# Local Stochastic Gradient Descent Ascent: Convergence Analysis and Communication Efficiency

**Yuyang Deng**
The Pennsylvania State University

**Mehrdad Mahdavi**
The Pennsylvania State University

## Abstract

Local SGD is a promising approach to overcome the communication overhead in distributed learning by reducing the synchronization frequency among worker nodes. Despite the recent theoretical advances of local SGD in empirical risk minimization, the efficiency of its counterpart in minimax optimization remains unexplored. Motivated by large scale minimax learning problems, such as adversarial robust learning and training generative adversarial networks (GANs), we propose local Stochastic Gradient Descent Ascent (local SGDA), where the primal and dual variables can be trained locally and averaged periodically to significantly reduce the number of communications. We show that local SGDA can provably optimize distributed minimax problems in both homogeneous and heterogeneous data with reduced number of communications and establish convergence rates under strongly-convex-strongly-concave and nonconvex-strongly-concave settings. In addition, we propose a novel variant local SGDA+, to solve nonconvex-nonconcave problems. We give corroborating empirical evidence on different distributed minimax problems.

## 1 Introduction

We study minimax optimization problems of the form

$$\min_{\boldsymbol{x}\in\mathbb{R}^{d_x}} \max_{\boldsymbol{y}\in\mathbb{R}^{d_y}} \left\{ F(\boldsymbol{x},\boldsymbol{y}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x},\boldsymbol{y}) \right\}, \quad (1)$$

where data are distributed across $n$ nodes so that each node $i$ will have its own objective function $f_i(\cdot,\cdot)$. The local objective function is defined as $f_i(\cdot,\cdot) = \mathbb{E}_{\xi\sim\mathcal{D}_i}[\ell(\cdot,\cdot;\xi)]$, where $\mathcal{D}_i$ is the local data distribution of $i$th client and $\ell$ is the loss function. Numerous machine learning problems fall in this category. A canonical instance is adversarially robust learning. Consider the following robust linear regression $\min_{\boldsymbol{x}\in\mathbb{R}^{d_x}} \max_{\boldsymbol{y}\in\mathbb{R}^{d_y}} \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{x}^\top(\boldsymbol{a}_i+\boldsymbol{y});\boldsymbol{b}_i) + \frac{\lambda_x}{2}\|\boldsymbol{x}\|^2 - \frac{\lambda_y}{2}\|\boldsymbol{y}\|^2$, where $\{(\boldsymbol{a}_i,\boldsymbol{b}_i)\}_{i=1}^n$ are input pairs of training data. We wish to learn a predictor $\boldsymbol{x}$ that is robust to small perturbation $\boldsymbol{y}$. Another popular minimax application is Generative Adversarial Network (GAN) [8], which can be formulated as: $\min_{\boldsymbol{x}\in\mathbb{R}^{d_x}} \max_{\boldsymbol{y}\in\mathbb{R}^{d_y}} \mathbb{E}_{\boldsymbol{a}\sim\mathcal{D}_{real}}[\ell(D_{\boldsymbol{y}}(\boldsymbol{a}))] + \mathbb{E}_{\boldsymbol{a}\sim\mathcal{D}_{\boldsymbol{x}}}[\ell(1-D_{\boldsymbol{y}}(\boldsymbol{a}))]$, where $\boldsymbol{x}$ is the parameter of the generator network $\mathcal{D}_{\boldsymbol{x}}$ and $\boldsymbol{y}$ is the parameter of the discriminator network $D_{\boldsymbol{y}}$.

The centrality of these applications in machine learning motivates considerable interest in efficiently solving minimax optimization problems. Among all popular algorithms, primal-dual stochastic gradient algorithms are definitely the most popular ones [38, 37, 48]. The most classic algorithm in this category is stochastic gradient descent ascent (SGDA), which has been proven to be an effective algorithm for minimax optimization both empirically and theoretically [29]. However, in practice, due to the huge volume of data, or to protect the privacy of user data (e.g., federated learning scenario [18, 19]), a distributed algorithm while lowering the communication cost is preferred and is the focus of this paper. A conventional distributed approach to solve (1) is *parameter server* model, where every client (user) sends its local stochastic gradient to a central node, and the central node performs stochastic gradient descent procedure on primal and dual variables by aggregating local stochastic gradients. Unfortunately, this approach causes heavy communication outage, which has been reported to be the main bottleneck slowing down the distributed optimization [1, 30, 44, 55].

