The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

CDMA: A Practical Cross-Device Federated Learning Algorithm for General Minimax Problems

Jiahao Xie[1] , **Chao Zhang**[*2], **Zebang Shen**[3], **Weijie Liu**[4,1], **Hui Qian**[1,5]
[1]College of Computer Science and Technology, Zhejiang University
[2]Advanced Technology Institute, Zhejiang University
[3]ETH Zürich
[4]Qiushi Academy for Advanced Studies, Zhejiang University
[5]State Key Lab of CAD&CG, Zhejiang University
xiejh@zju.edu.cn, zczju@zju.edu.cn, zebang.shen@inf.ethz.ch, westonhunter@zju.edu.cn, qianhui@zju.edu.cn

Abstract

Minimax problems arise in a wide range of important applications including robust adversarial learning and Generative Adversarial Network (GAN) training. Recently, algorithms for minimax problems in the Federated Learning (FL) paradigm have received considerable interest. Existing federated algorithms for general minimax problems require the full aggregation (i.e., aggregation of local model information from all clients) in each training round. Thus, they are inapplicable to an important setting of FL known as the cross-device setting, which involves numerous unreliable mobile/IoT devices. In this paper, we develop the first practical algorithm named CDMA for general minimax problems in the cross-device FL setting. CDMA is based on a Start-Immediately-With-Enough-Responses mechanism, in which the server first signals a subset of clients to perform local computation and then starts to aggregate the local results reported by clients once it receives responses from enough clients in each round. With this mechanism, CDMA is resilient to the low client availability. In addition, CDMA is incorporated with a lightweight global correction in the local update steps of clients, which mitigates the impact of slow network connections. We establish theoretical guarantees of CDMA under different choices of hyperparameters and conduct experiments on AUC maximization, robust adversarial network training, and GAN training tasks. Theoretical and experimental results demonstrate the efficiency of CDMA.

Introduction

During the last few years, minimax problems have found a surge of important applications such as AUC (area under the ROC curve) maximization (Ying, Wen, and Lyu 2016; Liu et al. 2020a), robust adversarial learning (Madry et al. 2018), and Generative Adversarial Network (GAN) (Goodfellow et al. 2014). Recently, distributed minimax algorithms in the Federated Learning (FL) paradigm have drawn significant attention due to the increasing concern of data privacy and the rapid growth of data volume (Mohri, Sivek, and Suresh 2019; Deng, Kamani, and Mahdavi 2020; Reisizadeh et al. 2020; Rasouli, Sun, and Rajagopal 2020; Beznosikov, Samokhin, and Gasnikov 2020; Hou et al. 2021; Guo et al. 2020; Yuan et al. 2021; Deng and Mahdavi 2021). Generally, these federated learning algorithms optimize a global minimax problem over multiple clients (data sources) under the coordination of a central server without transferring any client's private data (Kairouz et al. 2019), which helps protect data privacy. Besides, the local update strategy (i.e., performing multiple local update steps on clients before communicating with the server) is usually adopted in each training round of these algorithms to save communication efforts, which makes them scalable to large-scale problems.

Despite that existing federated minimax algorithms have achieved success in certain applications, there is a lack of practical algorithms for general minimax problems in an important FL setting, known as the *cross-device* setting. In this setting, the clients are numerous (up to 10^{10} (Kairouz et al. 2019)) unreliable mobile/IoT devices with relatively slow network connections. Actually, most existing federated minimax algorithms are designed for another setting known as the *cross-silo* setting (Reisizadeh et al. 2020; Rasouli, Sun, and Rajagopal 2020; Beznosikov, Samokhin, and Gasnikov 2020; Hou et al. 2021; Guo et al. 2020; Yuan et al. 2021; Deng and Mahdavi 2021; Sharma et al. 2022; Sun and Wei 2022), where the clients are a relatively small number of organizations or data centers with reliable network connections (Kairouz et al. 2019). Compared with the cross-silo setting, it is generally more challenging to solve a problem in the cross-device setting due to the low client availability and relatively slow network connections (Kairouz et al. 2019). Mohri, Sivek, and Suresh (2019) and Deng, Kamani, and Mahdavi (2020) investigate a special constrained problem, where the objective function is linear w.r.t. the dual variable, and propose two cross-device federated minimax algorithms. However, these algorithms heavily rely on the linear structure, which makes them infeasible to general minimax problems including AUC maximization, robust adversarial learning, and GAN training.

In this paper, we develop the first practical federated algorithm named Cross-Device Minimax Averaging (CDMA) that applies to general minimax problems in the cross-device setting. Each training round of CDMA consists of two phases: (i) the gradient collection phase and (ii) the parameter update phase. In the first phase, the server sends the global model parameters to a subset of clients, and aggre-

---

[*]Corresponding author.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

gates local gradients computed at these clients to construct a global correction for subsequent local updates. This correction mitigates the issue of *client drift* (i.e., the phenomenon that local models on clients are shifted away from the global model (Karimireddy et al. 2020)), and in turn mitigates the influence of slow network connections. In the second phase, the server sends the global correction and model parameters to another subset of clients, which perform multiple local updates before sending the resulting local models back to the server. Note that only a small subset of clients is required in each training round, which provides robustness against the low client availability in the cross-device setting. Our major contributions are summarized as follows.

1. We design a Start-Immediately-With-Enough-Responses (SIWER) mechanism for the server. That is, in each round, the server first sends a synchronization signal to a subset of clients and then starts constructing the global correction/updating the global model as long as it receives local results from enough clients. With this mechanism, CDMA is tolerant of client failure, and thus significantly mitigates the impact of the low client availability of the cross-device FL setting.
2. We construct a stochastic recursive momentum as the global correction term. As this correction term utilizes historical gradient information in a recursive manner, it only needs to aggregate information from a small portion of clients, and can greatly reduce the client drift without much additional communication cost.
3. We establish theoretical guarantees for CDMA under different settings of hyperparameters. In particular, we prove that, to reach a stationary point up to ε accuracy, the communication cost of CDMA has an $\tilde{\mathcal{O}}(1/\varepsilon^3)$[1] dependence on ε. We also prove that under appropriate assumptions, the communication cost of CDMA reduces as the number of local steps increases, which is the first such guarantee for federated minimax algorithms.

Empirical studies on AUC maximization, robust adversarial neural network training, and GAN training tasks demonstrate the efficiency of the proposed algorithm.

Notation. We use bold lowercase symbols (e.g., $\mathbf{x}$) to denote vectors. For a vector $\mathbf{x}$, we denote its j-th coordinate as $[\mathbf{x}]_j$. For a function $f(\mathbf{x}, \mathbf{y}) : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$, $\nabla_{\mathbf{x}} f$ and $\nabla_{\mathbf{y}} f$ denote its partial derivatives with respect to the first and second variables, respectively. For ease of notation, we also denote $\mathbf{z} := (\mathbf{x}^T, \mathbf{y}^T)^T$, $f(\mathbf{z}) := f(\mathbf{x}, \mathbf{y})$, and $\nabla f(\mathbf{x}, \mathbf{y}) := (\nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{y})^T, \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})^T)^T$. The Euclidean norm of a vector $\mathbf{x}$ is denoted by $\|\mathbf{x}\|$.

