local proximal subproblems exactly, FedCDR supports inexact proximal updates. For the convergence analysis, we assume that the stored state of each client satisfies

$$\mathbb{E}\left[\left\|x_{i,r}^t - \mathrm{prox}_{\eta f_i}(y_{i,r}^t)\right\|^2\right] \le \epsilon_{t,i,r}^2, \tag{6}$$

where $\epsilon_{t,i,r} \ge 0$ is a deterministic upper bound on the mean-square proximal approximation error. This condition also applies to inactive clients, whose states remain unchanged until their next participation. The exact variant is recovered when all approximation errors are zero. In practice, clients approximate the proximal solution using a finite number of local SGD steps. Increasing the number of local steps can improve the approximation accuracy at the cost of additional local computation, allowing a trade-off between communication and computation. We establish convergence guarantees for both the exact and inexact variants in Theorems 3 and 1, respectively. This trade-off is further examined in Section 6.4 and Appendix F.

5 Convergence Analysis

In this section, we establish convergence guarantees for the inexact and exact variants of *Algorithm 1* under Assumptions 1 and 2. For the convergence analysis, let $\tilde{x}$ be selected uniformly at random from $\{x_r^t : 0 \le t < T,\ 0 \le r < R\}$, where $R = n/C$.

Inexact proximal updates. We first consider the inexact variant, where the local proximal subproblems are solved approximately. The following theorem gives a convergence bound in terms of the mean-square approximation errors of the stored client states.

Theorem 1. *Under Assumptions 1 and 2, suppose that the sequence generated by Algorithm 1 satisfies the mean-square error bounds (6) for all stored client states, including inactive and carried-over states. Let $0 < \alpha < 2$, choose $0 < \gamma < (2-\alpha)/(2\alpha)$, and let $0 < \eta < \eta_0$, where η_0 is defined in (33). Then,*

$$\mathbb{E}\big[\|\nabla f(\tilde{x})\|^2\big] \le \frac{\mathcal{A}_1}{T}\big(f(x^0) - f^*\big) + \frac{\mathcal{A}_1}{nT}\sum_{t=0}^{T-1}\sum_{r=0}^{R-1}\sum_{i=1}^{n}\Big(\mathcal{B}_1\epsilon_{t,i,r}^2 + \mathcal{B}_2\epsilon_{t,i,r+1}^2\Big), \tag{7}$$

where $\mathcal{A}_1$, $\mathcal{B}_1$, and $\mathcal{B}_2$ are defined in (66).

If the deterministic accuracies $\epsilon_{t,i,r}$ satisfy

$$\frac{1}{n}\sum_{t=0}^{T-1}\sum_{r=0}^{R}\sum_{i=1}^{n}\epsilon_{t,i,r}^2 \le M \tag{8}$$

for some constant $M \ge 0$, for fixed $R = n/C$ and every $T \ge 1$, then running Algorithm 1 for at least $K := \max\Big\{1, \Big\lceil \frac{\mathcal{A}_1[f(x^0)-f^*+(\mathcal{B}_1+\mathcal{B}_2)M]}{\epsilon}\Big\rceil\Big\}$ meta-epochs yields an ϵ-approximate stationary point of (1) in the sense of Definition 2.

The following corollary gives a specific choice of coefficients in *Algorithm 1*.

Corollary 2. *Under the assumptions and error budget of Theorem 1, choose $\alpha = 1$, $\eta = \frac{1}{4L}$, and $\gamma = \frac{1}{25}$. Then,*

$$\mathbb{E}[\|\nabla f(\tilde{x})\|^2] \le \frac{625L}{2T}\big[(f(x^0) - f^*) + 325LM\big]. \tag{9}$$

In particular, if Algorithm 1 runs for at least $K := \max\Big\{1, \Big\lceil \frac{625L[(f(x^0)-f^*)+325LM]}{2\epsilon}\Big\rceil\Big\}$ meta-epochs, then $\tilde{x}$ is an ϵ-approximate stationary point of (1).

Exact proximal updates. We next consider the exact variant, where all local proximal subproblems are solved exactly. In this case, a separate analysis gives a tighter convergence bound.

Theorem 3. *Under Assumptions 1 and 2, suppose that Algorithm 1 uses exact proximal updates, $0 < \alpha < 2$, and $0 < \eta < \eta_1$, where η_1 is defined in (82). Then,*

$$\mathbb{E}[\|\nabla f(\tilde{x})\|^2] \le \frac{\mathcal{A}[f(x^0) - f^*]}{T}, \tag{10}$$

where $\mathcal{A}$ is defined in (105).

Moreover, if Algorithm 1 runs for at least $K := \max\Big\{1, \Big\lceil \frac{\mathcal{A}[f(x^0)-f^*]}{\epsilon}\Big\rceil\Big\}$ meta-epochs, then $\tilde{x}$ is an ϵ-approximate stationary point of (1).

The following corollary specifies the same proximal and relaxation coefficients as in Corollary 2.

Corollary 4. *Under the assumptions of Theorem 3, let $\alpha = 1$ and $\eta = \frac{1}{4L}$. Then,*

$$\mathbb{E}[\|\nabla f(\tilde{x})\|^2] \le \frac{125L[f(x^0) - f^*]}{T}. \tag{11}$$

In particular, if Algorithm 1 runs for at least $K := \max\Big\{1, \Big\lceil \frac{125L[f(x^0)-f^*]}{\epsilon}\Big\rceil\Big\}$ meta-epochs, then $\tilde{x}$ is an ϵ-approximate stationary point of (1).

Discussion. The proofs of Theorems 1 and 3, together with their corollaries, are provided in Appendices B and C, respectively.

For the exact variant, Corollary 4 establishes an $O(1/T)$ convergence rate, yielding an $O(\epsilon^{-1})$ meta-epoch complexity for finding an ϵ-approximate stationary point. For the inexact variant, Corollary 2 establishes the same convergence rate, provided that the accumulated approximation errors satisfy (8) with a constant M independent of T. Under this condition, the effect of the approximation errors

also decreases at a rate of $O(1/T)$. Since each meta-epoch consists of $R = n/C$ communication rounds, both variants achieve an $O(n/(C\epsilon))$ communication complexity for fixed n and C.

Importantly, neither convergence bound contains an explicit data-heterogeneity error term. In contrast to the heterogeneity-dependent bound in Malinovsky et al. [2023], our results do not involve a quantity such as $\tilde{\sigma}_*^2 := \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x^*)\|^2$. This is the main theoretical advantage of FedCDR.

Although both variants achieve an $O(1/T)$ rate, their convergence bounds differ. As shown in Corollaries 2 and 4, the inexact variant incurs an additional approximation-error term, whereas the exact variant provides a tighter theoretical bound for the specified coefficients. Improving the accuracy of the local proximal updates can reduce the approximation-error effect, but may require more local computation. Thus, the proximal approximation accuracy affects the convergence bound and local computational cost, while the asymptotic rate remains unchanged when (8) holds.

Choice of inexact accuracies. The error bounds in Theorem 1 apply to all stored client states, including those of inactive clients. One sufficient choice is to require that each newly computed proximal state in meta-epoch t has expected squared approximation error at most $M/[4R(t+1)^2]$, starting from the exact initialization (3). Since inactive clients retain their states, their approximation errors are carried over until the next update.

Let $\beta_t^2 := M/[4R(t+1)^2]$ for $t \geq 0$ and $\beta_{-1}^2 := 0$. Then one choice of deterministic error bounds is

$$\epsilon_{t,i,r}^2 = \begin{cases} \beta_{t-1}^2, & r = 0, \\ \beta_{t-1}^2 + \beta_t^2, & 1 \leq r < R, \\ \beta_t^2, & r = R. \end{cases} \tag{12}$$

Appendix B verifies that these bounds satisfy (8).

SGD for inexact proximal updates. In practice, the proximal operator $\mathrm{prox}_{\eta f_i}(v)$ can be approximated by running $N_{t,r}$ iterations (or $E_{t,r}$ epochs) of mini-batch SGD on the subproblem

$$\min_x \ \phi_i(x; v) := f_i(x) + \frac{1}{2\eta}\|x - v\|^2. \tag{13}$$

By the L-smoothness of f_i, we have

$$\begin{aligned} \phi_i(x; v) \geq \phi_i(z; v) + \langle \nabla \phi_i(z; v), x - z \rangle \\ + \frac{1/\eta - L}{2}\|x - z\|^2. \end{aligned} \tag{14}$$

Thus, for $\eta < 1/L$, the proximal subproblem is $(1/\eta - L)$-strongly convex and has a unique minimizer $x^\star = \mathrm{prox}_{\eta f_i}(v)$.

Under suitable step-size and gradient-noise conditions, standard SGD convergence results can bound $\mathbb{E}\|x_{i,r}^t - x^\star\|^2$ in terms of the number of local iterations $N_{t,r}$. Hence, increasing the local computation budget can improve the proximal approximation accuracy.

6 Experiments

We evaluate FedCDR on synthetic and benchmark datasets across different levels of data heterogeneity and client participation rates. FedCDR achieves the highest mean test accuracy across all nine evaluated settings and requires the fewest synchronization rounds in five of six participation settings. We further investigate the communication–computation trade-off, scalability to larger client populations and models, and robustness to client dropout.

6.1 Experimental Setup

Datasets and Data Heterogeneity. We consider three synthetic datasets, Synthetic-$(0, 0)$, Synthetic-$(1, 1)$, and Synthetic-$(5, 5)$, generated following [Li et al., 2020a]. In Synthetic-(β_1, β_2), the parameters β_1 and β_2 control variations in local model parameters and feature distributions, respectively, with larger values indicating greater heterogeneity. We also evaluate MNIST, CIFAR-10, and CIFAR-100 [Krizhevsky, 2009], using Dirichlet-based data partitions [Wang et al., 2020] with concentration

parameters $\rho \in \{0.1, 1.0\}$. Smaller ρ corresponds to greater data heterogeneity. Further details are provided in Appendix G.1.

Models. We use fully connected neural networks for the synthetic datasets and MNIST, and ResNet-20 [He et al., 2016] with group normalization [Wu and He, 2018] for CIFAR-10 and CIFAR-100. Model architectures are detailed in Appendix G.2.

Baselines. We compare FedCDR with FedAvg [McMahan et al., 2017], FedProx [Li et al., 2020a], SCAFFOLD [Karimireddy et al., 2020], FedDC [Gao et al., 2022], and FedDR [Tran-Dinh et al., 2021]. We also consider client-reshuffling variants of FedAvg, FedProx, SCAFFOLD, and FedDC, denoted by the prefix CR+.

Training and Evaluation. Unless otherwise specified, we consider 500 clients with a participation rate of 10% per communication round. For the main comparisons, all methods use $E = 5$ local epochs per participating client, with matched batch sizes, optimizer schedules, and hyperparameter tuning budgets. We use mini-batch SGD for local optimization, including the approximate proximal updates in FedCDR. Hyperparameters are selected by grid search based on validation performance. We report final top-1 test accuracy after 400 synchronizations as mean $\pm$ standard deviation over five independent runs. Additional implementation details are provided in Appendix G.2.

6.2 Main Results

Table 1: Top-1 test accuracy (%) on synthetic datasets after 400 communication rounds. Results are reported as mean $\pm$ standard deviation over five runs. The best mean accuracy in each column is highlighted in bold.

Method	**Synthetic-(0,0)**	**Synthetic-(1,1)**	**Synthetic-(5,5)**
FedAvg	86.23±0.32	74.73±0.35	41.89±1.15
FedProx	86.03±0.73	75.19±0.79	60.01±1.02
SCAFFOLD	89.71±0.21	88.14±0.20	78.27±0.29
FedDC	91.38±0.17	90.20±0.19	79.88±0.24
FedDR	90.84±0.46	90.11±0.21	78.46±0.98
CR+FedAvg	88.12±0.21	77.54±0.42	46.15±1.32
CR+FedProx	87.78±0.18	80.30±0.31	62.42±0.57
CR+SCAFFOLD	91.34±0.25	88.15±0.19	77.92±0.32
CR+FedDC	92.25±0.14	90.04±0.21	79.78±0.26
FedCDR	**92.91**±0.18	**92.23**±0.21	**83.95**±0.68

Table 2: Top-1 test accuracy (%) on benchmark datasets after 400 communication rounds. Results are reported as mean $\pm$ standard deviation over five runs. The best mean accuracy in each column is highlighted in bold.

Method	$\rho = 0.1$			$\rho = 1.0$		
	MNIST	CIFAR-10	CIFAR-100	MNIST	CIFAR-10	CIFAR-100
FedAvg	97.02±0.12	67.89±0.46	53.45±0.56	97.56±0.05	78.19±0.47	64.88±0.42
FedProx	97.19±0.08	65.11±0.55	53.67±0.83	97.82±0.08	79.11±0.16	65.13±0.35
SCAFFOLD	97.90±0.06	73.12±0.42	61.69±0.35	98.10±0.07	82.39±0.35	68.72±0.25
FedDC	98.04±0.03	75.38±0.30	64.80±0.35	98.10±0.03	83.40±0.27	69.04±0.36
FedDR	97.99±0.08	73.45±0.34	63.22±0.48	98.07±0.05	81.97±0.38	69.05±0.34
CR+FedAvg	97.45±0.14	69.79±0.70	55.68±0.70	97.79±0.07	78.29±0.50	65.89±0.41
CR+FedProx	97.55±0.05	68.01±0.39	55.78±0.85	97.64±0.08	78.53±0.22	65.41±0.54
CR+SCAFFOLD	98.06±0.05	74.45±0.40	63.47±0.34	98.02±0.06	82.35±0.30	69.67±0.25
CR+FedDC	98.00±0.03	75.24±0.25	64.56±0.35	98.29±0.03	83.56±0.24	69.28±0.34
FedCDR	**98.20**±0.01	**76.87**±0.42	**66.43**±0.61	**98.49**±0.09	**84.46**±0.31	**71.63**±0.32

Synthetic Datasets. Table 1 presents the results under different levels of data heterogeneity. FedCDR achieves the highest mean test accuracy across all three settings. As data heterogeneity increases, test accuracy decreases across all methods, while FedCDR's advantage over the best-performing baseline grows, reaching 4.07 percentage points on Synthetic-$(5, 5)$.

Benchmark Datasets. Table 2 reports the results on MNIST, CIFAR-10, and CIFAR-100. FedCDR achieves the highest mean test accuracy across all six settings. FedCDR shows smaller gains on MNIST but achieves improvements of over 1.4 percentage points over the best-performing baselines on both CIFAR-10 and CIFAR-100 at $\rho = 0.1$.

6.3 Communication Efficiency

We evaluate communication efficiency by measuring the number of communication rounds required to reach 90% test accuracy under different client participation rates.

As shown in Table 3, FedCDR requires the fewest communication rounds in five of the six settings. At 5% participation, FedCDR requires approximately half as many rounds as FedDR on both synthetic datasets. The advantage over FedDR is smaller at higher participation rates. At 50% participation, CR+FedDC requires slightly fewer rounds than FedCDR on Synthetic-$(0, 0)$ (76 versus 77). These results indicate that FedCDR is particularly effective at reducing communication rounds under low client participation.

Table 3: Communication rounds required to reach 90% top-1 test accuracy on synthetic datasets under different client participation rates. > 400 indicates that the target was not reached within 400 rounds. The fewest rounds in each setting are highlighted in bold.

Method	5% participation		10% participation		50% participation	
	$(0, 0)$	$(1, 1)$	$(0, 0)$	$(1, 1)$	$(0, 0)$	$(1, 1)$
FedAvg	> 400	> 400	> 400	> 400	> 400	> 400
FedProx	> 400	> 400	> 400	> 400	> 400	> 400
FedDR	303	389	148	168	92	123
CR+SCAFFOLD	320	> 400	203	> 400	87	312
CR+FedDC	204	378	134	279	**76**	145
FedCDR	**158**	**193**	**123**	**142**	77	**103**

6.4 Communication–Computation Trade-off

We examine the communication–computation trade-off in FedCDR by considering different numbers of local epochs E for the inexact proximal updates. Table 4 reports the communication rounds required to reach 98% test accuracy on MNIST. Increasing E reduces the required rounds under both heterogeneity settings. Specifically, increasing E from 5 to 20 reduces the rounds from 112 to 43 at $\rho = 0.1$ and from 89 to 32 at $\rho = 1.0$, with four times as many local epochs per participating client. These results illustrate how additional local computation can reduce the number of communication rounds in FedCDR. Further details are provided in Appendix F.

Table 4: Communication–computation trade-off of FedCDR on MNIST. Rounds denotes the communication rounds required to reach 98% test accuracy. Comm. and Local comp. denote the speedup factors in communication rounds and local computation per round, respectively, relative to $E = 10$.