A notable research effort to reduce the commutation complexity under a computation budget is to employ local SGD with periodic averaging [34, 46]. In local SGD, the idea is to perform multiple local up-

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. 

| Assumption | Setting | Results | Comm. Rounds | Convergence Rate |
|---|---|---|---|---|
| Strongly-Convex-Strongly-Concave | Homogeneous | Theorem 4.1 | $\tilde{\Omega}(n)$ | $\tilde{O}\left(\frac{\kappa^2\sigma^2}{\mu^2 nT}\right)$ |
| | Heterogeneous | Theorem 4.2 | $\Omega\left(\sqrt{nT}\right)$ | $O\left(\frac{\kappa^2\left(\Delta_x+\Delta_y+\sigma^2\right)}{\mu nT}\right)$ |
| Nonconvex-Strongly-Concave | Homogeneous | Theorem 5.1 | $\Omega\left(n^{1/3}T^{2/3}\right)$ | $O\left(\frac{L^2\sigma^2}{(nT)^{1/3}}\right)$ |
| | Heterogeneous | Theorem 5.1 | $\Omega\left(n^{1/3}T^{2/3}\right)$ | $O\left(\frac{L^2\sigma^2}{(nT)^{1/3}}+\frac{L^2\zeta_x}{T^{2/3}}+\frac{L^2\zeta_y}{n^{2/3}T^{1/3}}\right)$ |
| Nonconvex-PL condition | Homogeneous | Theorem 6.1 | $\Omega\left(T^{2/3}\right)$ | $O\left(\frac{\beta\sigma^2}{(nT)^{1/3}}\right)$ |
| | Heterogeneous | Theorem 6.1 | $\Omega\left(T^{2/3}\right)$ | $O\left(\frac{\beta\sigma^2}{(nT)^{1/3}}+\frac{\kappa^2L^2\zeta_y}{n^{2/3}T^{1/3}}+\frac{\kappa^2L^2\zeta_x}{n^{2/3}T}\right)$ |
| Nonconvex-One-Point-Concave | Homogeneous | Theorem 6.2 | $\Omega\left(T^{2/3}\right)$ | $O\left(\frac{L\sigma^2}{T^{1/6}}\right)$ |
| | Heterogeneous | Theorem 6.2 | $\Omega\left(T^{2/3}\right)$ | $O\left(\frac{L\sigma^2}{T^{1/6}}+\frac{L^2\zeta_x}{n^{1/3}T}+\frac{L^2\zeta_y}{(nT)^{1/3}}\right)$ |

Table 1: A summary of our results under different settings. We use $\tilde{O}(\cdot)$ and $\tilde{\Omega}(\cdot)$ to hide logarithmic term. $\Delta_x$ and $\Delta_y$ are heterogeneity at the optimum (see Definition 2). $\zeta_x$ and $\zeta_y$ denote gradient dissimilarity (see Definition 4).

dates, wherein clients update their own local models via SGD for multiple iterations, and the models of the different clients are averaged periodically. While this algorithm introduces additional noise due to local updates over fully synchronous SGD, it is shown that by careful choice of learning rate, local SGD can achieve same asymptotic performance as synchronous SGD, while benefiting from reduced communication rounds [46, 53, 49, 9, 15, 27]. Motivated by the success of local SGD and a key observation that in some minimax applications (e.g., aforementioned robust linear regression and GANs), the primal and dual variables can be trained in a distributed manner and locally, we extend local SGD to tackle minimax learning problems and propose *local stochastic gradient descent ascent* (local SGDA) algorithm. In local SGDA, local nodes will optimize their own version of primal and dual variables for multiple steps, and then they synchronize and do model averaging via central server. However, despite it being an extremely simple algorithm, and the thorough understanding of local SGD on minimization problem, local SGDA, as its counter-part in minimax problem, still lacks theoretical foundations. Thus, a natural question that arises is: ***Does local SGDA provably optimize distributed minimax problems too?***