Related Works

Single-machine minimax optimization algorithms. Minimax optimization has a long history dating back to (Brown 1951). The majority of existing works on minimax problems focus on the convex-concave regime (Korpelevich 1976; Nemirovski 2004; Nedić and Ozdaglar 2009; Liu and Wang 2015; Palaniappan and Bach 2016; Chavdarova et al. 2019). Recently, there have emerged a surge of algorithms for more general nonconvex minimax problems[2] (Sanjabi, Razaviyayn, and Lee 2018; Nouiehed et al. 2019; Lin, Jin, and Jordan 2020; Luo et al. 2020; Xu et al. 2020; Qiu et al. 2020; Yang, Kiyavash, and He 2020).

Federated minimax optimization algorithms. To tackle large-scale minimax problems, a few distributed minimax algorithms have been proposed in conventional centralized and decentralized settings (Srivastava, Nedić, and Stipanović 2011; Mateos-Núnez and Cortés 2015; Liu et al. 2020b; Beznosikov, Samokhin, and Gasnikov 2020; Rogozin et al. 2021). Recently, with the growing concern of privacy, distributed minimax optimization in the FL paradigm has received considerable interest. Several algorithms based on the local update strategy are proposed to solve minimax problems in the cross-silo FL setting (Reisizadeh et al. 2020; Rasouli, Sun, and Rajagopal 2020; Beznosikov, Samokhin, and Gasnikov 2020; Hou et al. 2021; Guo et al. 2020; Yuan et al. 2021; Deng and Mahdavi 2021; Sharma et al. 2022; Sun and Wei 2022). Though the algorithms in (Reisizadeh et al. 2020; Rasouli, Sun, and Rajagopal 2020; Beznosikov, Samokhin, and Gasnikov 2020; Hou et al. 2021; Guo et al. 2020; Yuan et al. 2021; Deng and Mahdavi 2021) perform multiple local update steps in each round to reduce the communication cost, they require *full aggregation* (i.e., aggregation of local model information from all clients), and thus do not apply to the cross-device setting. For a special constrained minimax problem of the form $\min_{\mathbf{x}} \max_{\mathbf{y} \in \mathbb{R}^N_+, \|\mathbf{y}\|_1 = 1} \sum_{i=1}^N [\mathbf{y}]_i f_i(\mathbf{x})$, where N denotes the total number of clients and f_i is the local loss function of client i, Mohri, Sivek, and Suresh (2019) and Deng, Kamani, and Mahdavi (2020) propose algorithms that in each round allow only a subset of clients to participate in training. However, the algorithms in (Mohri, Sivek, and Suresh 2019) and (Deng, Kamani, and Mahdavi 2020) heavily depend on the linear structure of the above special constrained problem, and do not apply to general minimax problems.

Problem Setting and Algorithm

We consider the following general unconstrained minimax optimization problem in the cross-device FL setting:

$$\min_{\mathbf{x} \in \mathbb{R}^p} \max_{\mathbf{y} \in \mathbb{R}^q} \left\{ f(\mathbf{x}, \mathbf{y}) := \mathbb{E}_{i \sim \mathcal{D}}[f_i(\mathbf{x}, \mathbf{y})] \right\}, \tag{1}$$

where $\mathcal{D}$ represents the client distribution, $f_i(\mathbf{x}, \mathbf{y}) := \frac{1}{n_i} \sum_{j=1}^{n_i} F_i(\mathbf{x}, \mathbf{y}; \zeta_{i,j})$ is the local loss function of client i, and $\zeta_{i,1}, \ldots, \zeta_{i,n_i}$ denote the local data points on client i. Generally, $\mathcal{D}$ can be any distribution over N clients. A typical example of $\mathcal{D}$ is the uniform distribution, which corresponds to the loss function $f = \frac{1}{N} \sum_{i=1}^N f_i$. Since the number of clients may be extremely large ($N \approx 10^{10}$) and the client availability is low in the cross-device setting, it is unlikely to access all clients at any one time during the learning procedure. This precludes the usage of cross-silo FL algorithms that rely on the full aggregation strategy.

---

[1] The symbol $\tilde{\mathcal{O}}$ suppresses a logarithmic factor in ε^{-1}.

[2] Nonconvex minimax problems include nonconvex-concave and nonconvex-nonconcave problems.

Algorithm 1: CDMA on the server.

```
Input: the initial model parameters $(\mathbf{x}_{-1}, \mathbf{y}_{-1})$, the number of rounds $T$, and the numbers $\{S_t\}_{t=0}^{T-1}$ of received clients.
$\mathbf{x}_0 \leftarrow \mathbf{x}_{-1}, \mathbf{y}_0 \leftarrow \mathbf{y}_{-1}$;
for $t = 0, 1, \dots, T-1$ do
    Gradient collection phase:
    Send $\{\mathbf{x}_t, \mathbf{y}_t, \mathbf{x}_{t-1}, \mathbf{y}_{t-1}\}$ to a random subset $\hat{\mathcal{S}}_t'$ of clients;
    Receive local gradient information from each client $i \in \mathcal{S}_t' \subseteq \hat{\mathcal{S}}_t'$, where $\mathcal{S}_t'$ is the subset of $\hat{\mathcal{S}}_t'$ containing the first $S_t$ clients that successfully respond;
    Compute $\mathbf{u}_t$ and $\mathbf{v}_t$ by Eq. (2);
    Parameter update phase:
    Send $(\mathbf{u}_t, \mathbf{v}_t)$ and $(\mathbf{x}_t, \mathbf{y}_t)$ to another random subset $\hat{\mathcal{S}}_t$ of clients;
    Receive local iterates $(\mathbf{x}_{t,i}^{(K)}, \mathbf{y}_{t,i}^{(K)})$ from each client $i \in \mathcal{S}_t \subseteq \hat{\mathcal{S}}_t$, where $\mathcal{S}_t$ is the subset of $\hat{\mathcal{S}}_t$ containing the first $S_t$ clients that successfully respond;
    $\mathbf{x}_{t+1} \leftarrow \frac{1}{|\mathcal{S}_t|}\sum_{i\in\mathcal{S}_t}\mathbf{x}_{t,i}^{(K)}, \mathbf{y}_{t+1} \leftarrow \frac{1}{|\mathcal{S}_t|}\sum_{i\in\mathcal{S}_t}\mathbf{y}_{t,i}^{(K)}$;
end for
```

Algorithm 2: CDMA on client $i \in \hat{\mathcal{S}}_t$ or $\hat{\mathcal{S}}_t'$.

```
Input: the number of local steps $K$, step sizes $\eta_t$ and $\gamma_t$, weight parameters $\alpha_t$ and $\beta$.
Gradient collection phase (for $i \in \hat{\mathcal{S}}_t'$):
1:  Receive $(\mathbf{x}_t, \mathbf{y}_t)$ and $(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})$ from the server.
2:  Compute $\nabla f_i(\mathbf{x}_t, \mathbf{y}_t) - (1-\alpha_t)\nabla f_i(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})$ and send it to the server;
Parameter update phase (for $i \in \hat{\mathcal{S}}_t$):
3:  Receive $(\mathbf{x}_t, \mathbf{y}_t)$ and $(\mathbf{u}_t, \mathbf{v}_t)$ from the server.
4:  Initialize local model $\mathbf{x}_{t,i}^{(0)} \leftarrow \mathbf{x}_t, \mathbf{y}_{t,i}^{(0)} \leftarrow \mathbf{y}_t$;
5:  for $k = 0, 1, \dots, K-1$ do
6:      Sample a minibatch $\mathcal{B}_{t,i}^{(k)}$ from local data;
7:      Compute $\mathbf{d}_{\mathbf{x},t,i}^{(k)}$ and $\mathbf{d}_{\mathbf{y},t,i}^{(k)}$ by (3);
8:      $\mathbf{x}_{t,i}^{(k+1)} \leftarrow \mathbf{x}_{t,i}^{(k)} - \eta_t \mathbf{d}_{\mathbf{x},t,i}^{(k)}$;
9:      $\mathbf{y}_{t,i}^{(k+1)} \leftarrow \mathbf{y}_{t,i}^{(k)} + \gamma_t \mathbf{d}_{\mathbf{y},t,i}^{(k)}$;
10: end for
11: Send $(\mathbf{x}_{t,i}^{(K)}, \mathbf{y}_{t,i}^{(K)})$ to the server;
```

To solve problem (1), we propose an algorithm named Cross-Device Minimax Averaging (CDMA), which is detailed in Algorithms 1-2. CDMA is based on a Start-Immediately-With-Enough-Responses (SIWER) mechanism. In this mechanism, the server first sends a synchronization signal to a subset of clients, collects local gradients/models from the clients, and starts to aggregate the received information to construct the global correction/update the global model once a sufficient number of devices have reported results. The SIWER mechanism is detailed in the following two phases.