Local epochs E	$\rho = 0.1$			$\rho = 1.0$		
	Rounds	Comm.	Local comp.	Rounds	Comm.	Local comp.
3	161	$0.37\times$	$3.33\times$	125	$0.39\times$	$3.33\times$
5	112	$0.54\times$	$2.00\times$	89	$0.55\times$	$2.00\times$
10	60	$1.00\times$	$1.00\times$	49	$1.00\times$	$1.00\times$
20	43	$1.40\times$	$0.50\times$	32	$1.53\times$	$0.50\times$

6.5 Scalability and Robustness

Scalability. We evaluate FedCDR with larger client populations and models. With 10,000 clients, FedCDR achieves the highest mean test accuracy across all six benchmark settings, as reported in Appendix H. We further consider ResNet-18 on CIFAR-10 and LoRA fine-tuning of Qwen2.5-0.5B on AG News. FedCDR achieves the highest mean accuracy on both tasks, reaching 86.23% and 92.09%, respectively. Detailed results are provided in Appendix H.2.

Robustness to Client Dropout. We examine robustness to client dropout by allowing each selected client to fail to respond independently with probability $q \in \{0, 0.1, 0.3, 0.5\}$. Nonresponding clients are skipped without retries. On CIFAR-10, FedCDR achieves the highest test accuracy across all dropout rates, maintaining 83.74% accuracy at $q = 0.5$, compared with 84.46% without dropout. Detailed results are provided in Appendix H.3.

7 Conclusion

In this work, we proposed FedCDR, a novel federated learning method that combines client reshuffling with Douglas–Rachford splitting to mitigate the effects of data heterogeneity. For smooth nonconvex objectives, we establish an $O(1/T)$ convergence rate for both exact and inexact variants. Both guarantees yield an $O(\epsilon^{-1})$ meta-epoch complexity and an $O(n/(C\epsilon))$ communication-round complexity, without explicit data-heterogeneity error terms in the convergence bounds. Experiments on synthetic and benchmark datasets show that FedCDR achieves the highest mean test accuracy in all nine evaluated settings and requires the fewest communication rounds in five of six participation settings. Additional experiments demonstrate the communication–computation trade-off and evaluate the scalability and robustness of FedCDR.

Acknowledgements

This work was partially supported by NSF IIS 2347592, 2348169, DBI 2405416, CCF 2348306, CNS 2347617, RISE 2536663.

Competing Interests

The authors declare no competing interests.

References

Randeep Bhatia, Nikos Papadis, Murali S. Kodialam, T. V. Lakshman, and Sayak Chakrabarty. Clove: Personalized federated learning through clustering of loss vector embeddings. *CoRR*, abs/2506.22427, 2025. doi: 10.48550/ARXIV.2506.22427. URL `https://doi.org/10.48550/arXiv.2506.22427`.

Yae Jee Cho, Pranay Sharma, Gauri Joshi, Zheng Xu, Satyen Kale, and Tong Zhang. On the convergence of federated averaging with cyclic client participation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 5677–5721. PMLR, 2023. URL `https://proceedings.mlr.press/v202/cho23b.html`.

Yury Demidovich, Petr Ostroukhov, Grigory Malinovsky, Samuel Horváth, Martin Takáč, Peter Richtárik, and Eduard Gorbunov. Methods with local steps and random reshuffling for generally smooth non-convex federated optimization, 2024. URL `https://arxiv.org/abs/2412.02781`.

Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10102–10111. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00987. URL `https://doi.org/10.1109/CVPR52688.2022.00987`.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/e32cc80bf07915058ce90722ee17bb71-Abstract.html`.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *CoRR*, abs/1811.03604, 2018. URL http://arxiv.org/abs/1811.03604.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Samuel Horváth, Maziar Sanjabi, Lin Xiao, Peter Richtárik, and Michael G. Rabbat. Fedshuffle: Recipes for better use of local work in federated learning. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://openreview.net/forum?id=Lgs5pQ1v30.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/karimireddy20a.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In Inderjit S. Dhillon, Dimitris S. Papailiopoulos, and Vivienne Sze, editors, *Proceedings of the Third Conference on Machine Learning and Systems, MLSys 2020, Austin, TX, USA, March 2-4, 2020*. mlsys.org, 2020a. URL https://proceedings.mlsys.org/paper_files/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b. URL https://openreview.net/forum?id=HJxNAnVtDS.

Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16:964–979, 1979. URL https://api.semanticscholar.org/CorpusID:120554278.

Grigory Malinovsky, Samuel Horváth, Konstantin Burlachenko, and Peter Richtárik. Federated learning with regularized client participation. *CoRR*, abs/2302.03662, 2023. doi: 10.48550/ARXIV.2302.03662. URL https://doi.org/10.48550/arXiv.2302.03662.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

Andreas Themelis and Panagiotis Patrinos. Douglas-rachford splitting and ADMM for nonconvex optimization: Tight convergence results. *SIAM J. Optim.*, 30(1):149–181, 2020. doi: 10.1137/18M1163993. URL https://doi.org/10.1137/18M1163993.

Quoc Tran-Dinh, Nhan H. Pham, Dzung T. Phan, and Lam M. Nguyen. Feddr - randomized douglas-rachford splitting algorithms for nonconvex federated composite optimization. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 30326–30338, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/fe7ee8fc1959cc7214fa21c4840dff0a-Abstract.html.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris S. Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=BkluqlSFDS.

Shiqiang Wang and Mingyue Ji. A unified analysis of federated learning with arbitrary client participation. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/79ba1b827d3fc58e129d1cbfc8ff69f2-Abstract-Conference.html.

Yuxin Wu and Kaiming He. Group normalization. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, volume 11217 of *Lecture Notes in Computer Science*, pages 3–19. Springer, 2018. doi: 10.1007/978-3-030-01261-8_1. URL https://doi.org/10.1007/978-3-030-01261-8_1.

A Derivation of *Algorithm 1* (FedCDR)

Distributed Douglas-Rachford. We begin by reformulating the standard federated learning problem with a consensus constraint into an equivalent optimization problem. To solve this problem in a distributed manner, we apply the Douglas–Rachford (DR) splitting method, which allows us to decompose the centralized formulation into local proximal updates and a server aggregation step. We further incorporate client reshuffling and allow for inexact local updates, which are practical considerations in federated learning. The resulting method is formalized in *Algorithm 1*.

Since the standard federated learning problem is formulated as (1), we can equivalently express it as the following constrained optimization problem by adding $x_1 = x_2 = \cdots = x_n$:

$$\min_{x_1,x_2,\ldots,x_n} \frac{1}{n}\sum_{i=1}^{n} f_i(x_i) \quad \text{s.t.} \quad x_1 = x_2 = \cdots = x_n, \tag{15}$$

where $\mathbf{x} := [x_1; x_2; \ldots; x_n] \in \mathbb{R}^{nd}$ represents the local parameters of all n clients. We write $\bar{f}(\mathbf{x}) := \frac{1}{n}\sum_i f_i(x_i)$ for the objective on $\mathbb{R}^{nd}$ to distinguish it from the original objective f on $\mathbb{R}^d$.

The constraint $x_1 = x_2 = \cdots = x_n$ can be expressed using the linear subspace $\mathcal{H} \subseteq \mathbb{R}^{nd}$, where

$$\mathcal{H} := \left\{\mathbf{x} \in \mathbb{R}^{nd} \mid x_1 = x_2 = \cdots = x_n\right\}.$$

To eliminate the explicit constraint in (15), we introduce the indicator function $I_{\mathcal{H}}$ associated with $\mathcal{H}$, defined as

$$I_{\mathcal{H}}(\mathbf{x}) := \begin{cases} 0, & \text{if } \mathbf{x} \in \mathcal{H}, \\ +\infty, & \text{otherwise.} \end{cases}$$

Using this indicator function, define $F(\mathbf{x}) := \bar{f}(\mathbf{x}) + I_{\mathcal{H}}(\mathbf{x})$. We can then reformulate the constrained problem as an unconstrained problem:

$$\min_{\mathbf{x}\in\mathbb{R}^{nd}} F(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^{nd}} \left\{ \frac{1}{n}\sum_{i=1}^{n} f_i(x_i) + I_{\mathcal{H}}(\mathbf{x}) \right\}. \tag{16}$$

In this setting, we apply DR splitting to separate the differentiable function $\bar{f}$ from the indicator function $I_{\mathcal{H}}$. For $0 < \eta < 1/L$, the corresponding relaxed DR iterations can be written as

$$\begin{cases} \mathbf{y}^{k+1} := \mathbf{y}^k + \alpha(\mathbf{w}^k - \mathbf{x}^k), \\ \mathbf{x}^{k+1} := \operatorname{prox}_{n\eta\bar{f}}(\mathbf{y}^{k+1}), \\ \mathbf{w}^{k+1} := \operatorname{prox}_{n\eta I_{\mathcal{H}}}(2\mathbf{x}^{k+1} - \mathbf{y}^{k+1}), \end{cases} \tag{17}$$

where k is the iteration index, $\eta > 0$ is the proximal coefficient, and $\alpha \in (0, 2]$ is a relaxation coefficient as discussed in Themelis and Patrinos [2020]. In classical DR splitting [Lions and Mercier, 1979], α is typically set to 1. For the shifted iteration order in (17), the initial states satisfy $\mathbf{x}^0 = \operatorname{prox}_{n\eta\bar{f}}(\mathbf{y}^0)$ and $\mathbf{w}^0 = \operatorname{prox}_{n\eta I_{\mathcal{H}}}(2\mathbf{x}^0 - \mathbf{y}^0)$. The convergence results in this paper use $0 < \alpha < 2$ together with the stated stepsize restrictions.

To clarify the two proximal operators, for any $\mathbf{v} \in \mathbb{R}^{nd}$ we have

$$\operatorname{prox}_{n\eta\bar{f}}(\mathbf{v}) := \arg\min_{\mathbf{z}\in\mathbb{R}^{nd}} \left\{ \bar{f}(\mathbf{z}) + \frac{1}{2n\eta}\|\mathbf{z} - \mathbf{v}\|^2 \right\},$$

and

$$\operatorname{prox}_{n\eta I_{\mathcal{H}}}(\mathbf{v}) := \arg\min_{\mathbf{z}\in\mathbb{R}^{nd}} \left\{ I_{\mathcal{H}}(\mathbf{z}) + \frac{1}{2n\eta}\|\mathbf{z} - \mathbf{v}\|^2 \right\}.$$

Since $I_{\mathcal{H}}$ is the indicator function of a linear subspace, we can simplify the second proximal step in the DR iteration as follows:

$$\begin{aligned} \mathbf{w}^{k+1} &= \operatorname{prox}_{n\eta I_{\mathcal{H}}}(2\mathbf{x}^{k+1} - \mathbf{y}^{k+1}) \\ &= \arg\min_{\mathbf{w}} \left\{ n\eta I_{\mathcal{H}}(\mathbf{w}) + \frac{1}{2}\|\mathbf{w} - (2\mathbf{x}^{k+1} - \mathbf{y}^{k+1})\|^2 \right\} \\ &\overset{(a)}{=} \arg\min_{w_1,\ldots,w_n} \left\{ \|\mathbf{w} - (2\mathbf{x}^{k+1} - \mathbf{y}^{k+1})\|^2 \;\middle|\; w_1 = \cdots = w_n \right\} \\ &\overset{(b)}{=} \arg\min_{w_1=\cdots=w_n} \sum_{i=1}^{n} \|w_i - (2x_i^{k+1} - y_i^{k+1})\|^2, \end{aligned} \tag{18}$$

where step (a) rewrites the indicator as an explicit constraint, and step (b) decomposes the norm coordinate-wise.

Defining $w := w_1$, the solution to (18) is the vector $(w; \dots; w)$, where

$$w = \arg\min_{v \in \mathbb{R}^d} \sum_{i=1}^{n} \|v - (2x_i^{k+1} - y_i^{k+1})\|^2 = \frac{1}{n} \sum_{i=1}^{n} (2x_i^{k+1} - y_i^{k+1}). \tag{19}$$

We give a detailed proof of (19).

Proof. Let $w^{(a)}$ denote the a-th coordinate of $w \in \mathbb{R}^d$. Since the squared Euclidean norm is separable across coordinates, we have

$$\begin{aligned} &\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \|w - (2x_i^{k+1} - y_i^{k+1})\|^2 \\ &\quad = \min_{w^{(1)}, \dots, w^{(d)}} \sum_{a=1}^{d} \sum_{i=1}^{n} \big(w^{(a)} - (2x_i^{k+1} - y_i^{k+1})^{(a)}\big)^2. \end{aligned}$$

For each $a \in \{1, \dots, d\}$, define $h_a(v) := \sum_{i=1}^{n} \big(v - (2x_i^{k+1} - y_i^{k+1})^{(a)}\big)^2$. The function h_a is a strictly convex quadratic, and its derivative is

$$h_a'(v) = 2nv - 2 \sum_{i=1}^{n} (2x_i^{k+1} - y_i^{k+1})^{(a)}.$$

Setting $h_a'(v) = 0$ gives

$$w^{(a)} = \frac{1}{n} \sum_{i=1}^{n} (2x_i^{k+1} - y_i^{k+1})^{(a)}.$$

Therefore, $w = \frac{1}{n} \sum_{i=1}^{n} (2x_i^{k+1} - y_i^{k+1})$, as claimed. □

Furthermore, since $\bar{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(x_i)$, we can simplify the first proximal step as follows:

$$\begin{aligned} \mathbf{x}^{k+1} &= \operatorname{prox}_{n\eta \bar{f}}(\mathbf{y}^{k+1}) \\ &= \arg\min_{\mathbf{x}} \left\{ \bar{f}(\mathbf{x}) + \frac{1}{2n\eta} \|\mathbf{x} - \mathbf{y}^{k+1}\|^2 \right\} \\ &\overset{(a)}{=} \arg\min_{\mathbf{x}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left[f_i(x_i) + \frac{1}{2\eta} \|x_i - y_i^{k+1}\|^2 \right] \right\} \\ &= \big(\operatorname{prox}_{\eta f_1}(y_1^{k+1}); \dots; \operatorname{prox}_{\eta f_n}(y_n^{k+1})\big), \end{aligned} \tag{20}$$

where step (a) follows from the separability of $\bar{f}$ and the product norm.

Therefore, we can implement the centralized DR iteration (17) in a distributed manner. Specifically, each client computes its local proximal step according to (20), while the server computes the average in (19). To streamline the iteration, introduce the auxiliary variable $\hat{x}_i^k := 2x_i^k - y_i^k$. Then the full-participation iteration takes the following form:

$$\begin{cases} y_i^{k+1} := y_i^k + \alpha(w^k - x_i^k), & i = 1, \dots, n, \\ x_i^{k+1} := \operatorname{prox}_{\eta f_i}(y_i^{k+1}), & i = 1, \dots, n, \\ \hat{x}_i^{k+1} := 2x_i^{k+1} - y_i^{k+1}, & i = 1, \dots, n, \\ w^{k+1} := \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \hat{x}_i^{k+1}. \end{cases} \tag{21}$$

The first three steps are performed independently by the clients, while the final step is executed by the server. With the consistent initial states specified above, (21) is equivalent to the centralized iteration (17).

Clients Reshuffling In our algorithm, we incorporate client reshuffling to improve client participation efficiency. At the beginning of each meta-epoch t, we generate a random permutation of all clients, denoted by $(\pi_1^t, \dots, \pi_n^t)$, where $\pi_j^t \in \{1, \dots, n\}$. The permuted list is then partitioned into batches of size C. Specifically, the batch selected at round r within meta-epoch t is

$$S_r^t := \{\pi_{rC+1}^t, \dots, \pi_{(r+1)C}^t\}, \qquad r = 0, \dots, R-1,$$

where C divides n and $R := n/C$. We denote the complete reshuffling schedule by $S^t := (S_0^t, \dots, S_{R-1}^t)$. Thus every client participates exactly once in each meta-epoch, and a new permutation is generated at the beginning of the next meta-epoch.

We now introduce the necessary notation and adapt (21) to client reshuffling. Each meta-epoch t consists of R inner communication rounds, indexed by $r \in \{0, 1, \dots, R-1\}$. In round r, only clients $i \in S_r^t$ update their local states. The global round index can be written as

$$k = tR + r, \qquad t \geq 0, \quad 0 \leq r < R.$$

This correspondence specifies the iteration order. Updating only the selected clients is a block-update version of (21), rather than merely a relabeling of the full-participation method.