We answer above question in the affirmative, by establishing the convergence rate of local SGDA in both **homogeneous data setting**, where local functions in (1) have the same distribution (IID), i.e., $\mathcal{D}_1 = \ldots = \mathcal{D}_n = \mathcal{D}$, and **heterogeneous data setting**, where local functions are not necessarily realized by the same distribution (non-IID). Our main contributions can be summarized as follows. We are the first to show that local SGDA provably optimizes the distributed minimax problem with communication efficiency, on both homogeneous and heterogeneous data. For strongly-convex-strongly-concave setting, we obtain the convergence rate of $\tilde{O}\left(\frac{1}{nT}\right)$ with $\tilde{\Omega}(n)$ communication rounds in homogeneous local functions setting, and $O\left(\frac{\Delta_x+\Delta_y}{nT}\right)$ with $\Omega(\sqrt{nT})$ communication rounds in heterogenous setting, where $\Delta_x + \Delta_y$ is the quantity reflecting heterogeneity. It recovers the same asymptotic rate and communication rounds as local SGD in the smooth strongly-convex minimization problem [16, 52, 51], up to a constant factor. For nonconvex-strongly-concave problem, we get the rate $O\left(\frac{1}{(nT)^{1/3}}\right)$ with $\Omega\left(T^{2/3}\right)$ communication rounds, under both data allocation settings. In addition, in order to efficiently solve the nonconvex-nonconcave minimax optimization problems, we propose a variant of local SGDA, dubbed as local SGDA+, a single loop algorithm to solve nonconvex-nonconcave problems. We establish its convergence rate on two classes of functions, which are nonconvex in $\boldsymbol{x}$ but satisfies Polyak-Łojasiewicz (PL) condition in $\boldsymbol{y}$ [13], and nonconvex in $\boldsymbol{x}$ but one-point-concave in $\boldsymbol{y}$. We summarize the obtained rates for different settings in Table 1.

## 2 Prior Art

**Single Machine Minimax Optimization.** The history of minimax optimization dates back to Brown [3], where he proposed a bilinear form minimax problem. Korpelevich [20] then proposed the extra gradient (EG) method to solve this bilinear problem. Following their path, Nemirovski [37], Nesterov [38] and Tseng [48] studied the general smooth convex-concave minimax problem, and proposed algorithms which achieve the same asymptotic rate $O(1/T)$. Du and Hu [6] prove the linear convergence of primal-dual gradient method on a class of convex-concave functions. The other popular algorithm for convex-concave optimization is Optimistic Gradient Descent Ascent (OGDA), which is widely studied and has many applications in machine learning [4, 25, 36]. For strongly-convex-concave

setting, Thekumprampil et al [47] proposed an algorithm combing Nesterov accelerated gradient descent and Mirror-Prox, which achieves near optimal rate $\tilde{O}(1/T^2)$. For strongly-convex-strongly-concave setting, Lin et al [28] leveraged the idea of accelerated gradient descent, and gave a nearly optimal minimax algorithm, which matches the lower bound given in [40]. Some literature [29, 39, 41, 47] also conduct trials on nonconvex-concave minimax optimization, and among them the most related work to us is [29], where they study the single machine SGDA, under nonconvex-(strongly)-concave case. Recently, due to the raise of GANs [8], a vast amount of work is devoted to nonconvex-nonconcave optimization [7, 31, 32].