(i) Gradient collection phase. In this phase, the server aggregates local gradients to construct a stochastic recursive momentum as the correction direction, which is inspired by recent global correction techniques in the federated minimization literature (Karimireddy et al. 2020, 2021). Specifically, the server sends $(\mathbf{x}_t, \mathbf{y}_t)$ and $(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})$ to a random subset of clients $\hat{\mathcal{S}}_t'$, collects local gradients from this subset, and computes the correction direction. Note that as some clients may be temporarily unavailable, the server only collects messages from the first S_t responded clients and then proceed. We denote these S_t clients as $\mathcal{S}_t'$. To ensure that at least S_t clients respond, the server can select a large enough $\hat{\mathcal{S}}_t'$. CDMA leverages the similarity of consecutive iterates to construct a lightweight estimator named the recursive momentum-based estimator, which is recursively constructed as

$$\begin{cases} \mathbf{u}_t = & (1-\alpha_t)\mathbf{u}_{t-1} + \frac{1}{|\mathcal{S}_t'|}\sum_{i\in\mathcal{S}_t'}(\nabla_{\mathbf{x}} f_i(\mathbf{x}_t, \mathbf{y}_t) \\ & -(1-\alpha_t)\nabla_{\mathbf{x}} f_i(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})), \\ \mathbf{v}_t = & (1-\alpha_t)\mathbf{v}_{t-1} + \frac{1}{|\mathcal{S}_t'|}\sum_{i\in\mathcal{S}_t'}(\nabla_{\mathbf{y}} f_i(\mathbf{x}_t, \mathbf{y}_t) \\ & -(1-\alpha_t)\nabla_{\mathbf{y}} f_i(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})), \end{cases} \tag{2}$$

for $t \geq 1$, where $\alpha_t \in (0, 1]$ is a hyperparameter. For $t = 0$, $\mathbf{u}_0$ (resp., $\mathbf{v}_0$) is set to $\frac{1}{|\mathcal{S}_0'|}\sum_{i\in\mathcal{S}_0'}\nabla_{\mathbf{x}} f_i(\mathbf{x}_0, \mathbf{y}_0)$ (resp., $\frac{1}{|\mathcal{S}_0'|}\sum_{i\in\mathcal{S}_0'}\nabla_{\mathbf{y}} f_i(\mathbf{x}_0, \mathbf{y}_0)$). Similar recursive momentum techniques have been extensively used in the optimization literature and generally lead to reduced variance with small batch sizes (Cutkosky and Orabona 2019; Xie et al. 2020; Zhang et al. 2020; Qiu et al. 2020; Tran-Dinh et al. 2021). Note that if $\alpha_t \equiv 1$, (2) degenerates to the minibatch estimator, i.e., $\mathbf{u}_t = \frac{1}{|\mathcal{S}_t'|}\sum_{i\in\mathcal{S}_t'}\nabla_{\mathbf{x}} f_i(\mathbf{x}_t, \mathbf{y}_t)$ and $\mathbf{v}_t = \frac{1}{|\mathcal{S}_t'|}\sum_{i\in\mathcal{S}_t'}\nabla_{\mathbf{y}} f_i(\mathbf{x}_t, \mathbf{y}_t)$. The minibatch estimator is the most widely-used and the simplest estimator in the optimization literature, which usually needs a large batch size to control the variance below a desired level (Johnson and Zhang 2013).

(ii) Parameter update phase. In this phase, clients update local parameters using both local stochastic gradients and the global correction constructed in the first phase. Specifically, the server first sends the current parameters $(\mathbf{x}_t, \mathbf{y}_t)$ and the global correction direction $(\mathbf{u}_t, \mathbf{v}_t)$ to a random subset of clients $\hat{\mathcal{S}}_t$, which is potentially different from $\hat{\mathcal{S}}_t'$ in the previous phase. Then, each active client in $\hat{\mathcal{S}}_t$ performs K local iterations and sends its local parameters to the server. Again, the server only collects messages from the first S_t responded clients, which we denote as $\mathcal{S}_t$, as some clients may be temporarily unavailable. Finally, the server averages the local parameters from $\mathcal{S}_t$ to produce the next global parameters. In each local iteration (line 6 to 9 in Algorithm 2), CDMA simultaneously takes a descent step on the local primal variable and an ascent step on the dual one. The primal and dual update directions are computed as

$$\begin{cases} \mathbf{d}_{\mathbf{x},t,i}^{(k)} = \nabla_{\mathbf{x}} F_i(\mathbf{x}_{t,i}^{(k)}, \mathbf{y}_{t,i}^{(k)}; \mathcal{B}_{t,i}^{(k)}) + \beta(\mathbf{u}_t - \nabla_{\mathbf{x}} F_i(\mathbf{x}_t, \mathbf{y}_t; \mathcal{B}_{t,i}^{(k)})) \\ \mathbf{d}_{\mathbf{y},t,i}^{(k)} = \nabla_{\mathbf{y}} F_i(\mathbf{x}_{t,i}^{(k)}, \mathbf{y}_{t,i}^{(k)}; \mathcal{B}_{t,i}^{(k)}) + \beta(\mathbf{v}_t - \nabla_{\mathbf{y}} F_i(\mathbf{x}_t, \mathbf{y}_t; \mathcal{B}_{t,i}^{(k)})), \end{cases} \tag{3}$$

where $\beta \in \{0, 1\}$ is a hyperparameter.

According to the choices of the weight parameters β and α_t, there are three specific versions of CDMA.

1. *CDMA with No Correction (CDMA-NC).* If $\beta = 0$, the local update directions in (3) are simply local minibatch gradients. Hence, the gradient collection phase is not needed, and CDMA degenerates to a single-phase algorithm, which we refer to as CDMA with No Correction (CDMA-NC).
2. *CDMA with $\beta = 1$ and $\alpha_t \equiv 1$ (CDMA-ONE).* When $\beta = 1$ and $\alpha_t \equiv 1$, the local update directions incorporate global correction, which is computed using the degenerate minibatch estimator. We denote CDMA with $\beta = 1$ and $\alpha_t \equiv 1$ as CDMA-ONE.
3. *CDMA with $\beta = 1$ and $\alpha_t \not\equiv 1$ (CDMA-ADA).* In the case $\beta = 1$ and $\alpha_t \not\equiv 1$, the correction direction is computed using the recursive momentum-based estimator. We refer to this version of CDMA as CDMA-ADA.