First, within meta-epoch t, at round r, each selected client $i \in S_r^t$ performs the local updates

$$\begin{cases} y_{i,r+1}^t := y_{i,r}^t + \alpha(x_r^t - x_{i,r}^t), \\ x_{i,r+1}^t \approx \operatorname{prox}_{\eta f_i}(y_{i,r+1}^t), \\ \hat{x}_{i,r+1}^t := 2x_{i,r+1}^t - y_{i,r+1}^t. \end{cases} \tag{22}$$

The equality $x_{i,r+1}^t = \operatorname{prox}_{\eta f_i}(y_{i,r+1}^t)$ corresponds to the exact variant; the inexact version is specified below. For inactive clients $i \notin S_r^t$, no local computation is performed, and their states remain unchanged:

$$\begin{cases} y_{i,r+1}^t := y_{i,r}^t, \\ x_{i,r+1}^t := x_{i,r}^t, \\ \hat{x}_{i,r+1}^t := \hat{x}_{i,r}^t. \end{cases} \tag{23}$$

After the participating clients complete their local updates, the full aggregation of the reflected states takes the form

$$x_{r+1}^t = \frac{1}{n} \sum_{i=1}^n \hat{x}_{i,r+1}^t. \tag{24}$$

We next specify the initialization and show that the server can maintain this identity using only the increments sent by participating clients.

Initialization: Choose an initial parameter $x^0 \in \operatorname{dom}(f)$ and coefficients $0 < \eta < 1/L$ and $\alpha \in (0, 2]$. Set $u := x^0$. Initialize each client $i \in [n]$ with

$$\begin{aligned} x_{i,0}^0 &:= u, \\ y_{i,0}^0 &:= u + \eta \nabla f_i(u), \\ \hat{x}_{i,0}^0 &:= u - \eta \nabla f_i(u), \\ x_0^0 &:= \frac{1}{n} \sum_{i=1}^n \hat{x}_{i,0}^0 = u - \eta \nabla f(u). \end{aligned} \tag{25}$$

The input $x^0 = u$ and the initial server variable x_0^0 are therefore not necessarily equal. By the L-smoothness of f_i, the function $f_i(v) + \frac{1}{2\eta}\|v - y_{i,0}^0\|^2$ is $(1/\eta - L)$-strongly convex and coercive when $\eta < 1/L$. Its gradient vanishes at $v = u$, since

$$\nabla f_i(u) + \frac{1}{\eta}(u - y_{i,0}^0) = 0.$$

Consequently, u is its unique minimizer, and the initialization satisfies

$$x_{i,0}^0 = \operatorname{prox}_{\eta f_i}(y_{i,0}^0) = u, \qquad \hat{x}_{i,0}^0 = 2x_{i,0}^0 - y_{i,0}^0. \tag{26}$$

Thus the initial local states are consistent with the DR proximal steps, and the server state is the mean of the reflected states. This initialization requires one exact local gradient from each client and one initial aggregation. Under the participation limit of C clients per round, the latter can be carried out in R initialization batches, in addition to the TR update rounds.

At the beginning of meta-epoch $t+1$, the state variables are carried over as

$$\begin{aligned} x_0^{t+1} &:= x_R^t, \\ x_{i,0}^{t+1} &:= x_{i,R}^t, \\ y_{i,0}^{t+1} &:= y_{i,R}^t, \\ \hat{x}_{i,0}^{t+1} &:= \hat{x}_{i,R}^t. \end{aligned} \tag{27}$$

Therefore, the overall process of our algorithm can be described as follows.

Meta-Epoch t: Sample a permutation of clients and partition them into $S^t = (S_0^t, S_1^t, \ldots, S_{R-1}^t)$, where each S_r^t contains C clients.

Inner Communication Round $r \in \{0, \ldots, R-1\}$:

- **Communication:** Each active client $i \in S_r^t$ receives x_r^t from the server.
- **Active Client Update:** Each active client performs the three local updates in (22).
- **Inactive Client Update:** Each inactive client maintains its previous states as in (23).
- **Communication:** Each active client $i \in S_r^t$ sends the increment $g_{i,r}^t := \hat{x}_{i,r+1}^t - \hat{x}_{i,r}^t$ to the server.
- **Server Aggregation:** The server uses the increments to update the global model, as derived below.

To simplify the server aggregation step, we use (23) to derive

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^n \hat{x}_{i,r+1}^t &= \frac{1}{n}\sum_{i\in S_r^t} \hat{x}_{i,r+1}^t + \frac{1}{n}\sum_{i\notin S_r^t} \hat{x}_{i,r+1}^t \\ &= \frac{1}{n}\sum_{i\in S_r^t} \hat{x}_{i,r+1}^t + \frac{1}{n}\sum_{i\notin S_r^t} \hat{x}_{i,r}^t \\ &= \frac{1}{n}\sum_{i=1}^n \hat{x}_{i,r}^t + \frac{1}{n}\sum_{i\in S_r^t} (\hat{x}_{i,r+1}^t - \hat{x}_{i,r}^t). \end{aligned} \tag{28}$$

Defining $g_{i,r}^t := \hat{x}_{i,r+1}^t - \hat{x}_{i,r}^t$ and $g_r^t := \frac{1}{n}\sum_{i\in S_r^t} g_{i,r}^t$, the server implements the update

$$x_{r+1}^t := x_r^t + g_r^t = x_r^t + \frac{1}{n}\sum_{i\in S_r^t} g_{i,r}^t. \tag{29}$$

We verify that this update agrees with the aggregation identity (24). At $(t,r) = (0,0)$, (25) gives $x_0^0 = \frac{1}{n}\sum_i \hat{x}_{i,0}^0$. Suppose $x_r^t = \frac{1}{n}\sum_i \hat{x}_{i,r}^t$ holds at the beginning of a round. Then (28) and (29) give

$$x_{r+1}^t = \frac{1}{n}\sum_{i=1}^n \hat{x}_{i,r+1}^t.$$

The carryover rule (27) preserves the same relation at the beginning of the next meta-epoch. Hence, by induction,

$$x_r^t = \frac{1}{n}\sum_{i=1}^n \hat{x}_{i,r}^t \qquad \text{for every } t \geq 0 \text{ and } 0 \leq r \leq R.$$

Thus the full-average expression and the incremental server update are equivalent under the stated initialization. The simplified update rule is implemented in *Algorithm 1*.

Furthermore, we allow the proximal operator $\mathrm{prox}_{\eta f_i}$ to be evaluated approximately. In this case, the exact local proximal step is replaced by an approximation $x_{i,r}^t \approx \mathrm{prox}_{\eta f_i}(y_{i,r}^t)$. To describe this error, let

$$x_{i,r}^t = z_{i,r}^t + e_{i,r}^t, \quad \text{where} \quad z_{i,r}^t := \mathrm{prox}_{\eta f_i}(y_{i,r}^t), \quad e_{i,r}^t := x_{i,r}^t - z_{i,r}^t. \tag{30}$$

Here $e_{i,r}^t$ denotes the proximal approximation error. The reference point $z_{i,r}^t$ is introduced only for the analysis; it need not be computed in the actual implementation of *Algorithm 1*. We assume deterministic nonnegative error bounds $\epsilon_{t,i,r}$ such that

$$\mathbb{E}\big[\|e_{i,r}^t\|^2\big] \le \epsilon_{t,i,r}^2. \tag{31}$$

The expectation is over client reshuffling and any randomness of the local solver. In particular, a pathwise accuracy bound $\|e_{i,r}^t\| \le \epsilon_{t,i,r}$ is sufficient for (31).

We also conceptually introduce the reference states at initialization, at meta-epoch boundaries, and after each inner communication round:

$$\begin{aligned} z_{i,0}^0 &:= \mathrm{prox}_{\eta f_i}(y_{i,0}^0) = u, \\ z_{i,0}^{t+1} &:= z_{i,R}^t, \\ z_{i,r+1}^t &:= \begin{cases} \mathrm{prox}_{\eta f_i}(y_{i,r+1}^t), & \text{if } i \in S_r^t, \\ z_{i,r}^t, & \text{if } i \notin S_r^t. \end{cases} \end{aligned} \tag{32}$$

Since inactive clients keep $y_{i,r}^t$ unchanged, their exact proximal reference $z_{i,r}^t$ remains unchanged as well. The same holds for $e_{i,r}^t$ by (30), and both states are carried over between meta-epochs. The initialization in (26) gives $e_{i,0}^0 = 0$. In the exact variant, $e_{i,r}^t = 0$ for all clients and rounds; in the inexact variant, the bounds in (31) must also cover errors inherited by inactive clients.

B Proofs of Convergence Analysis for the Inexact Variant of *Algorithm 1*

In this section, we prove Theorem 1 and Corollary 2. Throughout the proof, Assumptions 1 and 2 hold. We choose $0 < \alpha < 2$, $0 < \gamma < (2-\alpha)/(2\alpha)$, and $0 < \eta < \eta_0$, where

$$\eta_0 := \frac{2-\alpha-2\alpha\gamma}{2(1+\alpha\gamma)L} > 0. \tag{33}$$

Since $\eta < \eta_0$, we have

$$\begin{aligned} &2-\alpha-\alpha\eta L-2\eta^2L^2-2\alpha\gamma(1+\eta L)^2 \\ &= (1+\eta L)\big[2-\alpha-2\alpha\gamma-2(1+\alpha\gamma)\eta L\big] > 0. \end{aligned} \tag{34}$$

In particular, $\eta_0 < 1/L$, so the proximal subproblems have unique minimizers. We establish the following inequalities for every realized client order before taking expectations.

Lemma 5. *Let $\{(x_{i,r}^t, y_{i,r}^t, z_{i,r}^t, \hat{x}_{i,r}^t)\}$ be the sequence generated by Algorithm 1 with initialization (25). Then, for all $i \in [n]$, $t \geq 0$, and $0 \leq r \leq R$, the following equalities hold:*

$$y_{i,r}^t = z_{i,r}^t + \eta\nabla f_i(z_{i,r}^t), \qquad \hat{x}_{i,r}^t = 2x_{i,r}^t - y_{i,r}^t. \tag{35}$$

Proof. We prove (35) by induction.

Base case: By (25) and (26), $z_{i,0}^0 = x_{i,0}^0 = u$ and $y_{i,0}^0 = u+\eta\nabla f_i(u)$. Moreover, $\hat{x}_{i,0}^0 = 2x_{i,0}^0 - y_{i,0}^0$. Thus both equalities hold at $(t,r) = (0,0)$.

Inductive step: Suppose that both equalities hold at round r. We consider two cases:

- **Case 1:** If $i \in S_r^t$, then $z_{i,r+1}^t = \mathrm{prox}_{\eta f_i}(y_{i,r+1}^t)$. The proximal optimality condition gives
$$y_{i,r+1}^t = z_{i,r+1}^t + \eta\nabla f_i(z_{i,r+1}^t).$$
The client reflection update also gives $\hat{x}_{i,r+1}^t = 2x_{i,r+1}^t - y_{i,r+1}^t$.
- **Case 2:** If $i \notin S_r^t$, then (23) and (32) keep all four states unchanged. The induction hypothesis therefore applies.

Finally, (27) carries the identities from (t,R) to $(t+1,0)$, completing the proof. □

Lemma 6. *Let $\{(x_{i,r}^t, y_{i,r}^t, z_{i,r}^t, x_r^t)\}$ be generated by Algorithm 1. For every active client $i \in S_r^t$, we have*

$$\begin{aligned} \|x_r^t - x_{i,r}^t\|^2 \leq \frac{(1+\eta L)^2}{\alpha^2}\big[&2\|x_{i,r+1}^t - x_{i,r}^t\|^2 \\ &+ 4(\|e_{i,r}^t\|^2 + \|e_{i,r+1}^t\|^2)\big]. \end{aligned} \tag{36}$$

Proof. By the update of $y_{i,r+1}^t$ and Lemma 5,

$$\begin{aligned} x_r^t - x_{i,r}^t &= \frac{1}{\alpha}(y_{i,r+1}^t - y_{i,r}^t) \\ &= \frac{1}{\alpha}(z_{i,r+1}^t - z_{i,r}^t) + \frac{\eta}{\alpha}\big[\nabla f_i(z_{i,r+1}^t) - \nabla f_i(z_{i,r}^t)\big]. \end{aligned} \tag{37}$$

Therefore,

$$\begin{aligned} \|x_r^t - x_{i,r}^t\|^2 &\overset{(a)}{\leq} \frac{(1+\eta L)^2}{\alpha^2}\|z_{i,r+1}^t - z_{i,r}^t\|^2 \\ &\overset{(30)}{=} \frac{(1+\eta L)^2}{\alpha^2}\|x_{i,r+1}^t - x_{i,r}^t - (e_{i,r+1}^t - e_{i,r}^t)\|^2 \\ &\overset{(b)}{\leq} \frac{(1+\eta L)^2}{\alpha^2}\left[2\|x_{i,r+1}^t - x_{i,r}^t\|^2 + 4(\|e_{i,r}^t\|^2 + \|e_{i,r+1}^t\|^2)\right], \end{aligned} \tag{38}$$

where (a) uses L-smoothness and the triangle inequality, and (b) uses $\|v+w\|^2 \leq 2\|v\|^2 + 2\|w\|^2$. □

Lemma 7. *Let $\{(x_{i,r}^t, y_{i,r}^t, z_{i,r}^t, \hat{x}_{i,r}^t, x_r^t)\}$ be generated by Algorithm 1. Then,*

$$\begin{aligned}\|\nabla f(x_r^t)\| \le \frac{1+\eta L}{\eta}\Bigg[\left(\frac{1}{n}\sum_{i=1}^n \|x_r^t - x_{i,r}^t\|^2\right)^{1/2} \\ + \left(\frac{1}{n}\sum_{i=1}^n \|e_{i,r}^t\|^2\right)^{1/2}\Bigg].\end{aligned} \tag{39}$$

Consequently,

$$\|\nabla f(x_r^t)\|^2 \le \frac{(1+\eta L)^2}{n\eta^2}\sum_{i=1}^n \left[2\|x_r^t - x_{i,r}^t\|^2 + 2\|e_{i,r}^t\|^2\right]. \tag{40}$$

Proof. By the aggregation step in *Algorithm 1* and Lemma 5,

$$\begin{aligned}x_r^t &= \frac{1}{n}\sum_{i=1}^n \hat{x}_{i,r}^t = \frac{1}{n}\sum_{i=1}^n (2x_{i,r}^t - y_{i,r}^t) \\ &= \frac{1}{n}\sum_{i=1}^n \big[2x_{i,r}^t - z_{i,r}^t - \eta\nabla f_i(z_{i,r}^t)\big].\end{aligned} \tag{41}$$

Rearranging this *averaged* equality gives

$$\begin{aligned}\nabla f(x_r^t) = \frac{1}{n}\sum_{i=1}^n \big[\nabla f_i(x_r^t) - \nabla f_i(z_{i,r}^t)\big] \\ + \frac{1}{n\eta}\sum_{i=1}^n (2x_{i,r}^t - z_{i,r}^t - x_r^t).\end{aligned} \tag{42}$$

Using L-smoothness and (30),

$$\begin{aligned}\|\nabla f(x_r^t)\| &\overset{(a)}{\le} \frac{1}{n\eta}\sum_{i=1}^n \big[\eta L\|x_r^t - z_{i,r}^t\| + \|2x_{i,r}^t - z_{i,r}^t - x_r^t\|\big] \\ &\overset{(b)}{\le} \frac{1+\eta L}{n\eta}\sum_{i=1}^n \big(\|x_r^t - x_{i,r}^t\| + \|e_{i,r}^t\|\big),\end{aligned} \tag{43}$$

where (a) follows from the triangle inequality, and (b) applies the same inequality after $z_{i,r}^t = x_{i,r}^t - e_{i,r}^t$. Cauchy–Schwarz proves (39); the squared inequality follows from $(v+w)^2 \le 2v^2 + 2w^2$. □

Definition 4. Define the potential function $\mathcal{D}_r^t$ as

$$\mathcal{D}_r^t := \frac{1}{n}\sum_{i=1}^n \left[f_i(x_{i,r}^t) + \langle \nabla f_i(x_{i,r}^t), x_r^t - x_{i,r}^t\rangle + \frac{1}{2\eta}\|x_r^t - x_{i,r}^t\|^2\right]. \tag{44}$$

For continuity of the definition, we set $\mathcal{D}_R^t := \mathcal{D}_0^{t+1}$, consistent with (27).