**Distributed Minimax Optimization.** A few recent studies are devoted to decentralized minimax optimization. Srivastava et al [45] proposed a decentralized algorithm to solve the convex-concave saddle point problem over a network. Mateos and Cortes [33] proposed a subgradient method and prove the convergence under convex-concave case. Liu et al [32] analyzed the convergence of networked optimistic stochastic gradient descent ascent (OSGDA) on nonconvex-nonconcave setting. [43] studied a variant of local SGDA, and provided the convergence analysis on PL-PL and nonconvex-PL objective. We note that [2] also studies the convergence of local SGDA on strongly-convex-strongly-concave setting, but their analysis is not as tight as ours. Recently, federated adversarial training [43] and FedGAN [42] are proposed to solve large-scale and privacy-preserving minimax problem, which can be seen as application instances of our work.

**Local SGD.** Communication efficiency has been studied extensively in distributed SGD. The most related idea to this paper is local SGD or FedAvg [34]. FedAvg is firstly proposed by Mcmahan et al [34] to alleviate communication bottleneck in the distributed machine learning. Stich [46] was the first to prove that local SGD achieves $O\left(1/T\right)$ convergence rate with only $O(\sqrt{T})$ communication rounds on IID data for smooth strongly-convex loss functions. Haddadpour et al [9] analyzed the convergence of local SGD on nonconvex (PL condition) function, and proposed an adaptive synchronization scheme. [16] gave the tighter bound of local SGD, which directly reduces the $O(\sqrt{T})$ communication rounds in [46] to $O(n)$, under smooth strongly-convex setting. Recently, Yuan and Ma [54] proposed the first accelerated local SGD, which further reduced the communication rounds to $O(n^{1/3})$. [10] gave the analysis of local GD and SGD on smooth nonconvex functions in non-IID setting. Li et al [22] analyzed the convergence of FedAvg under non-IID data for strongly convex functions. [52, 51] investigated the difference between local SGD and mini-batch SGD, in both homogeneous and heterogeneous data settings.

**Algorithm 1:** `Local SGDA`

**input:** Synchronization gap $\tau$, Communication rounds $S$, Number of iterations $T = S\tau$, Initial local models $\boldsymbol{x}_i^{(0)}$, $\boldsymbol{y}_i^{(0)}$ for $i \in [n]$.

**parallel for** $i = 1, ..., n$ **do**
  **for** $s = 0, \ldots, S-1$ **do**
    all nodes send their local model $\boldsymbol{x}_i^{(s\tau)}$ and $\boldsymbol{y}_i^{(s\tau)}$ to server.
    $\boldsymbol{x}^{(s\tau)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(s\tau)}$
    $\boldsymbol{y}^{(s\tau)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i^{(s\tau)}$
    server sends $\boldsymbol{x}^{(s\tau)}$, $\boldsymbol{y}^{(s\tau)}$ to all nodes;
    each client initializes its local models: $\boldsymbol{x}_i^{(s\tau)} = \boldsymbol{x}^{(s\tau)}$ and $\boldsymbol{y}_i^{(s\tau)} = \boldsymbol{y}^{(s\tau)}$.
    **for** $t = s\tau, \ldots, (s+1)\tau - 1$ **do**
      sample a minibatch $\xi_i^t$ from local data
      $\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} - \eta_x \nabla_x f_i\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{y}_i^{(t)}; \xi_i^t\right)$
      $\boldsymbol{y}_i^{(t+1)} = \boldsymbol{y}_i^{(t)} + \eta_y \nabla_y f_i\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{y}_i^{(t)}; \xi_i^t\right)$
    **end**
  **end**
**end**

## 3 Local SGDA

In this section we formally introduce local SGDA algorithm for solving distributed minimax problems. The proposed algorithm can be viewed as a variant of SGDA, which is one of the most popular primal-dual stochastic gradient algorithm to solve centralized minimax optimization problems. Specifically, for solving the optimization problem in (1), at $t$th iteration, SGDA performs the following updates on primal and dual variables:

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \eta_x \nabla_x F\left(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}; \xi^t\right)$$
$$\boldsymbol{y}^{(t+1)} = \boldsymbol{y}^{(t)} + \eta_y \nabla_y F\left(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}; \xi^t\right),$$

where $\xi^t$ is minibatch sampled at $t$th iteration to compute stochastic gradient, and $\eta_x$ and $\eta_y$ are learning rates.