We highlight that since only a random subset of clients is involved in each round, CDMA is resilient to the low client availability. Further, when $\beta = 1$, the local update directions in (3) approximate global gradients $\nabla_{\mathbf{x}} f(\mathbf{x}_{t,i}^{(k)}, \mathbf{y}_{t,i}^{(k)})$ and $\nabla_{\mathbf{y}} f(\mathbf{x}_{t,i}^{(k)}, \mathbf{y}_{t,i}^{(k)})$ if $(\mathbf{x}_{t,i}^{(k)}, \mathbf{y}_{t,i}^{(k)})$ is close to $(\mathbf{x}_t, \mathbf{y}_t)$. This mitigates the issue of client drift. As we shall see in the next section, incorporating the global correction in local update steps also reduces the communication complexity.

We note that the concurrent work (Sun and Wei 2022) uses a similar global correction for cross-silo federated minimax learning. However, the computation of their global correction requires the full aggregation and thus do not apply to the cross-device setting. Besides, the analysis in (Sun and Wei 2022) focuses on strongly-convex-strongly-concave minimax problems, while we consider a class of nonconvex-nonconcave problems, which are more challenging.

Convergence Analysis

This section establishes convergence guarantees for CDMA. Here, we only present the major results and defer detailed analyses to the long version of the paper. The following common assumptions are needed throughout our analyses.

Assumption 1 (Bounded gradient dissimilarity). *There exist positive constants σ_1 and σ_2 such that $\forall \mathbf{x} \in \mathbb{R}^p, \mathbf{y} \in \mathbb{R}^q$,*

$$\begin{cases} \mathbb{E}_{i\sim\mathcal{D}}[\|\nabla_{\mathbf{x}} f_i(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{y})\|^2] \le \sigma_1^2, \\ \mathbb{E}_{i\sim\mathcal{D}}[\|\nabla_{\mathbf{y}} f_i(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y})\|^2] \le \sigma_2^2. \end{cases}$$

Assumption 2 (Lipschitz continuous gradients). *There exists a positive constant $L_f > 0$ such that for any i and $\zeta \in \{\zeta_{i,1}, \dots, \zeta_{i,n_i}\}$, the function $F_i(\cdot,\cdot;\zeta)$ has L_f-Lipschitz continuous gradients, i.e., $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^p$ and $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^q$,*

$$\left\| \begin{bmatrix} \nabla_{\mathbf{x}} F_i(\mathbf{x}_1,\mathbf{y}_1;\zeta) - \nabla_{\mathbf{x}} F_i(\mathbf{x}_2,\mathbf{y}_2;\zeta) \\ \nabla_{\mathbf{y}} F_i(\mathbf{x}_1,\mathbf{y}_1;\zeta) - \nabla_{\mathbf{y}} F_i(\mathbf{x}_2,\mathbf{y}_2;\zeta) \end{bmatrix} \right\| \le L_f \left\| \begin{bmatrix} \mathbf{x}_1 - \mathbf{x}_2 \\ \mathbf{y}_1 - \mathbf{y}_2 \end{bmatrix} \right\|.$$

Assumption 3 (Polyak-Łojasiewicz (PŁ) condition). *There exists a constant $\mu > 0$ such that $\forall \mathbf{x} \in \mathbb{R}^p, \mathbf{y} \in \mathbb{R}^q$,*

$$\|\nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y})\|^2 \ge 2\mu\big(\max_{\mathbf{y}' \in \mathbb{R}^q} f(\mathbf{x},\mathbf{y}') - f(\mathbf{x},\mathbf{y})\big).$$

We also assume that each client that successfully responds to the server in round t follows the underlying client distribution $\mathcal{D}$, which is a common assumption in the federated learning literature (Li et al. 2018, 2020; Karimireddy et al. 2021; Acar et al. 2021). In our analyses, we denote $S := \min\{S_0, \dots, S_{T-1}\}$.

A natural metric for measuring the performance of an algorithm on problem (1) is the gradient norm of the function $\Phi(\mathbf{x}) := \max_{\mathbf{y}\in\mathbb{R}^q} f(\mathbf{x},\mathbf{y})$. This metric is commonly used in analyzing algorithms for nonconvex-PŁ or nonconvex-strongly-concave minimax problems (Reisizadeh et al. 2020; Deng and Mahdavi 2021; Lin, Jin, and Jordan 2020; Luo et al. 2020). Note that $\Phi(\mathbf{x})$ has L_Φ-Lipschitz continuous gradients according to (Nouiehed et al. 2019), where $L_\Phi := (1 + \kappa/2)L_f$. For some $\varepsilon > 0$, a point $\mathbf{x} \in \mathbb{R}^p$ is said to be ε-stationary if $\|\nabla\Phi(\mathbf{x})\| \le \varepsilon$. Once an approximate primal solution $\tilde{\mathbf{x}}$ is obtained, one can easily find an approximate dual solution by solving $\max_{\mathbf{y}} f(\tilde{\mathbf{x}}, \mathbf{y})$.

Analysis of CDMA-NC

We first provide the theoretical analysis for CDMA-NC. Besides Assumptions 1-3, the bounded local variance assumption below is also needed.

Assumption 4 (Bounded local variance). *For any $(\mathbf{x},\mathbf{y}) \in \mathbb{R}^{p\times q}$, and any client i,*

$$\begin{cases} \frac{1}{n_i}\sum_{j=1}^{n_i} \|\nabla_{\mathbf{x}} F_i(\mathbf{x},\mathbf{y};\zeta_{i,j}) - \nabla_{\mathbf{x}} f_i(\mathbf{x},\mathbf{y})\|^2 \le G_1^2, \\ \frac{1}{n_i}\sum_{j=1}^{n_i} \|\nabla_{\mathbf{y}} f_i(\mathbf{x},\mathbf{y};\zeta_{i,j}) - \nabla_{\mathbf{y}} f_i(\mathbf{x},\mathbf{y})\|^2 \le G_2^2. \end{cases}$$

We summarize the convergence result of CDMA-NC in Theorem 1 below and defer the detailed analysis to the long version of this paper.

Theorem 1. *Suppose that Assumptions 1, 2, and 4 hold. Define the step sizes of CDMA-NC as $\gamma_t = \min\left\{\sqrt{\frac{20\mathcal{L}_0 S}{L_f T K^2 \sigma_2^2}}, \left(\frac{30\mathcal{L}_0}{L_f^2(\sigma_2^2+G_2^2)TK^3}\right)^{1/3}, \frac{1}{87 L_f K}\right\}$ and $\eta_t = \min\left\{\sqrt{\frac{20\mathcal{L}_0 S}{7(L_\Phi+L_f)TK^2\sigma_1^2}}, \left(\frac{3\mathcal{L}_0}{L_f^2(\sigma_1^2+G_1^2)TK^3}\right)^{1/3}, \frac{\gamma_t}{21\kappa^2}\right\}$. If the minibatches $\mathcal{B}_{t,i}^{(0)}, \dots, \mathcal{B}_{t,i}^{(K-1)}$ in CDMA-NC are drawn in a random reshuffling manner and K is an integral multiple of the epoch length, then*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla\Phi(\mathbf{x}_t)\|^2] = \mathcal{O}\Big(\frac{\kappa^2}{T^{2/3}} + \frac{\kappa^2}{\sqrt{ST}}\Big),$$

where $\mathcal{O}$ hides the dependence on $\mathcal{L}_0$, L_f, σ_1, and σ_2.