By the L-smoothness of f_i and $\eta < 1/L$,

$$\begin{aligned}\mathcal{D}_r^t &\ge \frac{1}{n}\sum_{i=1}^n \left[f_i(x_r^t) + \left(\frac{1}{2\eta} - \frac{L}{2}\right)\|x_r^t - x_{i,r}^t\|^2\right] \\ &\ge f(x_r^t) \ge f^*.\end{aligned} \tag{45}$$

Moreover, the initialization (25) gives $x_{i,0}^0 = u$ and $x_0^0 = u - \eta\nabla f(u)$, so

$$\begin{aligned}\mathcal{D}_0^0 &= f(u) + \langle \nabla f(u), -\eta\nabla f(u)\rangle + \frac{\eta}{2}\|\nabla f(u)\|^2 \\ &= f(u) - \frac{\eta}{2}\|\nabla f(u)\|^2 \le f(u) = f(x^0).\end{aligned} \tag{46}$$

Lemma 8. *Let $\mathcal{D}_r^t$ be the potential function defined in Definition 4. Then,*

$$\begin{aligned}\mathcal{D}_{r+1}^t \le \mathcal{D}_r^t &- \mathcal{C}_0 \sum_{i\in S_r^t} \|x_{i,r+1}^t - x_{i,r}^t\|^2 \\ &+ \frac{\mathcal{C}_1}{n}\sum_{i\in S_r^t}\|e_{i,r}^t\|^2 + \frac{\mathcal{C}_2}{n}\sum_{i\in S_r^t}\|e_{i,r+1}^t\|^2 \\ &+ \frac{\mathcal{C}_3}{n}\sum_{i\notin S_r^t}\|e_{i,r}^t\|^2,\end{aligned} \tag{47}$$

where the constants are

$$\begin{aligned}\mathcal{C}_0 &:= \frac{2-\alpha-\alpha\eta L - 2\eta^2L^2 - 2\alpha\gamma(1+\eta L)^2}{2n\alpha\eta} > 0,\\ \mathcal{C}_1 &:= \frac{(1+\eta L)^2}{\gamma\eta\alpha^2},\\ \mathcal{C}_2 &:= \mathcal{C}_1 + \frac{(1+\eta L)^2}{2\eta}, \qquad \mathcal{C}_3 := \frac{(1+\eta L)^2}{2\eta}.\end{aligned} \tag{48}$$

Proof. We first establish the following identity from Lemma 5 and (30):

$$\begin{aligned}&x_{i,r}^t + \eta\nabla f_i(x_{i,r}^t) - y_{i,r}^t \\ &= e_{i,r}^t + \eta\big[\nabla f_i(x_{i,r}^t) - \nabla f_i(z_{i,r}^t)\big],\\ &\|x_{i,r}^t + \eta\nabla f_i(x_{i,r}^t) - y_{i,r}^t\| \le (1+\eta L)\|e_{i,r}^t\|.\end{aligned} \tag{49}$$

First, consider the change of the server variable. Expanding the quadratic term in Definition 4 and using $x_{r+1}^t = n^{-1}\sum_i \hat{x}_{i,r+1}^t$, we obtain

$$\begin{aligned}\mathcal{D}_{r+1}^t &= \frac{1}{n}\sum_{i=1}^n\left[f_i(x_{i,r+1}^t) + \langle\nabla f_i(x_{i,r+1}^t), x_r^t - x_{i,r+1}^t\rangle + \frac{1}{2\eta}\|x_r^t - x_{i,r+1}^t\|^2\right]\\ &\quad - \frac{1}{2\eta}\|x_{r+1}^t - x_r^t\|^2\\ &\quad + \frac{1}{n\eta}\sum_{i=1}^n\left\langle x_{i,r+1}^t + \eta\nabla f_i(x_{i,r+1}^t) - y_{i,r+1}^t,\, x_{r+1}^t - x_r^t\right\rangle\\ &\overset{(a)}{\le} \frac{1}{n}\sum_{i=1}^n\left[f_i(x_{i,r+1}^t) + \langle\nabla f_i(x_{i,r+1}^t), x_r^t - x_{i,r+1}^t\rangle + \frac{1}{2\eta}\|x_r^t - x_{i,r+1}^t\|^2\right]\\ &\quad + \frac{(1+\eta L)^2}{2n\eta}\sum_{i=1}^n\|e_{i,r+1}^t\|^2,\end{aligned} \tag{50}$$

where (a) uses Young's inequality, Jensen's inequality, and (49).

Next, let $i \in S_r^t$. The update of $y_{i,r+1}^t$ implies

$$\begin{aligned}&\alpha(x_r^t - x_{i,r}^t) - (x_{i,r+1}^t - x_{i,r}^t)\\ &\quad - \eta\big[\nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t)\big]\\ &= (x_{i,r}^t + \eta\nabla f_i(x_{i,r}^t) - y_{i,r}^t)\\ &\quad - (x_{i,r+1}^t + \eta\nabla f_i(x_{i,r+1}^t) - y_{i,r+1}^t).\end{aligned} \tag{51}$$

Hence, by (49),

$$\begin{aligned}&\Big\|\alpha(x_r^t - x_{i,r}^t) - (x_{i,r+1}^t - x_{i,r}^t)\\ &\qquad - \eta[\nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t)]\Big\|^2\\ &\le 2(1+\eta L)^2\big(\|e_{i,r}^t\|^2 + \|e_{i,r+1}^t\|^2\big).\end{aligned} \tag{52}$$

Using L-smoothness in the form

$$f_i(x_{i,r+1}^t) - f_i(x_{i,r}^t) \le \langle \nabla f_i(x_{i,r+1}^t), x_{i,r+1}^t - x_{i,r}^t \rangle + \frac{L}{2}\|x_{i,r+1}^t - x_{i,r}^t\|^2,$$

we obtain the following local potential estimate:

$$\begin{aligned}
&f_i(x_{i,r+1}^t) + \langle \nabla f_i(x_{i,r+1}^t), x_r^t - x_{i,r+1}^t \rangle + \frac{1}{2\eta}\|x_r^t - x_{i,r+1}^t\|^2 \\
&\quad - f_i(x_{i,r}^t) - \langle \nabla f_i(x_{i,r}^t), x_r^t - x_{i,r}^t \rangle - \frac{1}{2\eta}\|x_r^t - x_{i,r}^t\|^2 \\
&\le -\frac{2-\alpha-\alpha\eta L - 2\eta^2 L^2}{2\alpha\eta}\|x_{i,r+1}^t - x_{i,r}^t\|^2 \\
&\quad + \frac{1}{\alpha\eta}\Big\langle \alpha(x_r^t - x_{i,r}^t) - (x_{i,r+1}^t - x_{i,r}^t) - \eta[\nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t)], \\
&\qquad\qquad \eta[\nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t)] - (x_{i,r+1}^t - x_{i,r}^t) \Big\rangle.
\end{aligned} \tag{53}$$

In the last step, the mixed inner products cancel; we also used $\|\nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t)\| \le L\|x_{i,r+1}^t - x_{i,r}^t\|$. Applying Young's inequality to the inner product in (53), followed by (52), gives

$$\begin{aligned}
&\frac{1}{\alpha\eta}\Big\langle \alpha(x_r^t - x_{i,r}^t) - (x_{i,r+1}^t - x_{i,r}^t) - \eta[\nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t)], \\
&\qquad\qquad \eta[\nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t)] - (x_{i,r+1}^t - x_{i,r}^t) \Big\rangle \\
&\le \frac{(1+\eta L)^2}{\gamma\eta\alpha^2}\big(\|e_{i,r}^t\|^2 + \|e_{i,r+1}^t\|^2\big) + \frac{\gamma(1+\eta L)^2}{2\eta}\|x_{i,r+1}^t - x_{i,r}^t\|^2.
\end{aligned} \tag{54}$$

Since $\mathcal{C}_0$ is defined with the smaller descent coefficient in (48), (53)–(54) imply

$$\begin{aligned}
&f_i(x_{i,r+1}^t) + \langle \nabla f_i(x_{i,r+1}^t), x_r^t - x_{i,r+1}^t \rangle + \frac{1}{2\eta}\|x_r^t - x_{i,r+1}^t\|^2 \\
&\quad - f_i(x_{i,r}^t) - \langle \nabla f_i(x_{i,r}^t), x_r^t - x_{i,r}^t \rangle - \frac{1}{2\eta}\|x_r^t - x_{i,r}^t\|^2 \\
&\le -n\mathcal{C}_0\|x_{i,r+1}^t - x_{i,r}^t\|^2 + \mathcal{C}_1\big(\|e_{i,r}^t\|^2 + \|e_{i,r+1}^t\|^2\big).
\end{aligned} \tag{55}$$

For $i \notin S_r^t$, $x_{i,r+1}^t = x_{i,r}^t$ and $e_{i,r+1}^t = e_{i,r}^t$. Summing (55) over all active clients, and using (50), yields (47). □

For each client i, let $\tau_i^t \in \{0, \dots, R-1\}$ denote its unique participation round in meta-epoch t. This index is well defined because the client batches form a partition of $[n]$.

Lemma 9. *For every complete client traversal,*

$$\begin{aligned}
&\mathcal{C}_0 \sum_{i=1}^{n} \|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2 \\
&\le \mathcal{D}_0^t - \mathcal{D}_R^t + \frac{\max\{\mathcal{C}_1, \mathcal{C}_3\}}{n}\sum_{r=0}^{R-1}\sum_{i=1}^{n}\|e_{i,r}^t\|^2 \\
&\quad + \frac{\mathcal{C}_2}{n}\sum_{r=0}^{R-1}\sum_{i=1}^{n}\|e_{i,r+1}^t\|^2.
\end{aligned} \tag{56}$$

Proof. Summing (47) over $r = 0, \dots, R-1$ yields

$$\begin{aligned}\mathcal{D}_R^t - \mathcal{D}_0^t \le &- \mathcal{C}_0 \sum_{r=0}^{R-1} \sum_{i \in S_r^t} \|x_{i,r+1}^t - x_{i,r}^t\|^2 \\ &+ \frac{\mathcal{C}_1}{n} \sum_{r=0}^{R-1} \sum_{i \in S_r^t} \|e_{i,r}^t\|^2 + \frac{\mathcal{C}_2}{n} \sum_{r=0}^{R-1} \sum_{i \in S_r^t} \|e_{i,r+1}^t\|^2 \\ &+ \frac{\mathcal{C}_3}{n} \sum_{r=0}^{R-1} \sum_{i \notin S_r^t} \|e_{i,r}^t\|^2.\end{aligned} \tag{57}$$

Since each client is active in exactly one round,

$$\sum_{r=0}^{R-1} \sum_{i \in S_r^t} \|x_{i,r+1}^t - x_{i,r}^t\|^2 = \sum_{i=1}^{n} \|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2.$$

Bounding the two sums of old errors by $\max\{\mathcal{C}_1, \mathcal{C}_3\}$ times the all-client sum proves (56). □

Unlike a bound obtained by conditioning on one remaining batch, (56) holds for every realized permutation. We next bound the gradient over the same complete traversal.

Lemma 10. *For $0 < \eta < 1/L$ and $0 < \alpha \le 2$, every complete client traversal satisfies*

$$\begin{aligned}&\frac{1}{R} \sum_{r=0}^{R-1} \|\nabla f(x_r^t)\|^2 \\ &\le \frac{2(1+\eta L)^4 (1+\alpha^{-1})^2}{n\eta^2} \sum_{i=1}^{n} \|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2 \\ &\quad + \frac{8(1+\eta L)^4 (1+\alpha^{-1})^2 + 4(1+\eta L)^2}{n\eta^2} \sum_{i=1}^{n} \big(\|e_{i,\tau_i^t}^t\|^2 + \|e_{i,\tau_i^t+1}^t\|^2\big).\end{aligned} \tag{58}$$

Proof. Fix a meta-epoch t. By the reflection update and (49), the increment $g_{i,\tau_i^t}^t$ already defined in (29) satisfies

$$\begin{aligned}g_{i,\tau_i^t}^t &= \hat{x}_{i,\tau_i^t+1}^t - \hat{x}_{i,\tau_i^t}^t \\ &= (x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t) - \eta\big[\nabla f_i(x_{i,\tau_i^t+1}^t) - \nabla f_i(x_{i,\tau_i^t}^t)\big] \\ &\quad + (x_{i,\tau_i^t+1}^t + \eta \nabla f_i(x_{i,\tau_i^t+1}^t) - y_{i,\tau_i^t+1}^t) \\ &\quad - (x_{i,\tau_i^t}^t + \eta \nabla f_i(x_{i,\tau_i^t}^t) - y_{i,\tau_i^t}^t).\end{aligned} \tag{59}$$

Therefore, using L-smoothness,

$$\begin{aligned}\|g_{i,\tau_i^t}^t\| \le (1+\eta L)\big[&\|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\| \\ &+ \|e_{i,\tau_i^t}^t\| + \|e_{i,\tau_i^t+1}^t\|\big].\end{aligned} \tag{60}$$

By (29), the difference between any two server states x_r^t, x_q^t is the sum, up to sign, of $g_{i,\tau_i^t}^t/n$ over the clients visited between these rounds. No client appears twice. Cauchy–Schwarz and (60) imply

$$\begin{aligned}\|x_r^t - x_q^t\| &\le \left(\frac{1}{n} \sum_{i=1}^{n} \|g_{i,\tau_i^t}^t\|^2\right)^{1/2} \\ &\le (1+\eta L)\left[\left(\frac{1}{n} \sum_{i=1}^{n} \|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2\right)^{1/2}\right. \\ &\qquad \left. + \left(\frac{2}{n} \sum_{i=1}^{n} (\|e_{i,\tau_i^t}^t\|^2 + \|e_{i,\tau_i^t+1}^t\|^2)\right)^{1/2}\right].\end{aligned} \tag{61}$$

For a fixed client i, we consider its state before and after the unique visit.

- **Case 1:** If $r \le \tau_i^t$, then $x_{i,r}^t = x_{i,\tau_i^t}^t$ and

$$x_{\tau_i^t}^t - x_{i,r}^t = \frac{y_{i,\tau_i^t+1}^t - y_{i,\tau_i^t}^t}{\alpha}.$$

- **Case 2:** If $r > \tau_i^t$, then $x_{i,r}^t = x_{i,\tau_i^t+1}^t$, and hence

$$x_{\tau_i^t}^t - x_{i,r}^t = \frac{y_{i,\tau_i^t+1}^t - y_{i,\tau_i^t}^t}{\alpha} - (x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t).$$

Using (51), (49), and $|1-\alpha| \le 1$, both cases give

$$\begin{aligned}\|x_{\tau_i^t}^t - x_{i,r}^t\| \le \frac{1+\eta L}{\alpha}\big[&\|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\| \\ &+ \|e_{i,\tau_i^t}^t\| + \|e_{i,\tau_i^t+1}^t\|\big].\end{aligned} \tag{62}$$

Combining (61) and (62), and using the triangle inequality, yields

$$\begin{aligned}&\left(\frac{1}{n}\sum_{i=1}^n \|x_r^t - x_{i,r}^t\|^2\right)^{1/2} \\ &\le (1+\eta L)(1+\alpha^{-1})\Bigg[\left(\frac{1}{n}\sum_{i=1}^n \|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2\right)^{1/2} \\ &\qquad + \left(\frac{2}{n}\sum_{i=1}^n (\|e_{i,\tau_i^t}^t\|^2 + \|e_{i,\tau_i^t+1}^t\|^2)\right)^{1/2}\Bigg].\end{aligned} \tag{63}$$

At round r, the error of client i is either $e_{i,\tau_i^t}^t$ or $e_{i,\tau_i^t+1}^t$. Therefore,

$$\frac{1}{n}\sum_{i=1}^n \|e_{i,r}^t\|^2 \le \frac{1}{n}\sum_{i=1}^n \big(\|e_{i,\tau_i^t}^t\|^2 + \|e_{i,\tau_i^t+1}^t\|^2\big). \tag{64}$$

Substituting (63) and (64) into (39), and using $(v+w)^2 \le 2v^2 + 2w^2$ and

$$\big[\sqrt{2}(1+\eta L)(1+\alpha^{-1}) + 1\big]^2 \le 4(1+\eta L)^2(1+\alpha^{-1})^2 + 2,$$

we obtain

$$\begin{aligned}\|\nabla f(x_r^t)\|^2 \le{}& \frac{2(1+\eta L)^4(1+\alpha^{-1})^2}{n\eta^2}\sum_{i=1}^n \|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2 \\ &+ \frac{8(1+\eta L)^4(1+\alpha^{-1})^2 + 4(1+\eta L)^2}{n\eta^2}\sum_{i=1}^n (\|e_{i,\tau_i^t}^t\|^2 + \|e_{i,\tau_i^t+1}^t\|^2).\end{aligned} \tag{65}$$

The bound holds for each r and hence also for the average over $r = 0, \dots, R-1$. □

Define the constants in Theorem 1 by

$$\begin{aligned}\mathcal{A}_1 &:= \frac{2(1+\eta L)^4(1+\alpha^{-1})^2}{n\eta^2\mathcal{C}_0}, \\ \mathcal{B}_1 &:= \max\{\mathcal{C}_1, \mathcal{C}_3\} + \frac{8(1+\eta L)^4(1+\alpha^{-1})^2 + 4(1+\eta L)^2}{\eta^2\mathcal{A}_1}, \\ \mathcal{B}_2 &:= \mathcal{C}_2 + \frac{8(1+\eta L)^4(1+\alpha^{-1})^2 + 4(1+\eta L)^2}{\eta^2\mathcal{A}_1}.\end{aligned} \tag{66}$$