Theorem 1 indicates that CDMA-NC finds an ε-stationary point in $\mathcal{O}(\frac{\kappa^3}{\varepsilon^3} + \frac{\kappa^4}{S\varepsilon^4})$ communication rounds. Actually, when all clients participate in each round, CDMA-NC reduces to the local SGDA algorithm in (Sharma et al. 2022), and the second term of our communication cost can be canceled by slightly modifying our analysis, which recovers the $\mathcal{O}(\kappa^3/\varepsilon^3)$ communication cost in (Sharma et al. 2022). As our paper focuses on the cross-device setting where the total number of clients is extremely large, it is unrealistic to require all clients to participate in each round, and Theorem 1 shows that as long as $S = \mathcal{O}(\kappa/\varepsilon)$, CDMA-NC achieves the same $\mathcal{O}(\kappa^3/\varepsilon^3)$ communication cost as local SGDA.

Analysis of CDMA-ONE

The analysis of CDMA-ONE is similar to that of CDMA-NC. Compared with CDMA-NC, CDMA-ONE incorporates global correction into local updates. Consequently, CDMA-ONE has a smaller client drift compared with that in CDMA-NC, which in turn leads to a better convergence rate. The theorem below establishes the convergence rate of CDMA-ONE.

Theorem 2. *Define the step sizes η_t and γ_t of CDMA-ONE as $\eta_t = \min\left\{\frac{\gamma_t}{21\kappa^2}, \sqrt{\frac{40\mathcal{L}_0 S}{21(L_\Phi+L_f)TK^2\sigma_1^2}}\right\}$ and $\gamma_t = \min\left\{\frac{1}{87L_fK}, \sqrt{\frac{4\mathcal{L}_0 S}{3L_fTK^2\sigma_2^2}}\right\}$, where $\mathcal{L}_0 := \Phi(\mathbf{x}_0) - \Phi(\mathbf{x}^*) + \frac{1}{20}(\Phi(\mathbf{x}_0) - f(\mathbf{z}_0))$. If Assumptions 1-3 hold, then*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\Phi(\mathbf{x}_t)\|^2] = \mathcal{O}(\frac{\kappa^2}{T} + \frac{\kappa^2}{\sqrt{ST}}),$$

where $\mathcal{O}$ hides the dependence on $\mathcal{L}_0$, L_f, σ_1, and σ_2.

By Theorem 2, CDMA-ONE has a communication cost of $\mathcal{O}(\kappa^2/\varepsilon^2+\kappa^4/(S\varepsilon^4))$, better than the $\mathcal{O}(\kappa^3/\varepsilon^3+\kappa^4/(S\varepsilon^4))$ cost of CDMA-NC. This demonstrates that incorporating global gradient estimates into local updates is more efficient than merely using local stochastic gradients.

Remark 1. *When $K = 1$, CDMA-ONE coincides with Parallel SGDA, which directly parallelizes the single-machine SGDA algorithm (Lin, Jin, and Jordan 2020) over the clients. Under the stronger assumption that $f(\mathbf{x}, \cdot)$ is μ-strongly concave given any $\mathbf{x}$, Lin, Jin, and Jordan (2020) proves that Parallel SGDA has an $\mathcal{O}(\kappa^2/\varepsilon^2 + \kappa^3/(S\varepsilon^5))$ communication complexity*[3]*. This complexity is inferior to that of CDMA-ONE in terms of the dependence on ε.*

Analysis of CDMA-ADA

As mentioned in the previous section, CDMA-ADA utilizes the variance-reduced estimator (2) which incorporates historical gradient information. This results in a more accurate global correction direction compared to the degenerate one used in CDMA-ONE and accelerates the convergence. The convergence rate of CDMA-ADA is presented in the next theorem.

Theorem 3. *Define α_t, η_t, and γ_t in CDMA-ADA as $\alpha_t = 200000L_f^2K^2\gamma_t^2$, $\eta_t = \frac{\gamma_t}{12\kappa^2}$, and $\gamma_t = \min\{\frac{1}{2L_f}, \frac{5.78\text{e}-4}{L_fK}, \frac{S^{1/3}\hat{\mathcal{L}}_0^{1/3}}{L_f^{2/3}K(t+1)^{1/3}(\sigma_1^2/\kappa^2+\sigma_2^2)^{1/3}}\}$, where $\hat{\mathcal{L}}_0 := \Phi(\mathbf{x}_0) - \Phi(\mathbf{x}^*) + \frac{1}{7}(\Phi(\mathbf{x}_0) - f(\mathbf{z}_0))$. Suppose that Assumptions 1-3 hold and $S \geq (\sigma_1^2/\kappa^2 + \sigma_2^2)/(L_f\hat{\mathcal{L}}_0)$. Then,*

$$\frac{1}{KT}\sum_{t=0}^{T-1}\frac{1}{|\mathcal{S}_t|}\sum_{k=0}^{K-1}\sum_{i\in\mathcal{S}_t}\mathbb{E}[\|\nabla\Phi(\mathbf{x}_{t,i}^{(k)})\|^2]$$
$$= \tilde{\mathcal{O}}(\kappa^2/T + \kappa^2(1+1/K)/(S^{1/3}T^{2/3})),$$

where $\tilde{\mathcal{O}}$ hides logarithmic factors and the dependence on $\mathcal{L}_0$, L_f, σ_1, and σ_2.

Theorem 3 indicates that the communication cost of CDMA-ADA to reach an ϵ-stationary point is

$$\tilde{\mathcal{O}}(\kappa^2/\varepsilon^2 + \kappa^3(1+1/K)^{3/2}/(S^{1/2}\varepsilon^3)), \tag{4}$$

which outperforms those of CDMA-NC and CDMA-ONE obtained in previous subsections. Actually, the communication complexity bound of CDMA-ADA can be further improved in terms of the dependence on the condition number κ under the following δ-bounded Hessian dissimilarity (δ-BHD) assumption (Shamir, Srebro, and Zhang 2014; Arjevani and Shamir 2015; Reddi et al. 2016).

Assumption 5 (δ-bounded Hessian dissimilarity)**.** *There exists a constant $\delta > 0$ such that for any client i and any $\zeta \in \{\zeta_{i,1}, \ldots, \zeta_{i,n_i}\}$, $\|\nabla^2 F_i(\mathbf{x}, \mathbf{y}; \zeta) - \nabla^2 f(\mathbf{x}, \mathbf{y})\| \leq 2\delta$.*

This assumption is widely used in the optimization literature to characterize the second-order similarity of objectives, which usually results in an improved convergence rate (Shamir, Srebro, and Zhang 2014; Arjevani and Shamir 2015; Reddi et al. 2016; Meng et al. 2020) compared with those rates obtained merely under first-order conditions. Note that under Assumption 2, Assumption 5 immediately holds with $\delta = L_f$. Actually, δ reduces as the data heterogeneity on different clients decreases and $\delta = 0$ when the data on different clients follow the same distribution. With this additional δ-BHD assumption, we prove that CDMA-ADA has the following modified communication cost (see the long version of this paper for details)

$$\tilde{\mathcal{O}}(\tfrac{\kappa}{\varepsilon^2}(\tfrac{\kappa}{K} + c) + (1 + \tfrac{\kappa}{cK})^3\tfrac{c\kappa^2}{S^{1/2}\varepsilon^3}). \tag{5}$$

The complexity bound (5) decreases as the number of local steps K increases. If we set $K = 1$, the bound in (5) reads $\tilde{\mathcal{O}}(\frac{\kappa^2}{\varepsilon^2} + \frac{\kappa^5}{S^{1/2}c^2\varepsilon^3})$, which is inferior to the bound in (4). When $K \geq \kappa/c$, the complexity becomes $\tilde{\mathcal{O}}(\frac{c\kappa}{\varepsilon^2} + \frac{c\kappa^2}{S^{1/2}\varepsilon^3})$ and outperforms that in (4) as $c := \max\{1, \delta/\mu\} \leq \kappa$. By contrast, in Theorem 3 where the δ-BHD assumption is absent (i.e., $\delta = L_f$), the order of the communication complexity of CDMA-ADA remains the same, though it also decreases as K increases.

Note that the complexities of CDMA-NC and CDMA-ONE obtained in Theorems 1 and 2 are independent of K. Nevertheless, our experimental results show that multiple local steps result in better performance for these two algorithms (check the next section). Actually, similar gaps between the theoretical analyses and the practical performance on the effect of local steps widely exist in the FL literature (Khaled, Mishchenko, and Richtárik 2020; Karimireddy et al. 2020; Beznosikov, Samokhin, and Gasnikov 2020; Karimireddy et al. 2021). Recently, several studies on lower bounds of communication complexities for distributed optimization under different assumptions show that local steps do not provide speed-up in some settings (Woodworth, Patel, and Srebro 2020; Beznosikov, Samokhin, and Gasnikov 2020). The investigation on the communication lower bound and whether local steps improve the communication complexities of CDMA-NC and CDMA-ONE under other assumptions is left for future work.

We compare the communication complexities obtained in this section with that of Parallel SGDA in Table 1.

---

[3] Note that Lin, Jin, and Jordan (2020) also shows another complexity of $\mathcal{O}(\kappa^2/\varepsilon^2)$ if $S = \mathcal{O}(\kappa/\varepsilon^2)$. However, when ε is small, such setting of S is unsuitable for cross-device FL where only a fraction of clients are available at one time.

Algorithm	Communication complexity
Parallel SGDA	$\mathcal{O}(\frac{\kappa^2}{\varepsilon^2} + \frac{\kappa^3}{S\varepsilon^5})^\dagger$
CDMA-NC	$\mathcal{O}(\frac{\kappa^3}{\varepsilon^3} + \frac{\kappa^4}{S\varepsilon^4})$
CDMA-ONE	$\mathcal{O}(\frac{\kappa^2}{\varepsilon^2} + \frac{\kappa^4}{S\varepsilon^4})$
CDMA-ADA	$\tilde{\mathcal{O}}(\frac{\kappa^2}{\varepsilon^2} + (1 + \frac{1}{K})^{3/2} \frac{\kappa^3}{S^{1/2}\varepsilon^3})$
	$\tilde{\mathcal{O}}(\frac{\kappa}{\varepsilon^2}(\frac{\kappa}{K} + c) + (1 + \frac{\kappa}{cK})^3 \frac{c\kappa^2}{S^{1/2}\varepsilon^3})^\sharp$

† This entry relies on the extra condition that $f(\mathbf{x}, \cdot)$ is μ-strongly concave $\forall \mathbf{x}$, which is stronger than Assumption 3.
$^\sharp$ This result requires the additional Assumption 5. Note that $c := \max\{1, \delta/\mu\} \leq \kappa$.

Table 1: Communication costs of algorithms for minimax problems in cross-device FL.

Experiments

To demonstrate the efficiency of the proposed algorithm, we conduct experiments on three tasks: (i) AUC maximization, (ii) robust adversarial neural network training, and (iii) GAN training. We compare the proposed three versions of CDMA with Parallel SGDA (Lin, Jin, and Jordan 2020), and the cross-device versions of Extra Step Local SGD (Beznosikov, Samokhin, and Gasnikov 2020), Local SGDA+ (Deng and Mahdavi 2021), Catalyst-Scaffold-S (Hou et al. 2021), CODA+, and CODASCA (Yuan et al. 2021). We note that Local SGDA+, Extra Step Local SGD, Catalyst-Scaffold-S, CODA+, and CODASCA are state-of-the-art federated minimax algorithms which rely on the full aggregation strategy and thus only apply to the cross-silo setting. Here, we use their cross-device variants by only involving a subset of clients in each round, which are actually *not* theoretically guaranteed.

In our experimental setting, there are 500 clients in total and the client distribution $\mathcal{D}$ is the uniform distribution. Since communication is often the major bottleneck in cross-device FL and clients generally have much slower upload than download bandwidth (Kairouz et al. 2019), we compare the algorithms given the same amount of data transferred from clients to the server. To ensure that each algorithm has the same amount of communication per round, for CDMA-NC, Parallel SGDA, Local SGDA+, Extra Step Local SGD and CODA+, which only require to send parameters from clients to the server, the server selects a random client subset of size $\hat{S} = 16$ in each round. For the rest algorithms, in which clients send both parameters and gradient estimates to the server, the size of the randomly selected client subset is fixed to $\hat{S} = 8$ in each round. Moreover, to simulate the low client availability, we set the size of clients that successfully respond to the server as $S_t = \lceil p_t \hat{S} \rceil$, where the response probability p_t is uniformly sampled from $[0.5, 1]$. Throughout the experiments, we use the same random seed for all algorithms for reproducibility. We do grid-search on hyperparameters for the algorithms and select the best hyperparameters. All experiments were implemented in Pytorch and run on 4 workstations, each with 2 Intel E5-2680 v4 CPUs (28 cores), 4 NVIDIA 2080Ti GPUs, and 378GB memory.[4]

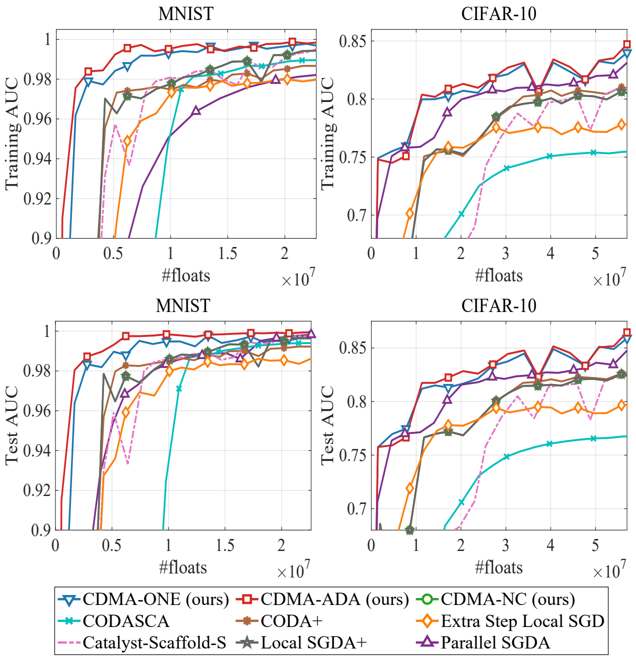

Figure 1: Results on the AUC maximization task (top: training results, bottom: test results). The left and right columns correspond to MNIST and CIFAR-10 datasets, respectively.

AUC Maximization

In the first experiment, we consider the ℓ_2-relaxed AUC maximization problem in the minimax formulation (Ying, Wen, and Lyu 2016; Liu et al. 2020a). This formulation is widely used in binary classification tasks, especially on imbalanced data. We use two datasets: MNIST (LeCun et al. 1998) and CIFAR-10 (Krizhevsky, Hinton et al. 2009). To make the data heterogeneous, the training data is first sorted according to the original class label and then equally partitioned into 500 clients so that all data points on one client are from the same class. Each algorithm runs exactly one local epoch ($K = 12$ local steps for MNIST and $K = 10$ for CIFAR-10) with the minibatch size $B = 10$ on the selected clients in each round.