Lemma 11. *For every meta-epoch t, the following inequality holds:*

$$\begin{aligned}&\frac{1}{R}\sum_{r=0}^{R-1} \|\nabla f(x_r^t)\|^2 \\ &\le \mathcal{A}_1(\mathcal{D}_0^t - \mathcal{D}_R^t) \\ &\quad + \frac{\mathcal{A}_1\mathcal{B}_1}{n}\sum_{r=0}^{R-1}\sum_{i=1}^n \|e_{i,r}^t\|^2 + \frac{\mathcal{A}_1\mathcal{B}_2}{n}\sum_{r=0}^{R-1}\sum_{i=1}^n \|e_{i,r+1}^t\|^2.\end{aligned} \tag{67}$$

Proof. By Lemma 10, we have

$$\begin{aligned}
&\frac{1}{R}\sum_{r=0}^{R-1}\|\nabla f(x_r^t)\|^2 \\
&\le \frac{2(1+\eta L)^4(1+\alpha^{-1})^2}{n\eta^2}\sum_{i=1}^{n}\|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2 \\
&\quad + \frac{8(1+\eta L)^4(1+\alpha^{-1})^2 + 4(1+\eta L)^2}{n\eta^2}\sum_{i=1}^{n}(\|e_{i,\tau_i^t}^t\|^2 + \|e_{i,\tau_i^t+1}^t\|^2).
\end{aligned} \tag{68}$$

The first sum is bounded by (56). Moreover, each endpoint error belongs to the corresponding all-client error sum:

$$\begin{aligned}
&\sum_{i=1}^{n}(\|e_{i,\tau_i^t}^t\|^2 + \|e_{i,\tau_i^t+1}^t\|^2) \\
&\le \sum_{r=0}^{R-1}\sum_{i=1}^{n}(\|e_{i,r}^t\|^2 + \|e_{i,r+1}^t\|^2).
\end{aligned} \tag{69}$$

Combining these inequalities and using (66) proves (67). All the preceding inequalities are pathwise; no conditional expectation over the current reshuffling batch is used. □

Definition 5. (Meta-epoch Average Gradient Norm) We define

$$\mathcal{G}^t := \frac{1}{R}\sum_{r=0}^{R-1}\|\nabla f(x_r^t)\|^2. \tag{70}$$

Proof of Theorem 1. By Lemma 11 and Definition 5, we have

$$\begin{aligned}
\mathcal{G}^t \le& \mathcal{A}_1(\mathcal{D}_0^t - \mathcal{D}_R^t) \\
&+ \frac{\mathcal{A}_1}{n}\sum_{r=0}^{R-1}\sum_{i=1}^{n}\big[\mathcal{B}_1\|e_{i,r}^t\|^2 + \mathcal{B}_2\|e_{i,r+1}^t\|^2\big].
\end{aligned} \tag{71}$$

Summing (71) from $t=0$ to $T-1$ and using $\mathcal{D}_R^t = \mathcal{D}_0^{t+1}$ gives

$$\begin{aligned}
\sum_{t=0}^{T-1}\mathcal{G}^t \le& \mathcal{A}_1(\mathcal{D}_0^0 - \mathcal{D}_0^T) \\
&+ \frac{\mathcal{A}_1}{n}\sum_{t=0}^{T-1}\sum_{r=0}^{R-1}\sum_{i=1}^{n}(\mathcal{B}_1\|e_{i,r}^t\|^2 + \mathcal{B}_2\|e_{i,r+1}^t\|^2).
\end{aligned} \tag{72}$$

By the initialization (25),

$$\mathcal{D}_0^0 = f(u) - \frac{\eta}{2}\|\nabla f(u)\|^2 \le f(x^0). \tag{73}$$

Furthermore, the L-smoothness of f_i and $\eta < 1/L$ yield

$$\begin{aligned}
\mathcal{D}_r^t &= \frac{1}{n}\sum_{i=1}^{n}\left[f_i(x_{i,r}^t) + \langle\nabla f_i(x_{i,r}^t), x_r^t - x_{i,r}^t\rangle + \frac{1}{2\eta}\|x_r^t - x_{i,r}^t\|^2\right] \\
&\ge \frac{1}{n}\sum_{i=1}^{n}\left[f_i(x_r^t) + \left(\frac{1}{2\eta} - \frac{L}{2}\right)\|x_r^t - x_{i,r}^t\|^2\right] \\
&\ge f(x_r^t) \ge f^*.
\end{aligned} \tag{74}$$

Applying these two bounds to (72), dividing by T, and taking expectation over all client orders and local-solver randomness gives

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\mathcal{G}^t] \le& \frac{\mathcal{A}_1}{T}[f(x^0) - f^*] \\
&+ \frac{\mathcal{A}_1}{nT}\sum_{t=0}^{T-1}\sum_{r=0}^{R-1}\sum_{i=1}^{n}\big[\mathcal{B}_1\mathbb{E}\|e_{i,r}^t\|^2 + \mathcal{B}_2\mathbb{E}\|e_{i,r+1}^t\|^2\big].
\end{aligned} \tag{75}$$

Let $\tilde{x}$ be selected independently and uniformly from $\{x_r^t : 0 \le t \le T-1,\ 0 \le r \le R-1\}$. By Definition 5,

$$\begin{aligned}\mathbb{E}\|\nabla f(\tilde{x})\|^2 &= \frac{1}{TR}\sum_{t=0}^{T-1}\sum_{r=0}^{R-1}\mathbb{E}\|\nabla f(x_r^t)\|^2 \\ &= \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\mathcal{G}^t].\end{aligned} \tag{76}$$

Using (31) proves the first claim (7) of Theorem 1.

Under the error budget (8), both sums of deterministic squared errors satisfy

$$\begin{aligned}\frac{1}{n}\sum_{t=0}^{T-1}\sum_{r=0}^{R-1}\sum_{i=1}^{n}\epsilon_{t,i,r}^2 \le M, \\ \frac{1}{n}\sum_{t=0}^{T-1}\sum_{r=0}^{R-1}\sum_{i=1}^{n}\epsilon_{t,i,r+1}^2 \le M.\end{aligned} \tag{77}$$

Consequently,

$$\mathbb{E}\|\nabla f(\tilde{x})\|^2 \le \frac{\mathcal{A}_1}{T}\big[f(x^0) - f^* + (\mathcal{B}_1+\mathcal{B}_2)M\big]. \tag{78}$$

Therefore, an ϵ-approximate stationary point is obtained whenever

$$T \ge \max\left\{1, \left\lceil \frac{\mathcal{A}_1[f(x^0) - f^* + (\mathcal{B}_1+\mathcal{B}_2)M]}{\epsilon} \right\rceil\right\}.$$

This proves the stated $O(\epsilon^{-1})$ meta-epoch complexity. □

A sufficient proximal-error schedule. For a concrete sufficient condition on the local solver, let the new squared proximal error generated by each client in meta-epoch t have conditional expectation at most

$$\beta_t^2 := \frac{M}{4R(t+1)^2}.$$

The initial error is zero by (26). Each newly generated error remains unchanged until that client is updated again. Counting states $r = 0, \dots, R$, including duplicated meta-epoch boundaries, each generated error appears at most $2R$ times. Therefore,

$$\begin{aligned}\frac{1}{n}\sum_{t=0}^{\infty}\sum_{r=0}^{R}\sum_{i=1}^{n}\mathbb{E}\|e_{i,r}^t\|^2 &\le 2R\sum_{t=0}^{\infty}\beta_t^2 \\ &= \frac{M}{2}\sum_{t=0}^{\infty}\frac{1}{(t+1)^2} = \frac{M\pi^2}{12} \le M.\end{aligned} \tag{79}$$

An unconditional expected error bound at each update gives the same conclusion.

To obtain deterministic bounds in (31), set $\beta_{-1}^2 = 0$ and

$$\epsilon_{t,i,r}^2 := \begin{cases}\beta_{t-1}^2, & r = 0, \\ \beta_{t-1}^2 + \beta_t^2, & 1 \le r \le R-1, \\ \beta_t^2, & r = R.\end{cases} \tag{80}$$

These bounds hold whether a client has already been visited or not. For every meta-epoch,

$$\frac{1}{n}\sum_{r=0}^{R}\sum_{i=1}^{n}\epsilon_{t,i,r}^2 = R(\beta_{t-1}^2 + \beta_t^2).$$

Summing over $t \ge 0$ gives $M\pi^2/12 \le M$, also when $R = 1$ or $M = 0$. A fixed local SGD budget does not by itself guarantee this summability condition.

Proof of Corollary 2. Choosing $\alpha = 1$, $\eta = 1/(4L)$, and $\gamma = 1/25$, we have $\eta_0 = 23/(52L) > \eta$. From (48),

$$\mathcal{C}_0 = \frac{L}{n}, \quad \mathcal{C}_1 = \frac{625}{4}L, \quad \mathcal{C}_2 = \frac{1275}{8}L, \quad \mathcal{C}_3 = \frac{25}{8}L.$$

Thus (66) gives

$$\begin{aligned} \mathcal{A}_1 &= \frac{625}{2}L, \\ \mathcal{B}_1 = \frac{16057}{100}L, &\qquad \mathcal{B}_2 = \frac{32739}{200}L, \\ \mathcal{B}_1 + \mathcal{B}_2 &= \frac{64853}{200}L \leq 325L. \end{aligned} \tag{81}$$

Substituting these constants into (78) proves (9) and the stated meta-epoch complexity. □

C Proofs of Convergence Analysis for the Exact Variant of *Algorithm 1*

In this section, we prove Theorem 3 and Corollary 4. We consider the exact proximal updates of *Algorithm 1* under Assumptions 1 and 2. Let $0 < \alpha < 2$ and choose $0 < \eta < \eta_1$, where

$$\eta_1 := \frac{2-\alpha}{2L}. \tag{82}$$

Then $\eta < 1/L$ and

$$2 - \alpha - \alpha\eta L - 2\eta^2 L^2 = (1+\eta L)(2-\alpha-2\eta L) > 0. \tag{83}$$

The proof below holds for every realized client order.

Lemma 12. *Let $\{(x_{i,r}^t, y_{i,r}^t, \hat{x}_{i,r}^t)\}$ be generated by the exact variant of Algorithm 1 with initialization (25). For all $i \in [n]$, $t \geq 0$, and $0 \leq r \leq R$, we have*

$$y_{i,r}^t = x_{i,r}^t + \eta \nabla f_i(x_{i,r}^t), \qquad \hat{x}_{i,r}^t = 2x_{i,r}^t - y_{i,r}^t. \tag{84}$$

Proof. We prove (84) by induction.

Base case: By (25) and (26), $x_{i,0}^0 = u = \operatorname{prox}_{\eta f_i}(y_{i,0}^0)$. The proximal optimality condition gives $y_{i,0}^0 = x_{i,0}^0 + \eta \nabla f_i(x_{i,0}^0)$. The second identity follows from the initialization of $\hat{x}_{i,0}^0$.

Inductive step: Suppose that both identities hold at round r. We consider two cases:

- **Case 1:** If $i \in S_r^t$, then $x_{i,r+1}^t = \operatorname{prox}_{\eta f_i}(y_{i,r+1}^t)$. Hence,

  $$y_{i,r+1}^t = x_{i,r+1}^t + \eta \nabla f_i(x_{i,r+1}^t),$$

  and the reflection update yields $\hat{x}_{i,r+1}^t = 2x_{i,r+1}^t - y_{i,r+1}^t$.
- **Case 2:** If $i \notin S_r^t$, all local states remain unchanged by (23), so the induction hypothesis applies.

Finally, the carryover rule (27) preserves both identities at $(t+1, 0)$. This completes the proof. □

Lemma 13. *Let $\{(x_r^t, x_{i,r}^t, y_{i,r}^t)\}$ be generated by the exact variant of Algorithm 1. For every $i \in S_r^t$, we have*

$$\|x_r^t - x_{i,r}^t\|^2 \leq \frac{(1+\eta L)^2}{\alpha^2} \|x_{i,r+1}^t - x_{i,r}^t\|^2. \tag{85}$$

Proof. By the update of $y_{i,r+1}^t$ and Lemma 12, we have

$$\begin{aligned} x_r^t - x_{i,r}^t &= \frac{1}{\alpha}(y_{i,r+1}^t - y_{i,r}^t) \\ &= \frac{1}{\alpha}(x_{i,r+1}^t - x_{i,r}^t) + \frac{\eta}{\alpha}\big[\nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t)\big]. \end{aligned} \tag{86}$$

Therefore, the triangle inequality and L-smoothness give

$$\begin{aligned} \|x_r^t - x_{i,r}^t\|^2 &\leq \frac{1}{\alpha^2}\big(\|x_{i,r+1}^t - x_{i,r}^t\| + \eta\|\nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t)\|\big)^2 \\ &\leq \frac{(1+\eta L)^2}{\alpha^2}\|x_{i,r+1}^t - x_{i,r}^t\|^2. \end{aligned}$$

□

Lemma 14. *Let $\{(x_r^t, x_{i,r}^t, \hat{x}_{i,r}^t)\}$ be generated by the exact variant of Algorithm 1. Then,*

$$\|\nabla f(x_r^t)\|^2 \leq \frac{(1+\eta L)^2}{n\eta^2} \sum_{i=1}^n \|x_{i,r}^t - x_r^t\|^2. \tag{87}$$

Proof. From the aggregation step and Lemma 12,

$$\begin{aligned} x_r^t &= \frac{1}{n}\sum_{i=1}^n \hat{x}_{i,r}^t = \frac{1}{n}\sum_{i=1}^n (2x_{i,r}^t - y_{i,r}^t) \\ &= \frac{1}{n}\sum_{i=1}^n \big[x_{i,r}^t - \eta \nabla f_i(x_{i,r}^t)\big]. \end{aligned} \tag{88}$$

Rearranging this averaged equality, we obtain

$$\begin{aligned} \nabla f(x_r^t) = \frac{1}{n}\sum_{i=1}^n \big[\nabla f_i(x_r^t) - \nabla f_i(x_{i,r}^t)\big] \\ + \frac{1}{n\eta}\sum_{i=1}^n (x_{i,r}^t - x_r^t). \end{aligned} \tag{89}$$

By the triangle inequality, L-smoothness, and Jensen's inequality,

$$\begin{aligned} \|\nabla f(x_r^t)\|^2 &\le \frac{1}{\eta^2}\left(\frac{1}{n}\sum_{i=1}^n \big[\|x_{i,r}^t - x_r^t\| + \eta\|\nabla f_i(x_r^t) - \nabla f_i(x_{i,r}^t)\|\big]\right)^2 \\ &\le \frac{(1+\eta L)^2}{n\eta^2}\sum_{i=1}^n \|x_{i,r}^t - x_r^t\|^2. \end{aligned} \tag{90}$$

□

Lemma 15. *Let $\mathcal{D}_r^t$ be the potential function defined in Definition 4. Then, for every round of the exact variant,*

$$\begin{aligned} \mathcal{D}_{r+1}^t \le \mathcal{D}_r^t - \frac{2-\alpha-\alpha\eta L - 2\eta^2 L^2}{2n\alpha\eta}\sum_{i\in S_r^t}\|x_{i,r+1}^t - x_{i,r}^t\|^2 \\ - \frac{1}{2\eta}\|x_{r+1}^t - x_r^t\|^2. \end{aligned} \tag{91}$$

Proof. We first consider the change of the server variable. For exact proximal updates, Lemma 12 and the aggregation identity give

$$\begin{aligned} \mathcal{D}_{r+1}^t &= \frac{1}{n}\sum_{i=1}^n \left[f_i(x_{i,r+1}^t) + \langle \nabla f_i(x_{i,r+1}^t), x_r^t - x_{i,r+1}^t\rangle + \frac{1}{2\eta}\|x_r^t - x_{i,r+1}^t\|^2\right] \\ &\quad + \frac{1}{n\eta}\sum_{i=1}^n \langle x_r^t - \hat{x}_{i,r+1}^t, x_{r+1}^t - x_r^t\rangle + \frac{1}{2\eta}\|x_{r+1}^t - x_r^t\|^2 \\ &= \frac{1}{n}\sum_{i=1}^n \left[f_i(x_{i,r+1}^t) + \langle \nabla f_i(x_{i,r+1}^t), x_r^t - x_{i,r+1}^t\rangle + \frac{1}{2\eta}\|x_r^t - x_{i,r+1}^t\|^2\right] \\ &\quad - \frac{1}{2\eta}\|x_{r+1}^t - x_r^t\|^2, \end{aligned} \tag{92}$$

where the last equality uses $x_{r+1}^t = n^{-1}\sum_{i=1}^n \hat{x}_{i,r+1}^t$.

Next, let $i \in S_r^t$. By L-smoothness,

$$\begin{aligned} f_i(x_{i,r+1}^t) - f_i(x_{i,r}^t) \le \langle \nabla f_i(x_{i,r+1}^t), x_{i,r+1}^t - x_{i,r}^t\rangle \\ + \frac{L}{2}\|x_{i,r+1}^t - x_{i,r}^t\|^2. \end{aligned} \tag{93}$$

Expanding the squared distances and applying (93), we obtain

$$\begin{aligned} & f_i(x_{i,r+1}^t) + \langle \nabla f_i(x_{i,r+1}^t), x_r^t - x_{i,r+1}^t \rangle + \frac{1}{2\eta}\|x_r^t - x_{i,r+1}^t\|^2 \\ & \quad - f_i(x_{i,r}^t) - \langle \nabla f_i(x_{i,r}^t), x_r^t - x_{i,r}^t \rangle - \frac{1}{2\eta}\|x_r^t - x_{i,r}^t\|^2 \\ & \le \frac{1+\eta L}{2\eta}\|x_{i,r+1}^t - x_{i,r}^t\|^2 \\ & \quad + \left\langle \nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t) - \frac{1}{\eta}(x_{i,r+1}^t - x_{i,r}^t),\, x_r^t - x_{i,r}^t \right\rangle. \end{aligned} \tag{94}$$

Substituting (86) into the last inner product, the mixed terms cancel:

$$\begin{aligned} & \left\langle \nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t) - \frac{1}{\eta}(x_{i,r+1}^t - x_{i,r}^t),\, x_r^t - x_{i,r}^t \right\rangle \\ & = \frac{\eta}{\alpha}\|\nabla f_i(x_{i,r+1}^t) - \nabla f_i(x_{i,r}^t)\|^2 - \frac{1}{\alpha\eta}\|x_{i,r+1}^t - x_{i,r}^t\|^2 \\ & \le \left(\frac{\eta L^2}{\alpha} - \frac{1}{\alpha\eta}\right)\|x_{i,r+1}^t - x_{i,r}^t\|^2. \end{aligned} \tag{95}$$

Consequently, (94) gives

$$\begin{aligned} & f_i(x_{i,r+1}^t) + \langle \nabla f_i(x_{i,r+1}^t), x_r^t - x_{i,r+1}^t \rangle + \frac{1}{2\eta}\|x_r^t - x_{i,r+1}^t\|^2 \\ & \quad - f_i(x_{i,r}^t) - \langle \nabla f_i(x_{i,r}^t), x_r^t - x_{i,r}^t \rangle - \frac{1}{2\eta}\|x_r^t - x_{i,r}^t\|^2 \\ & \le -\frac{2-\alpha-\alpha\eta L - 2\eta^2 L^2}{2\alpha\eta}\|x_{i,r+1}^t - x_{i,r}^t\|^2. \end{aligned} \tag{96}$$

For $i \notin S_r^t$, the local states remain unchanged. Summing (96) over the active clients and using (92) proves (91). □

Lemma 16. *Let $\mathcal{D}_r^t$ be defined as in Definition 4, and let τ_i^t be the unique participation round of client i in meta-epoch t, as defined in Appendix B. Then,*

$$\begin{aligned} & \frac{2-\alpha-\alpha\eta L - 2\eta^2 L^2}{2n\alpha\eta}\sum_{i=1}^n \|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2 \\ & \le \mathcal{D}_0^t - \mathcal{D}_R^t. \end{aligned} \tag{97}$$

Proof. Summing Lemma 15 over $r = 0, \dots, R-1$ gives

$$\begin{aligned} \mathcal{D}_R^t - \mathcal{D}_0^t \le & -\frac{2-\alpha-\alpha\eta L - 2\eta^2 L^2}{2n\alpha\eta}\sum_{r=0}^{R-1}\sum_{i\in S_r^t}\|x_{i,r+1}^t - x_{i,r}^t\|^2 \\ & - \frac{1}{2\eta}\sum_{r=0}^{R-1}\|x_{r+1}^t - x_r^t\|^2. \end{aligned} \tag{98}$$

Each client participates exactly once, so

$$\sum_{r=0}^{R-1}\sum_{i\in S_r^t}\|x_{i,r+1}^t - x_{i,r}^t\|^2 = \sum_{i=1}^n \|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2.$$

Dropping the nonnegative squared server increments proves (97). The inequality holds for each realized client permutation. □

Lemma 17. *Let $\mathcal{D}_r^t$ be defined as in Definition 4. For every complete client traversal, we have*

$$\begin{aligned} & \frac{\alpha\eta(2-\alpha-\alpha\eta L - 2\eta^2 L^2)}{2(1+\eta L)^4(1+\alpha)^2}\frac{1}{R}\sum_{r=0}^{R-1}\|\nabla f(x_r^t)\|^2 \\ & \le \mathcal{D}_0^t - \mathcal{D}_R^t. \end{aligned} \tag{99}$$

Proof. We first bound the change in the server states. By Lemma 12, the reflection increment of the client participating at round τ_i^t satisfies

$$\begin{aligned} g_{i,\tau_i^t}^t &= \hat{x}_{i,\tau_i^t+1}^t - \hat{x}_{i,\tau_i^t}^t \\ &= (x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t) \\ &\quad - \eta\big[\nabla f_i(x_{i,\tau_i^t+1}^t) - \nabla f_i(x_{i,\tau_i^t}^t)\big]. \end{aligned} \tag{100}$$

Hence, by L-smoothness,

$$\|g_{i,\tau_i^t}^t\| \le (1+\eta L)\|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|.$$

By the server update in (29), the difference between any two server states is a sum, up to sign, of at most one such increment per client. Cauchy–Schwarz yields

$$\begin{aligned} \|x_r^t - x_q^t\| &\le \left(\frac{1}{n}\sum_{i=1}^n \|g_{i,\tau_i^t}^t\|^2\right)^{1/2} \\ &\le (1+\eta L)\left(\frac{1}{n}\sum_{i=1}^n \|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2\right)^{1/2}, \end{aligned} \tag{101}$$

for all $0 \le r, q \le R$.

Next, we compare a client's state at round r with the server state at its participation round. By (86), we consider two cases:

- **Case 1:** If $r \le \tau_i^t$, then $x_{i,r}^t = x_{i,\tau_i^t}^t$ and

$$x_{\tau_i^t}^t - x_{i,r}^t = \frac{1}{\alpha}\Big[x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t + \eta\big(\nabla f_i(x_{i,\tau_i^t+1}^t) - \nabla f_i(x_{i,\tau_i^t}^t)\big)\Big].$$

- **Case 2:** If $r > \tau_i^t$, then $x_{i,r}^t = x_{i,\tau_i^t+1}^t$ and

$$\begin{aligned} x_{\tau_i^t}^t - x_{i,r}^t = \frac{1-\alpha}{\alpha}(x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t) \\ + \frac{\eta}{\alpha}\big[\nabla f_i(x_{i,\tau_i^t+1}^t) - \nabla f_i(x_{i,\tau_i^t}^t)\big]. \end{aligned}$$

Since $|1-\alpha| \le 1$, both cases imply

$$\|x_{\tau_i^t}^t - x_{i,r}^t\| \le \frac{1+\eta L}{\alpha}\|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|. \tag{102}$$

Combining (101) and (102), and applying the triangle inequality, gives

$$\begin{aligned} &\left(\frac{1}{n}\sum_{i=1}^n \|x_r^t - x_{i,r}^t\|^2\right)^{1/2} \\ &\le (1+\eta L)(1+\alpha^{-1})\left(\frac{1}{n}\sum_{i=1}^n \|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2\right)^{1/2}. \end{aligned} \tag{103}$$

Using Lemma 14, we obtain

$$\frac{1}{R}\sum_{r=0}^{R-1}\|\nabla f(x_r^t)\|^2 \le \frac{(1+\eta L)^4(1+\alpha^{-1})^2}{n\eta^2}\sum_{i=1}^n \|x_{i,\tau_i^t+1}^t - x_{i,\tau_i^t}^t\|^2. \tag{104}$$

Finally, applying Lemma 16 to (104) proves (99). All these inequalities hold pathwise, without conditioning on a partially completed reshuffling. □

Proof of Theorem 3. First, we define the constant $\mathcal{A}$ as

$$\mathcal{A} := \frac{2(1+\eta L)^4(1+\alpha)^2}{\alpha\eta(2-\alpha-\alpha\eta L-2\eta^2L^2)}. \tag{105}$$

It is positive by (83). By Lemma 17 and Definition 5, we have

$$\mathcal{G}^t \le \mathcal{A}(\mathcal{D}_0^t - \mathcal{D}_R^t). \tag{106}$$

Summing (106) over $t = 0, \dots, T-1$ and using $\mathcal{D}_R^t = \mathcal{D}_0^{t+1}$, we obtain

$$\sum_{t=0}^{T-1} \mathcal{G}^t \le \mathcal{A}(\mathcal{D}_0^0 - \mathcal{D}_0^T). \tag{107}$$

By the initialization (25), $x_{i,0}^0 = u = x^0$ and $x_0^0 = u - \eta\nabla f(u)$. Therefore,

$$\begin{aligned} \mathcal{D}_0^0 &= f(u) + \langle \nabla f(u), -\eta\nabla f(u)\rangle + \frac{\eta}{2}\|\nabla f(u)\|^2 \\ &= f(x^0) - \frac{\eta}{2}\|\nabla f(x^0)\|^2 \le f(x^0). \end{aligned} \tag{108}$$

Moreover, the L-smoothness of f_i implies

$$\begin{aligned} \mathcal{D}_r^t &= \frac{1}{n}\sum_{i=1}^n \left[f_i(x_{i,r}^t) + \langle \nabla f_i(x_{i,r}^t), x_r^t - x_{i,r}^t\rangle + \frac{1}{2\eta}\|x_r^t - x_{i,r}^t\|^2 \right] \\ &\ge \frac{1}{n}\sum_{i=1}^n \left[f_i(x_r^t) + \left(\frac{1}{2\eta} - \frac{L}{2}\right)\|x_r^t - x_{i,r}^t\|^2 \right] \\ &\ge f(x_r^t) \ge f^*, \end{aligned} \tag{109}$$

where the last step uses $\eta < 1/L$. Combining (107), (108), and (109) gives

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\mathcal{G}^t] \le \frac{\mathcal{A}}{T}[f(x^0) - f^*]. \tag{110}$$

Finally, let $\tilde{x}$ be selected independently and uniformly from $\{x_r^t : 0 \le t \le T-1,\ 0 \le r \le R-1\}$. Then

$$\begin{aligned} \mathbb{E}\|\nabla f(\tilde{x})\|^2 &= \frac{1}{TR}\sum_{t=0}^{T-1}\sum_{r=0}^{R-1} \mathbb{E}\|\nabla f(x_r^t)\|^2 \\ &= \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\mathcal{G}^t] \\ &\le \frac{\mathcal{A}}{T}[f(x^0) - f^*]. \end{aligned} \tag{111}$$

Thus, for any $\epsilon > 0$, choosing

$$T \ge \max\left\{1, \left\lceil \frac{\mathcal{A}[f(x^0) - f^*]}{\epsilon} \right\rceil\right\}$$

proves the stated $O(\epsilon^{-1})$ meta-epoch complexity. □

Proof of Corollary 4. Choosing $\alpha = 1$ and $\eta = 1/(4L)$, we have $\eta_1 = 1/(2L) > \eta$ and

$$\mathcal{A} = \frac{2(5/4)^4(1+1)^2}{(1/(4L))(2-1-1/4-2/16)} = 125L.$$

Substituting this value into (111) proves (11) and Corollary 4. □

The proofs in Appendices B and C apply to every complete client traversal. They establish convergence under client reshuffling but do not compare random permutations with deterministic client orders. As in *Algorithm 1*, every scheduled client must return its update; skipped updates and unbounded delays require a separate analysis.

D Client Coverage and Participation Imbalance

Under partial participation, the distinction between independent client sampling and client reshuffling concerns how client selections are coupled across communication rounds. In standard independent sampling, each round draws a uniform cohort of C distinct clients from the n clients, but successive cohorts are drawn independently. Thus, a client is selected with probability $\gamma = C/n$ in each round and may be selected repeatedly before other clients have participated. Client reshuffling instead partitions a random permutation of all clients into $R = n/C$ disjoint cohorts, ensuring that every client participates exactly once per meta-epoch.

Client coverage under independent sampling. Consider B communication rounds of independent uniform cohort sampling. For any fixed client i, its number of participations satisfies

$$X_i = \sum_{k=1}^{B} \mathbf{1}\{i \in S_k\} \sim \mathrm{Binomial}(B, \gamma), \qquad \mathbb{E}[X_i] = B\gamma, \quad \mathrm{Var}(X_i) = B\gamma(1-\gamma).$$

The probability that client i is never selected is

$$\Pr(X_i = 0) = (1-\gamma)^B.$$

Consequently, the expected number and fraction of uncovered clients are

$$\mathbb{E}[U_B] = n(1-\gamma)^B, \qquad \zeta_B := \frac{\mathbb{E}[U_B]}{n} = (1-\gamma)^B.$$

These identities are exact for cohorts sampled uniformly without replacement *within* each round and independently *across* rounds. Table 5 illustrates the resulting finite-horizon coverage. At 5% participation, for example, 20 independent rounds leave an expected 35.85% of clients unvisited; by contrast, reshuffling visits every client within $R = 20$ rounds.

Table 5: Expected fraction of clients not selected after B independent communication rounds, $\zeta_B = (1-\gamma)^B$. Entries below 0.001% are displayed as $< 0.001\%$.

Rounds B	$\gamma = 5\%$	$\gamma = 10\%$	$\gamma = 50\%$
20	35.85%	12.16%	$< 0.001\%$
50	7.69%	0.52%	$< 0.001\%$
200	0.0035%	$< 0.001\%$	$< 0.001\%$

Participation imbalance. Even when every client has participated at least once, independent sampling can produce unequal participation counts over a finite horizon. For a fixed client, the coefficient of variation of X_i is

$$\mathrm{CV}(X_i) = \frac{\sqrt{\mathrm{Var}(X_i)}}{\mathbb{E}[X_i]} = \sqrt{\frac{1-\gamma}{B\gamma}}.$$

For any $\delta \in (0,1)$, the multiplicative Chernoff bound gives

$$\Pr(|X_i - B\gamma| \geq \delta B\gamma) \leq 2\exp\left(-\frac{\delta^2 B\gamma}{3}\right). \tag{112}$$

Therefore, a sufficient condition for this per-client deviation probability to be at most α_{tail} is $B\gamma \geq 3\delta^{-2}\log(2/\alpha_{\mathrm{tail}})$. Table 6 reports these sufficient thresholds. Because the Chernoff bound can be conservative, failing to meet a threshold does not establish that the corresponding deviation must occur. Moreover, simultaneous control of all n clients would require an additional multiple-client argument.

Implications for client reshuffling. Uniform independent sampling is not inherently biased in a single round: with appropriate aggregation, its sampled client gradient is an unbiased estimate of the full-client gradient. Its finite-horizon coverage and participation counts are nevertheless random, and persistent client states may remain unrefreshed for many rounds. Reshuffling removes this coverage uncertainty within each meta-epoch, although its without-replacement cohorts are dependent across

Table 6: Sufficient lower bounds on $B\gamma$ from (112) for one client's participation count to lie within a relative tolerance of its mean with the indicated probability.

Relative tolerance	95%	90%	75%	50%
10%	1106.7	898.7	623.8	415.9
20%	276.7	224.7	156.0	104.0
30%	123.0	99.9	69.3	46.2
50%	44.3	35.9	25.0	16.6

rounds and cannot be analyzed as independent samples. The convergence analysis in Appendices B and C explicitly handles that difficulty.

The observed communication benefit is task-dependent rather than a direct consequence of the coverage formula. For example, in a matched-$E = 5$ CIFAR-10 experiment with $\rho = 1.0$, FedCDR reached 75% test accuracy after 103, 70, and 57 synchronizations at 5%, 10%, and 50% participation, respectively. Decreasing participation from 50% to 5% increased the observed round count by approximately 1.81 times, while the factor n/C increased tenfold.

E Communication, Computation, and Memory Overhead

E.1 Communication and Persistent Client State

Let d be the number of trainable parameters and assume 32-bit floating-point transmission, so one d-dimensional vector occupies $4d$ bytes. For each selected client, FedAvg, FedProx, FedDR, and FedCDR transmit one model-sized vector in each direction per communication round, corresponding to $8d$ bytes in total. SCAFFOLD and the FedDC implementation considered here additionally exchange correction information, for a total of $16d$ bytes per selected client and round. Client reshuffling changes the participation schedule, not these per-participation message sizes. Table 7 also compares the persistent client-specific state retained between participations.

Table 7: Per-participation communication and persistent client-specific state. Communication is in bytes for d trainable parameters stored as 32-bit floats. Persistent state counts exclude the shared global model, optimizer buffers, and temporary working memory.