We report the AUC value on the training and test datasets versus the amount of communication (the total number of FLOATs received by the server) in Figure 1. We observe that CDMA-ADA and CDMA-ONE are superior to other algorithms on both datasets. In addition, among the theoretically-guaranteed algorithms, CDMA-ADA achieves the best performance, CDMA-ONE is only inferior to CDMA-ADA, and CDMA-NC and Parallel SGDA have comparable performance. This corroborates our theory and demonstrates the efficiency of the SWIER aggregation mechanism as well as the stocahstic recursive momentum.

---

[4] Source code: https://github.com/xjiajiahao/federated-minimax

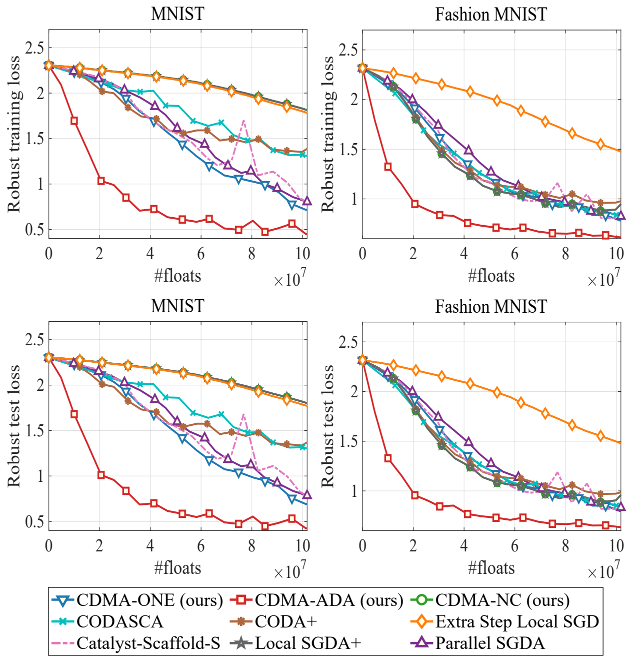

Figure 2: Results on the robust neural network training task. The top (resp., bottom) row depicts the training (resp., test) results. The left (resp., right) column corresponds to the MNIST (resp., Fashion MNIST) dataset.

Robust Adversarial Network Training

In the second experiment, we focus on training a classification model that is robust against adversarial noise (Deng and Mahdavi 2021). We use MNIST and Fashion MNIST (Xiao, Rasul, and Vollgraf 2017) datasets in this experiment. The data partitioning scheme is the same as the AUC experiment and each pixel value of the images is rescaled to $[-1, 1]$. The performance measure is the robust loss, which is the loss function value of the classification model on the optimally perturbed dataset. We defer details of the model and the problem formulation to the long version due to the space limit. Similar to the previous experiment, each algorithm runs with the minibatch size $B = 10$ and the number of local steps $K = 12$. Note that each selected client makes one pass of its local dataset in each round. The experimental results on the training and test datasets are shown in Figure 2. It shows that CDMA-ADA achieves the best performance and CDMA-ONE outperforms the other two theoretically-guaranteed algorithms CDMA-NC and Parallel SGDA.

GAN Training

In the last experiment, we test the performance of our algorithms for training GANs on MNIST, Fashion MNIST, and CelebA (Liu et al. 2015) datasets. We measure the performance of the compared algorithms with the Inception Score (IS) (Chavdarova et al. 2019) on MNIST and Fashion MNIST, and the more complicated Fréchet Inception Distance (FID) (Heusel et al. 2017) on the harder dataset CelebA. The experimental setting in more details is deferred to the long version of this paper.

Algorithm	IS on MNIST	IS on F-MNIST	FID on CelebA
CDMA-ADA (ours)	**8.50**	**8.75**	**35.84**
CDMA-ONE (ours)	8.34	8.57	37.55
CDMA-NC (ours)	8.31	7.58	44.71
CODASCA	7.85	8.18	38.33
CODA+	8.33	8.23	47.61
Catalyst-Scaffold-S	8.13	8.38	39.37
Local SGDA+	8.29	7.61	79.24
Extra Step Local SGD	7.50	7.61	59.81
Parallel SGDA	6.08	5.74	222.65

Table 2: The best IS obtained by the compared algorithms on MNIST and Fashion MNIST (F-MNIST), and the best FID obtained on the CelebA dataset. Larger IS is better, while smaller FID is better.

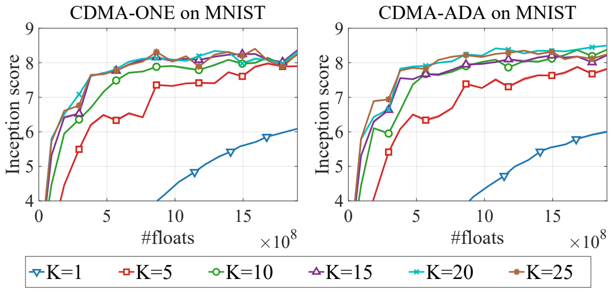

Figure 3: Results of CDMA-ONE (left) and CDMA-ADA (right) with different K for GAN training on MNIST.

Table 2 presents the best IS (resp., FID) obtained by the algorithms on MNIST and Fashion MNIST (resp., CelebA) given a fixed amount of communication for each dataset. The results show that CDMA-ADA (resp., CDMA-ONE) achieves the best (resp., second best) performance among the compared algorithms.

In order to validate the effect of local steps, we also conduct empirical studies on the three versions of CDMA with different numbers of local steps $K \in \{1, 5, 10, 15, 20, 25\}$ on MNIST and Fashion MNIST. Figure 3 reports the results of CDMA-ONE and CDMA-ADA on MNIST. We can see that both CDMA-ONE and CDMA-ADA with multiple local steps attain significantly higher inception scores than with only one local step, and achieve the best performance when $K \in \{15, 20, 25\}$ and $K = 20$, respectively.

Conclusion and Future Work

We propose the first practical algorithm for general minimax problems in the cross-device FL setting. Equipped with the SIWER mechanism and the global correction, our algorithm is resilient to the low client availability and alleviates the impact of slow network connections. Theoretical analyses and experimental results show the efficiency of our algorithm. Directions for future work include the extension of CDMA to constrained problems and the study of lower bounds for minimax problems in the cross-device FL setting.

Acknowledgments

This work is supported by National Key Research and Development Program of China under Grant 2020AAA0107400, Zhejiang Provincial Natural Science Foundation of China under Grant No. LZ18F020002, and National Natural Science Foundation of China (Grant No: 62206248).

References

Acar, D. A. E.; Zhao, Y.; Matas, R.; Mattina, M.; Whatmough, P.; and Saligrama, V. 2021. Federated Learning Based on Dynamic Regularization. In *International Conference on Learning Representations*.

Arjevani, Y.; and Shamir, O. 2015. Communication Complexity of Distributed Convex Learning and Optimization. *Advances in Neural Information Processing Systems*, 28: 1756–1764.

Beznosikov, A.; Samokhin, V.; and Gasnikov, A. 2020. Distributed Saddle-Point Problems: Lower Bounds, Optimal Algorithms and Federated GANs. *arXiv preprint arXiv:2010.13112*.

Brown, G. W. 1951. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1): 374–376.

Chavdarova, T.; Gidel, G.; Fleuret, F.; and Lacoste-Julien, S. 2019. Reducing noise in GAN training with variance reduced extragradient. In *Advances in Neural Information Processing Systems*, 391–401.