Method	Download	Upload	Total	Persistent state
FedAvg	$4d$	$4d$	$8d$	none
FedProx	$4d$	$4d$	$8d$	none
SCAFFOLD	$8d$	$8d$	$16d$	one d-vector
FedDC	$8d$	$8d$	$16d$	two d-vectors
FedDR	$4d$	$4d$	$8d$	two d-vectors
FedCDR	$4d$	$4d$	$8d$	two d-vectors

In particular, FedCDR retains the two independent Douglas–Rachford states $(x_i, y_i) \in \mathbb{R}^{2d}$ for each client. The reflected state $\hat{x}_i = 2x_i - y_i$ can be recomputed from these states, and the transmitted increment g_i is temporary. Consequently, retaining the states of all n clients requires $2nd$ floating-point entries, or $8nd$ bytes in this precision, irrespective of where the states are physically stored. This additional storage is a cost of FedCDR relative to stateless FedAvg and FedProx, and it matches the client-specific state count of FedDR.

E.2 Local Computation

For a client with m_i training examples and mini-batch size b, E local epochs correspond to approximately $E\lceil m_i/b \rceil$ mini-batch gradient evaluations per participation. The main empirical comparisons use the same local-epoch budget $E = 5$ for every method; FedCDR uses these steps to solve its proximal subproblem approximately. The proximal quadratic and the correction terms in other methods add vector operations, so matching local epochs does not imply identical measured runtime. Conversely, assigning more local epochs to one method cannot be treated as a matched-computation comparison.

E.3 Time to Target under Matched Local Computation

Communication rounds alone do not determine wall-clock time. In a synchronous cross-device implementation, the elapsed time to a target includes server processing, the slowest participating client's local computation, and network transfer. Schematically, with K_{target} rounds this is

$$\tau_{\text{target}}^{\text{deploy}} = \sum_{k=1}^{K_{\text{target}}} \left(\tau_{\text{server},k} + \max_{i \in S_k} \tau_{\text{local},i,k} + \tau_{\text{down},k} + \tau_{\text{up},k} \right).$$

Our experiments run selected clients sequentially on one GPU. For comparison, we reconstruct the compute-only critical path of each round as

$$\tau_{\text{comp},k}^{\text{parallel}} = \tau_{\text{server},k} + \max_{i \in S_k} \tau_{\text{local},i,k}.$$

This reconstruction excludes network latency and is not a measurement of distributed deployment time. The separately reported simulator wall-clock includes sequential local execution.

As Table 8 shows, FedCDR reaches the target in fewer rounds than the listed baselines in this setting. Its reconstructed per-round compute critical path is close to FedDR's (1.143 versus 1.139 seconds),

Table 8: Time to 75% top-1 test accuracy on CIFAR-10 ($\rho = 1.0$) under matched $E = 5$. Results are from one run per method with the same seed, cohort size, batch size, and optimizer schedule. The compute critical path is reconstructed from single-GPU timings; simulator wall-clock is measured directly.

Method	Rounds	Critical path per round (s)	Bytes/client per round	Compute to target (s)	Simulator time (min)
FedAvg	347	0.953	$8d$	331	43.0
FedProx	356	1.171	$8d$	417	64.7
SCAFFOLD	134	1.263	$16d$	169	20.7
FedDC	89	1.193	$16d$	106	16.8
FedDR	123	1.139	$8d$	140	19.4
FedCDR	**70**	1.143	$8d$	**80**	**11.0**

while its lower round count reduces the cumulative compute critical path from 140 to 80 seconds. FedCDR also has the lowest measured simulator time among these methods. These single-run, single-GPU results demonstrate the trade-off for the tested configuration.

F Communication–Computation Trade-off and System Heterogeneity

F.1 Local Computation and Proximal Approximation Accuracy

FedCDR approximates each local proximal update using a finite number of mini-batch SGD steps. The number of local epochs E therefore controls the computation spent on each proximal subproblem. As shown in Table 4, increasing E from 5 to 20 reduces the rounds needed to reach 98% test accuracy on MNIST from 112 to 43 for $\rho = 0.1$ and from 89 to 32 for $\rho = 1.0$. The reduction in communication rounds comes with greater local work per client participation.

Theorem 1 quantifies the effect of inexact proximal updates through their mean-square approximation errors. Under the accumulated error condition (8), its convergence bound includes a term proportional to the error budget M. More accurate local solutions can reduce this contribution, at the cost of additional computation. However, a fixed number of local epochs does not automatically satisfy (8): the condition covers all stored client states, including inactive and carried-over states, and requires an accumulated bound M independent of the training horizon.

To examine proximal accuracy directly, we sampled proximal subproblems and their corresponding warm starts from training on MNIST with $\rho = 0.1$. For each sampled problem, we computed a high-accuracy reference solution $x_{i,t}^{\mathrm{ref}}$ and reran the local solver for $E \in \{3, 5, 10, 20\}$ epochs from the same starting point $x_{i,t}^{(0)}$. In addition to the squared error $\|x_{i,t}^{(E)} - x_{i,t}^{\mathrm{ref}}\|^2$, we measured the relative squared error

$$\mathrm{RelErr}_{i,t}(E) = \frac{\|x_{i,t}^{(E)} - x_{i,t}^{\mathrm{ref}}\|^2}{\|x_{i,t}^{(0)} - x_{i,t}^{\mathrm{ref}}\|^2}. \tag{113}$$

Table 9 reports the diagnostic averages over the sampled problems, clients, and rounds, using three diagnostic seeds. The communication-round counts are taken from the corresponding training runs in Table 4.

Table 9: Proximal approximation diagnostics for FedCDR on MNIST with $\rho = 0.1$. Errors are averaged over sampled proximal subproblems and three diagnostic seeds. The rounds to 98% test accuracy come from the training experiments and are shown for comparison.

E	Mean SGD steps	Mean squared error	Mean relative error	Rounds to 98% accuracy
3	18	10.30	0.32	161
5	30	7.36	0.18	112
10	60	3.49	0.12	60
20	120	2.21	0.06	43

In this diagnostic, increasing E from 3 to 20 reduces the mean squared proximal error from 10.30 to 2.21 and the mean relative error from 0.32 to 0.06, alongside the decrease in communication rounds from 161 to 43. These measurements are consistent with the interpretation that more local work improves the approximate proximal solutions in the tested setting.

F.2 Partial Local Updates under System Heterogeneity

We also examine heterogeneous local computation using a simulated global-clock protocol following Li et al. [2020a]. At each round, a fraction $p \in \{0\%, 50\%, 90\%\}$ of the selected clients is designated as potentially compute-limited. Each such client is assigned a local epoch budget drawn uniformly from $\{1, \ldots, E\}$, while the other selected clients run E epochs. Thus, $p = 0\%$ is the homogeneous-computation case, and higher values of p increase the prevalence of reduced local budgets. In this simulation, the CR+baseline implementations discard unfinished partial updates at the clock deadline, whereas FedCDR incorporates the partial proximal updates returned by participating clients.

The nominal local budgets also differ by method in the reported configuration: $E = 5$ for CR+FedAvg, CR+SCAFFOLD, and CR+FedDC, and $E = 10$ for FedCDR, as specified in Appendix G. Accordingly, the results below characterize each method's response to the simulated straggler fraction.

Table 10: Communication rounds to reach 98% top-1 test accuracy on MNIST under simulated straggler fractions p. > 400 indicates that the target was not reached within 400 rounds. Nominal local budgets are $E = 5$ for CR+baselines and $E = 10$ for FedCDR.

Methods	$\rho = 0.1$			$\rho = 1.0$		
	0%	50%	90%	0%	50%	90%
CR+FedAvg	386	> 400	> 400	319	> 400	> 400
CR+SCAFFOLD	85	248	> 400	75	136	> 400
CR+FedDC	78	212	> 400	57	94	379
FedCDR	60	82	116	49	76	104

As shown in Table 10, the required rounds increase with the simulated straggler fraction for every method. FedCDR reaches the target within 116 rounds at $\rho = 0.1$ and 104 rounds at $\rho = 1.0$ even when $p = 90\%$; several CR+baseline configurations do not reach the target within the 400-round budget. The observed differences are consistent with the benefits of retaining partial updates under the tested protocol, but they also reflect the methods' different local budgets and implementation choices.

G Experiment Details

G.1 Datasets and Data Partitioning

G.1.1 Synthetic Datasets

We generate three synthetic classification datasets following [Li et al., 2020a]. Each dataset contains 500 clients and a 10-class problem with feature dimension 60. For client k, the label is generated by

$$y = \arg\max_{c \in \{1,\dots,10\}} [W_k x + b_k]_c, \tag{114}$$

where $W_k \in \mathbb{R}^{10\times 60}$ and $b_k \in \mathbb{R}^{10}$ are client-specific parameters. Their deviations from shared reference parameters (W, b) are sampled according to

$$\begin{aligned} W_k - W \sim \mathcal{N}(u_k, 1), \quad b_k - b \sim \mathcal{N}(u_k, 1), \\ u_k \sim \mathcal{N}(0, \beta_1). \end{aligned} \tag{115}$$

The parameter β_1 controls the variation in the client-specific model parameters.

Client features have a shared diagonal covariance matrix $\Sigma \in \mathbb{R}^{60\times 60}$, with $\Sigma_{j,j} = j^{-1.2}$, and client-dependent means:

$$\begin{aligned} x \mid k \sim \mathcal{N}(v_k, \Sigma), \\ v_k \sim \mathcal{N}(B_k, 1), \quad B_k \sim \mathcal{N}(0, \beta_2). \end{aligned} \tag{116}$$

Here β_2 controls the variation in feature distributions across clients. The Gaussian draws for vector- and matrix-valued quantities are understood componentwise. Since the conditional Gaussian distributions retain unit variance, $\beta_1 = \beta_2 = 0$ removes the higher-level mean shifts but does not force all clients to have identical local distributions.

We consider Synthetic-$(0, 0)$, Synthetic-$(1, 1)$, and Synthetic-$(5, 5)$, where the two coordinates specify (β_1, β_2) and larger values yield greater heterogeneity under this construction. Client sample counts are generated from a log-normal distribution,

$$n_k \sim \text{LogNormal}(\mu = 2, \sigma = 2) + 10, \tag{117}$$

and are capped at 50. Here μ and σ are the mean and standard deviation of the underlying normal distribution.

G.1.2 Benchmark Datasets

We use MNIST, CIFAR-10, and CIFAR-100. MNIST contains $60{,}000$ training and $10{,}000$ test grayscale images of size 28×28 in 10 classes. CIFAR-10 and CIFAR-100 [Krizhevsky, 2009] each contain $50{,}000$ training and $10{,}000$ test color images of size 32×32, with 10 and 100 classes, respectively.

For the main experiments, each benchmark dataset is distributed across $J = 500$ clients using a class-wise Dirichlet partitioning scheme [Wang et al., 2020]. For each class $k \in \{1, \dots, K\}$, we sample the proportions of its training examples assigned to the J clients according to

$$\begin{aligned} (p_{k,1}, \dots, p_{k,J}) \sim \text{Dirichlet}_J(\rho, \dots, \rho), \\ \sum_{j=1}^{J} p_{k,j} = 1. \end{aligned} \tag{118}$$

Here, $p_{k,j}$ denotes the proportion of training examples from class k assigned to client j. We have $K = 10$ for MNIST and CIFAR-10 and $K = 100$ for CIFAR-100. We consider $\rho \in \{0.1, 1.0\}$, where smaller values of ρ result in more heterogeneous client label distributions. The original test set is retained as a shared global test set for all methods.

To introduce client-size imbalance, client sample counts are drawn from

$$N_j \sim \text{LogNormal}(\mu = 4, \sigma = 2) + 30 \tag{119}$$

and capped at 500. The class proportions and client sample counts jointly specify the heterogeneous training partitions; μ and σ again describe the underlying normal distribution.

G.2 Models, Training Protocol, and Hyperparameters

G.2.1 Model Architectures

For the synthetic datasets, we use a fully connected network with widths $60\times32\times10$. For MNIST, we use a $784\times128\times10$ fully connected network [Li et al., 2020b]. For CIFAR-10 and CIFAR-100, we use ResNet-20 [He et al., 2016], replacing batch normalization with group normalization (GN) [Wu and He, 2018]. Unlike batch normalization, GN does not compute normalization statistics across the mini-batch. Table 11 summarizes the architecture.

Table 11: ResNet-20 with group normalization for CIFAR-10 and CIFAR-100. Each residual block contains two 3×3 convolutions and a skip connection. The first block of Stages 2 and 3 downsamples spatially.

Stage	**Output size**	**Layers**
Stem	32×32	3×3 conv, 16 channels
Stage 1	32×32	3 residual blocks, 16 channels, GN($G = 4$)
Stage 2	16×16	3 residual blocks, 32 channels, GN($G = 8$)
Stage 3	8×8	3 residual blocks, 64 channels, GN($G = 8$)
Output	1×1	Global average pooling, FC-10/FC-100

G.2.2 Client Participation and Local Epochs

Unless otherwise specified, the experiments use 500 clients, with 10% participating per communication round. In the main accuracy comparisons (Tables 1 and 2) and the participation-rate experiments (Table 3), all methods use $E = 5$ local epochs per participating client, including the approximate proximal solvers in FedProx and FedCDR. These matched-budget experiments use the same batch sizes, optimizer schedules, and hyperparameter-tuning budgets across methods. Final accuracies in the main comparison are measured after 400 communication rounds and reported as mean $\pm$ standard deviation over five independent runs.

Other experiments vary local computation explicitly. Table 4 compares FedCDR at $E \in \{3, 5, 10, 20\}$, while the straggler simulation in Table 10 uses nominal $E = 5$ for the CR+baselines and $E = 10$ for FedCDR.

G.2.3 Optimization and Hyperparameter Selection

We use mini-batch stochastic gradient descent (SGD) for local optimization, including inexact proximal subproblem solutions. The local batch size is 16 on the synthetic datasets and 32 on MNIST, CIFAR-10, and CIFAR-100. SGD uses momentum 0.9. For weight decay, the objective includes the penalty $\frac{\lambda_{\mathrm{wd}}}{2}\|w\|_2^2$ with $\lambda_{\mathrm{wd}} = 5 \times 10^{-4}$.

Hyperparameters are selected by grid search based on validation performance. The initial local learning rate ℓ_0 is chosen from $\{0.01, 0.02, 0.05, 0.1\}$. For communication round $k = 0, \ldots, K_{\mathrm{sync}} - 1$, where K_{sync} is the total number of scheduled communication rounds, the local learning rate follows

$$\ell_k = \begin{cases} \ell_0, & 0 \le k < \frac{K_{\mathrm{sync}}}{2}, \\ 0.1\ell_0, & \frac{K_{\mathrm{sync}}}{2} \le k < \frac{3K_{\mathrm{sync}}}{4}, \\ 0.01\ell_0, & \frac{3K_{\mathrm{sync}}}{4} \le k < K_{\mathrm{sync}}. \end{cases} \tag{120}$$

We use the communication-round index here to distinguish this schedule from the meta-epoch index t in the convergence analysis.

For FedProx, we search the proximal coefficient $\mu_{\mathrm{prox}} \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. For FedDC, the drift-regularization weight α_{DC} is selected from $\{0.03, 0.1, 0.3\}$ on the synthetic datasets and MNIST, and from $\{0.003, 0.01, 0.03\}$ on CIFAR-10 and CIFAR-100. For FedCDR, we search the proximal coefficient $\eta \in \{10^{-3}, 10^{-2}, 10^{-1}, 1\}$ and relaxation parameter $\alpha \in \{0.5, 1.0, 1.5, 1.99\}$.

G.2.4 Preprocessing

For CIFAR-10 and CIFAR-100, training images undergo random cropping and horizontal flipping. We use dataset-specific channel-wise normalization. For CIFAR-10, the means are $(0.4914, 0.4824, 0.4467)$, and the standard deviations are $(0.2471, 0.2435, 0.2616)$. For CIFAR-100, the means are $(0.5071, 0.4867, 0.4408)$, and the standard deviations are $(0.2675, 0.2565, 0.2761)$.

G.2.5 Runtime Environment

Experiments are run on Ubuntu 20.04 using a compute node with a 24-core 2.80 GHz Intel processor, four NVIDIA L40S GPUs, and 256 GB of RAM. In the single-GPU simulator used for the time-to-target analysis in Appendix E, participating clients are processed sequentially. Simulator execution time therefore does not directly represent the wall-clock latency of a synchronous deployment with parallel client computation and network communication.

H Scalability and Robustness

H.1 Scaling the Number of Clients

We evaluate FedCDR with $10{,}000$ clients and a 5% participation rate (500 clients per communication round). The experiments use MNIST, CIFAR-10, and CIFAR-100 with $\rho \in \{0.1, 1.0\}$, and retain the corresponding model architectures. Learning rates and regularization parameters are tuned separately for each method. Table 12 reports final test accuracy after 400 communication rounds.

Table 12: Top-1 test accuracy (%) on benchmark datasets with $10{,}000$ clients and 5% participation after 400 communication rounds. Entries are mean $\pm$ standard deviation.

Method	$\rho = 0.1$			$\rho = 1.0$		
	MNIST	CIFAR-10	CIFAR-100	MNIST	CIFAR-10	CIFAR-100
FedAvg	96.06±0.20	63.70±0.97	52.03±1.12	97.02±0.18	75.67±0.78	62.97±1.00
FedProx	96.38±0.17	65.60±0.93	54.73±0.98	97.13±0.13	76.30±0.68	63.78±0.94
SCAFFOLD	96.90±0.16	70.34±0.77	62.67±0.80	97.99±0.12	80.05±0.63	64.89±0.73
FedDC	97.08±0.12	71.54±0.64	62.80±0.76	97.89±0.09	81.92±0.54	65.35±0.68
FedDR	97.00±0.16	70.93±0.70	62.34±0.85	97.56±0.10	81.30±0.61	65.00±0.78
CR+FedAvg	97.37±0.16	65.25±0.75	54.68±0.86	97.48±0.12	76.21±0.46	63.24±0.82
CR+FedProx	97.46±0.12	66.37±0.73	56.77±0.68	97.57±0.09	77.87±0.42	64.07±0.74
CR+SCAFFOLD	97.36±0.09	72.98±0.60	63.27±0.51	98.00±0.08	81.99±0.38	68.37±0.61
CR+FedDC	97.57±0.08	73.16±0.46	63.70±0.45	98.03±0.06	82.46±0.24	68.88±0.50
FedCDR	**98.00**±0.07	**75.02**±0.43	**65.99**±0.48	**98.32**±0.06	**84.78**±0.32	**70.98**±0.46

FedCDR achieves the highest mean test accuracy in all six settings. At $\rho = 0.1$, it exceeds the strongest baseline, CR+FedDC, by 1.86 percentage points on CIFAR-10 and 2.29 points on CIFAR-100. Across the four baseline pairs for which reshuffling variants are evaluated, the reshuffling variant has higher mean accuracy and lower reported standard deviation in each of the six settings. These results demonstrate the effectiveness of FedCDR in the evaluated $10{,}000$-client setting.

H.2 Scaling to Larger Models

We additionally evaluate FedCDR on two tasks with larger models. The first uses ResNet-18 (approximately 11.2 million parameters) on CIFAR-10 at $\rho = 1.0$, with 500 clients, 10% participation, $E = 5$ local epochs, and 400 communication rounds. The second uses LoRA fine-tuning of Qwen2.5-0.5B on AG News. LoRA is applied to the attention projections with rank $r = 8$; the experiment uses 100 clients, a Dirichlet partition with $\rho = 0.1$, 10% participation, and 200 communication rounds, with local optimizer steps matched across methods.

Table 13: Final top-1 test accuracy (%) on larger-model tasks. Entries are mean $\pm$ standard deviation over three independent runs.

Method	**ResNet-18 / CIFAR-10**	**Qwen2.5-0.5B / AG News**
CR+FedAvg	83.23 ± 0.56	89.00 ± 0.41
CR+FedProx	83.09 ± 0.45	90.24 ± 0.42
CR+SCAFFOLD	84.15 ± 0.37	91.51 ± 0.34
CR+FedDC	85.28 ± 0.29	91.86 ± 0.30
FedDR	85.90 ± 0.38	90.64 ± 0.42
FedCDR	$\mathbf{86.23 \pm 0.33}$	$\mathbf{92.09 \pm 0.33}$

As shown in Table 13, FedCDR achieves the highest mean accuracy on both tasks. Its improvement over the strongest baseline is 0.33 percentage points for ResNet-18 and 0.23 points for Qwen2.5-0.5B with LoRA. The latter result extends the evaluation to a language task and a larger pretrained backbone.

H.3 Robustness to Client Dropout

We examine complete non-response by allowing each scheduled client to fail to respond independently with probability $q \in \{0, 0.1, 0.3, 0.5\}$. The server skips nonresponding clients without retries, and their persistent states remain unchanged. All methods use the same dropout masks. We evaluate CIFAR-10 with $\rho = 1.0$, $E = 5$, and 400 scheduled communication rounds; Table 14 reports the resulting test accuracies.

Table 14: Final top-1 test accuracy (%) on CIFAR-10 with $\rho = 1.0$, $E = 5$, and 400 scheduled communication rounds under independent client non-response.

Method	$q = 0$	$q = 0.1$	$q = 0.3$	$q = 0.5$
CR+FedAvg	78.29	78.77	77.88	76.27
CR+FedProx	78.53	78.00	77.14	76.95
CR+SCAFFOLD	82.35	80.81	77.09	73.89
CR+FedDC	83.56	81.19	75.08	70.65
FedDR	81.97	80.36	79.87	78.04
FedCDR	**84.46**	**84.78**	**83.46**	**83.74**

FedCDR achieves the highest test accuracy at every evaluated dropout probability. At $q = 0.5$, it attains 83.74% accuracy, compared with 78.04% for the strongest baseline, FedDR, and 84.46% for FedCDR without dropout.

I Comparison of Nonconvex FL Convergence Guarantees

Complexity measure for FedCDR. We first distinguish meta-epochs from communication rounds. With n clients and C participating clients per round, each FedCDR meta-epoch contains $R = n/C$ communication rounds. After T meta-epochs, the total number of communication rounds is therefore $RT = nT/C$. For the parameter choices in Corollaries 4 and 2, respectively, the exact and inexact variants satisfy

$$\mathbb{E}\big[\|\nabla f(\tilde{x})\|^2\big] \le \begin{cases} \dfrac{125L}{T}\big(f(x^0) - f^*\big), & \text{exact,} \\ \dfrac{625L}{2T}\big(f(x^0) - f^* + 325LM\big), & \text{inexact.} \end{cases} \tag{121}$$

The inexact bound requires the mean-square proximal accuracy condition (6) for all stored client states and the accumulated error budget (8), where M is independent of T. Thus, for fixed n and C, both variants have $O(\epsilon^{-1})$ meta-epoch complexity and $O(n/(C\epsilon))$ communication-round complexity. The factor n/C is the number of rounds in a meta-epoch.

I.1 Comparison with Nonconvex FL Methods

Table 15: Representative convergence analyses for nonconvex FL. The guarantees apply under different algorithmic and stochastic assumptions and should not be interpreted as a uniform ranking.

Method	Main assumptions	Relevant guarantee and heterogeneity dependence
FedProx [Li et al., 2020a]	Smooth nonconvex losses with a lower Hessian bound; inexact local subproblems; B-local dissimilarity in the original analysis.	An $O(\epsilon^{-1})$ communication-round bound follows under a positive sufficient-decrease coefficient. The coefficient and its admissibility conditions depend on the dissimilarity parameter B.
SCAFFOLD [Karimireddy et al., 2020]	Smooth nonconvex losses and bounded-variance stochastic gradients; partial client participation.	Its round bound contains a stochastic-noise term of order $O(\epsilon^{-2})$ and an $O(\epsilon^{-1})$ term. Neither term requires a client-gradient dissimilarity parameter; the noise term vanishes when the stochastic gradient variance is zero.
FedDC [Gao et al., 2022]	Smooth nonconvex local losses with a lower Hessian bound and bounded client-dissimilarity parameter B.	The analysis establishes a per-round expected decrease with a positive coefficient depending on the dissimilarity and participation assumptions.
FedCDR	Smooth nonconvex local losses and a lower-bounded objective; exact proximal updates, or inexact updates satisfying the stated error conditions.	$O(\epsilon^{-1})$ meta-epochs, corresponding to $O(n/(C\epsilon))$ communication rounds for fixed n and C, without an explicit data-heterogeneity error term.

The original FedProx analysis [Li et al., 2020a] uses a B-local dissimilarity condition to ensure sufficient expected descent, while the FedDC analysis also invokes bounded client dissimilarity [Gao et al., 2022]. These observations concern the cited analyses and do not imply that every convergence analysis of these algorithms must depend on a dissimilarity measure.

SCAFFOLD provides a useful contrasting example: its nonconvex guarantee does not require a client-gradient dissimilarity bound [Karimireddy et al., 2020]. Writing N for the total number of clients, S for the number sampled per round, K for the local gradient steps, σ^2 for the stochastic gradient variance, and $F = f(x^0) - f^*$, its round complexity has the form

$$O\left(\frac{LF\sigma^2}{KS\epsilon^2} + \frac{LF}{\epsilon}\left(\frac{N}{S}\right)^{2/3}\right). \tag{122}$$

Table 16: Theoretical guarantees for representative client-reshuffling methods. The iteration counts and stationarity criteria are specified separately because the analyses use different objective assumptions.

Method	Objective and update model	Guarantee and heterogeneity dependence
RR–CLI [Malinovsky et al., 2023]	Convex or strongly convex smooth losses; reshuffling of clients and local data.	Convergence guarantees for convex objectives, including an $O(Q^{-2})$-type dependence after stepsize tuning in the strong-convexity setting, where Q counts communication rounds in the cited analysis. The bounds involve variance quantities at an optimum, such as $\tilde{\sigma}_*^2$; they do not give the same general nonconvex stationarity guarantee as FedCDR.
Clipped RR–CLI [Demidovich et al., 2024]	Generalized (L_0, L_1)-smooth nonconvex losses; client reshuffling, local steps, and clipping.	$O(\epsilon^{-1})$ meta-epochs for a generalized gradient-type stationarity criterion under suitable stepsize conditions. Heterogeneity quantities $\Delta_\star$ and $\Delta^\star$ appear in the bound and in sufficient stepsize conditions.
FedCDR	Standard L-smooth nonconvex losses; client reshuffling with Douglas–Rachford splitting; exact or controlled inexact proximal updates.	$O(\epsilon^{-1})$ meta-epochs for $\mathbb{E}\|\nabla f(\tilde{x})\|^2 \le \epsilon$; $O(n/(C\epsilon))$ communication rounds for fixed n, C. No explicit heterogeneity measure appears in the convergence bounds.

I.2 Comparison with Client-Reshuffling Methods

RR–CLI [Malinovsky et al., 2023] establishes rates under convexity assumptions. Its theoretical bounds include variance quantities evaluated at an optimum, such as

$$\tilde{\sigma}_*^2 := \frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x^*)\|^2. \tag{123}$$

This quantity measures the mismatch between client gradients at a global minimizer and can remain nonzero even when $\nabla f(x^*) = 0$.

Clipped RR–CLI extends client reshuffling to nonconvex objectives under generalized smoothness [Demidovich et al., 2024]. Its Theorem 4 and Corollary 4 control a criterion of the form

$$\mathbb{E}\left[\min_{0\le t<T} \min\left\{\frac{\|\nabla f(x_t)\|^2}{L_0}, \frac{\|\nabla f(x_t)\|}{L_1}\right\}\right] \le \epsilon. \tag{124}$$

Its sufficient stepsize conditions contain quantities $\Delta_\star$ and $\Delta^\star$ that reflect differences between local and global objective values. These quantities therefore affect the parameter choices required by the cited guarantee. On the other hand, generalized (L_0, L_1)-smoothness is broader than the standard L-smoothness assumed in our analysis, and (124) is not identical to the expected squared-gradient criterion for FedCDR.

J Limitations

Our evaluation is limited by computational resources and does not cover ImageNet-scale datasets or full-parameter federated training of large language models. Although we evaluate LoRA fine-tuning of Qwen2.5-0.5B, this does not establish scalability to full-parameter LLM training. Our convergence analysis assumes standard smoothness and guarantees stationarity for an iterate selected uniformly at random, rather than necessarily for the last iterate. Extending the analysis to generalized smoothness and establishing last-iterate convergence guarantees remain directions for future work.

K The Use of LLM

LLM-based tools were used to assist with manuscript grammar correcting. The authors reviewed and take responsibility for all technical content, experiments, and references.

L Broader Impacts

This paper proposes FedCDR, a federated learning optimization algorithm that addresses the data heterogeneity challenge under client-reshuffling-based participation schemes. As a foundational contribution to distributed optimization, we discuss its potential societal implications below.

Positive impacts. Federated learning is designed to enable collaborative model training without sharing raw data, thereby preserving data privacy. Our method improves the efficiency and robustness of this paradigm in heterogeneous settings, which may facilitate the broader adoption of privacy-preserving machine learning in domains such as healthcare, mobile computing, and finance. Furthermore, the client-reshuffling mechanism in FedCDR ensures fairer client participation compared to naive sampling strategies, potentially reducing representation bias that arises when certain clients are systematically underrepresented during training. The communication–computation trade-off supported by our method also makes federated learning more accessible to resource-constrained devices, promoting inclusivity in collaborative learning systems.

Potential negative impacts. As a general-purpose optimization method, FedCDR does not introduce risks beyond those inherent to federated learning and machine learning more broadly. However, we note that improved federated learning algorithms could lower the barrier to training models on sensitive distributed data, which, if deployed without proper governance, may raise concerns related to data misuse or unintended model behaviors. We encourage practitioners to adopt appropriate safeguards, including differential privacy mechanisms, secure aggregation protocols, and thorough model auditing, when deploying federated learning systems in sensitive applications. Our work does not involve human subjects, personally identifiable information, or the generation of harmful content.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The paper introduces FedCDR, which combines client reshuffling with Douglas–Rachford splitting and supports controlled inexact proximal updates. Under the stated smoothness and proximal-error assumptions, the analysis establishes $O(\epsilon^{-1})$ meta-epoch complexity, equivalent to $O(n/(C\epsilon))$ communication rounds, without an explicit heterogeneity error term; Section 6 reports the corresponding empirical evaluation.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Appendix J discusses limitations in evaluation scale, the distinction between LoRA fine-tuning and full-parameter LLM training, the standard-smoothness assumption, and the absence of a last-iterate convergence guarantee.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Theorems 1 and 3, together with Corollaries 2 and 4, state their assumptions and distinguish exact from inexact proximal updates, including the required accumulated-error budget. Their proofs and supporting lemmas are provided in Appendices B and C.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Algorithm 1, Section 6, and Appendix G document the training protocol, data-generation and partitioning schemes, model architectures, optimizer settings, preprocessing, and hyperparameter searches. These details provide a basis for independently reproducing the main experimental comparisons.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: Source code and executable reproduction scripts are not included in the camera-ready materials. The paper specifies FedCDR in Algorithm 1 and describes the benchmark and synthetic-data generation settings in Appendix G, but does not provide an openly accessible implementation.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 6 describes the datasets, baselines, client participation, local-epoch budgets, and evaluation protocol. Appendix G specifies the class-wise Dirichlet partition, model architectures, mini-batch SGD with momentum, batch sizes, weight decay, learning-rate schedule, search ranges, and dataset-specific normalization.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: The primary fixed-budget comparisons in Tables 1 and 2 report mean $\pm$ standard deviation over five independent runs, and Table 12 also reports mean $\pm$ standard deviation. The larger-model experiments in Table 13 report mean $\pm$ standard deviation over three independent runs, while several communication, trade-off, and dropout tables do not report error bars.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix G.2 specifies Ubuntu 20.04, a 24-core 2.80 GHz Intel processor, four NVIDIA L40S GPUs, and 256 GB RAM. Appendix E.3 reports measured execution times for selected time-to-target experiments and describes the sequential-client, single-GPU simulation setting.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The study develops a federated optimization method and uses public benchmark datasets without recruiting human subjects or collecting private participant data. Potential downstream privacy, security, and deployment concerns are discussed in Appendix L.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix L discusses potential benefits of more effective heterogeneous federated training and possible privacy, security, and misuse risks. It also identifies deployment safeguards, including secure aggregation, differential privacy, and model auditing.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release a newly trained pretrained language model, an image generator, or a scraped dataset that requires controlled access. Qwen2.5-0.5B is used only as an existing pretrained backbone in a LoRA fine-tuning evaluation.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper credits the benchmark datasets, synthetic-data generation protocols, and baseline methods used in the experiments, and identifies AG News and Qwen2.5-0.5B in Appendix H.2. The authors have reviewed the applicable asset licenses and terms of use.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification: FedCDR is a new algorithmic method, but the paper does not release a new dataset, model checkpoint, or software package as a standalone asset.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: No crowdsourcing tasks, participant recruitment, or human-subject experiments were conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

    Justification: The study uses public benchmark datasets and does not conduct research with recruited human subjects; therefore, participant-risk disclosures and IRB approval are not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as an important or original component of FedCDR's optimization algorithm; the LoRA experiment with Qwen2.5-0.5B is an evaluation task rather than an LLM-driven optimization method. The paper separately includes an author-use statement in Appendix K.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.