Cutkosky, A.; and Orabona, F. 2019. Momentum-based variance reduction in non-convex SGD. In *Advances in Neural Information Processing Systems*, 15236–15245.

Deng, Y.; Kamani, M. M.; and Mahdavi, M. 2020. Distributionally robust federated averaging. *Advances in Neural Information Processing Systems*, 33: 15111–15122.

Deng, Y.; and Mahdavi, M. 2021. Local stochastic gradient descent ascent: Convergence analysis and communication efficiency. In *International Conference on Artificial Intelligence and Statistics*, 1387–1395. PMLR.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Guo, Z.; Liu, M.; Yuan, Z.; Shen, L.; Liu, W.; and Yang, T. 2020. Communication-Efficient Distributed Stochastic AUC Maximization with Deep Neural Networks. In *International Conference on Machine Learning*, 3864–3874. PMLR.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Hou, C.; Thekumparampil, K. K.; Fanti, G.; and Oh, S. 2021. Efficient Algorithms for Federated Saddle Point Optimization. *arXiv preprint arXiv:2102.06333*.

Johnson, R.; and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, 315–323.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*.

Karimireddy, S. P.; Jaggi, M.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S. U.; and Suresh, A. T. 2021. Breaking the centralized barrier for cross-device federated learning. *Advances in Neural Information Processing Systems*, 34.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 5132–5143. PMLR.

Khaled, A.; Mishchenko, K.; and Richtárik, P. 2020. Tighter theory for local SGD on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, 4519–4529. PMLR.

Korpelevich, G. 1976. The extragradient method for finding saddle points and other problems. *Matecon*, 12: 747–756.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Citeseer.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*.

Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2020. On the Convergence of FedAvg on Non-IID Data. In *International Conference on Learning Representations*.

Lin, T.; Jin, C.; and Jordan, M. 2020. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, 6083–6093. PMLR.

Liu, M.; Yuan, Z.; Ying, Y.; and Yang, T. 2020a. Stochastic AUC Maximization with Deep Neural Networks. In *International Conference on Learning Representations*.

Liu, M.; Zhang, W.; Mroueh, Y.; Cui, X.; Ross, J.; Yang, T.; and Das, P. 2020b. A decentralized parallel algorithm for training generative adversarial nets. *Advances in Neural Information Processing Systems*, 33: 11056–11070.

Liu, Q.; and Wang, J. 2015. A Projection Neural Network for Constrained Quadratic Minimax Optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 26(11): 2891–2900.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Luo, L.; Ye, H.; Huang, Z.; and Zhang, T. 2020. Stochastic Recursive Gradient Descent Ascent for Stochastic Nonconvex-Strongly-Concave Minimax Problems. *Advances in Neural Information Processing Systems*, 33.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Mateos-Núnez, D.; and Cortés, J. 2015. Distributed subgradient methods for saddle-point problems. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 5462–5467. IEEE.

Meng, S. Y.; Vaswani, S.; Laradji, I. H.; Schmidt, M.; and Lacoste-Julien, S. 2020. Fast and furious convergence: Stochastic second order methods under interpolation. In *International Conference on Artificial Intelligence and Statistics*, 1375–1386. PMLR.

Mohri, M.; Sivek, G.; and Suresh, A. T. 2019. Agnostic Federated Learning. In *International Conference on Machine Learning*, 4615–4625.

Nedić, A.; and Ozdaglar, A. 2009. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1): 205–228.

Nemirovski, A. 2004. Prox-method with rate of convergence O (1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1): 229–251.

Nouiehed, M.; Sanjabi, M.; Huang, T.; Lee, J. D.; and Razaviyayn, M. 2019. Solving a class of non-convex min-max games using iterative first order methods. In *Advances in Neural Information Processing Systems*, 14934–14942.

Palaniappan, B.; and Bach, F. 2016. Stochastic variance reduction methods for saddle-point problems. In *Advances in Neural Information Processing Systems*, 1416–1424.

Qiu, S.; Yang, Z.; Wei, X.; Ye, J.; and Wang, Z. 2020. Single-Timescale Stochastic Nonconvex-Concave Optimization for Smooth Nonlinear TD Learning. *arXiv preprint arXiv:2008.10103*.

Rasouli, M.; Sun, T.; and Rajagopal, R. 2020. Fedgan: Federated generative adversarial networks for distributed data. *arXiv preprint arXiv:2006.07228*.

Reddi, S. J.; Konečnỳ, J.; Richtárik, P.; Póczós, B.; and Smola, A. 2016. AIDE: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*.

Reisizadeh, A.; Farnia, F.; Pedarsani, R.; and Jadbabaie, A. 2020. Robust federated learning: The case of affine distribution shifts. *Advances in Neural Information Processing Systems*, 33: 21554–21565.

Rogozin, A.; Beznosikov, A.; Dvinskikh, D.; Kovalev, D.; Dvurechensky, P.; and Gasnikov, A. 2021. Decentralized distributed optimization for saddle point problems. *arXiv preprint arXiv:2102.07758*.

Sanjabi, M.; Razaviyayn, M.; and Lee, J. D. 2018. Solving non-convex non-concave min-max games under polyak-łojasiewicz condition. *arXiv preprint arXiv:1812.02878*.

Shamir, O.; Srebro, N.; and Zhang, T. 2014. Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, 1000–1008. PMLR.

Sharma, P.; Panda, R.; Joshi, G.; and Varshney, P. 2022. Federated minimax optimization: Improved convergence analyses and algorithms. In *International Conference on Machine Learning*, 19683–19730. PMLR.

Srivastava, K.; Nedić, A.; and Stipanović, D. 2011. Distributed min-max optimization in networks. In *2011 17th International Conference on Digital Signal Processing (DSP)*, 1–8. IEEE.

Sun, Z.; and Wei, E. 2022. A Communication-efficient Algorithm with Linear Convergence for Federated Minimax Learning. *arXiv preprint arXiv:2206.01132*.

Tran-Dinh, Q.; Pham, N. H.; Phan, D. T.; and Nguyen, L. M. 2021. A hybrid stochastic optimization framework for composite nonconvex optimization. *Mathematical Programming*, 1–67.

Woodworth, B. E.; Patel, K. K.; and Srebro, N. 2020. Minibatch vs Local SGD for Heterogeneous Distributed Learning. *Advances in Neural Information Processing Systems*, 33: 6281–6292.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, J.; Shen, Z.; Zhang, C.; Wang, B.; and Qian, H. 2020. Efficient projection-free online methods with stochastic recursive gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 6446–6453.

Xu, T.; Wang, Z.; Liang, Y.; and Poor, H. V. 2020. Enhanced first and zeroth order variance reduced algorithms for min-max optimization. *arXiv preprint arXiv:2006.09361*.

Yang, J.; Kiyavash, N.; and He, N. 2020. Global convergence and variance-reduced optimization for a class of nonconvex-nonconcave minimax problems. *arXiv preprint arXiv:2002.09621*.

Ying, Y.; Wen, L.; and Lyu, S. 2016. Stochastic online AUC maximization. In *Advances in neural information processing systems*, 451–459.

Yuan, Z.; Guo, Z.; Xu, Y.; Ying, Y.; and Yang, T. 2021. Federated deep AUC maximization for hetergeneous data with a constant communication complexity. In *International Conference on Machine Learning*, 12219–12229. PMLR.

Zhang, M.; Shen, Z.; Mokhtari, A.; Hassani, H.; and Karbasi, A. 2020. One sample stochastic frank-wolfe. In *International Conference on Artificial Intelligence and Statistics*, 4012–4023. PMLR.