The key difficulty of deploying SGDA in a distributed setting stems from the fact that after $t$th updating, server needs to communicate global models $\boldsymbol{x}^{(t)}$ and $\boldsymbol{y}^{(t)}$ to all nodes, so clients can locally evaluate the gradient on $\boldsymbol{x}^{(t)}$ and $\boldsymbol{y}^{(t)}$. Meanwhile local users should send their local gradients back to the server for aggregation/averaging. This suffers from heavy communication cost and could hinder the scalability of the algorithm

as communication is known to be a major bottleneck that slows down the training process [1, 30, 44, 55].

As mentioned earlier, to mitigate the communication bottleneck, a popular idea is to update models locally via SGD, and then average them periodically [34, 46]. Motivated by this, we advocate a local primal-dual algorithm for minimax optimization as detailed in Algorithm 1. To formally present the steps of proposed Local SGDA algorithm, consider $S$ as the rounds of communication between server and clients, and $\tau$ as the number of local updates performed by clients between two consecutive communication rounds. The algorithm proceeds for $T = S\tau$ iterations and at $t$th local iteration, the $i$th node locally performs the SGDA on its own local primal and dual variables

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} - \eta_x \nabla_x f_i\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{y}_i^{(t)}; \xi_i^t\right),$$
$$\boldsymbol{y}_i^{(t+1)} = \boldsymbol{y}_i^{(t)} + \eta_y \nabla_y f_i\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{y}_i^{(t)}; \xi_i^t\right),$$

for $\tau$ iterations, where $\xi_i^t$ is the minibatch sampled by $i$th client from its local data to compute local stochastic gradient at iteration $t$. At $s$th synchronization round, the server aggregates local models $\boldsymbol{x}_i^{(s\tau)}$ and $\boldsymbol{y}_i^{(s\tau)}$, to perform the averaging: $\boldsymbol{x}^{(s\tau)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(s\tau)}$ and $\boldsymbol{y}^{(s\tau)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i^{(s\tau)}$. Then, the server sends the averaged models back to local nodes. We note that compared to fully synchronous distributed SGDA, which requires $T$ communication round, in local SGDA we only require $T/\tau$ communications. Despite its simplicity, we are not aware of any prior result that establishes the convergence rate of local methods in minimax setting. In the following sections, we show that the proposed algorithm enjoys a fast convergence rate while significantly reducing the communication rounds by properly choosing the number of local updates $\tau$.

## 4 Strongly-Convex-Strongly-Concave Case

In this section we will present the convergence analysis of local SGDA for strongly-convex-strongly-concave functions, under both homogeneous and heterogeneous data settings. In the strongly-convex-strongly-concave minimax problem, our goal is to find the *saddle point* of global objective, as defined below:

**Definition 1.** *The tuple $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is said to be saddle point of convex-concave function $F(\boldsymbol{x}, \boldsymbol{y})$ if $F(\boldsymbol{x}^*, \boldsymbol{y}) \leq F(\boldsymbol{x}^*, \boldsymbol{y}^*) \leq F(\boldsymbol{x}^*, \boldsymbol{y}), \forall \boldsymbol{x} \in \mathbb{R}^{d_x}, \boldsymbol{y} \in \mathbb{R}^{d_y}$.*

To facilitate our analysis, we make the following standard assumptions on objective function and noise of stochastic gradients.

**Assumption 1** (Strong Convexity)**.** *$f_i(\boldsymbol{x}, \boldsymbol{y})$ is strongly convex in $\boldsymbol{x}$, which implies there exists a $\mu > 0$ such that $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{d_x}, \boldsymbol{y} \in \mathbb{R}^{d_y}$ it holds that $f_i(\boldsymbol{x}, \boldsymbol{y}) \geq f_i(\boldsymbol{x}', \boldsymbol{y}) + \langle \nabla_x f_i(\boldsymbol{x}', \boldsymbol{y}), \boldsymbol{x}' - \boldsymbol{x}\rangle + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|^2$.*

**Assumption 2** (Strong Concavity)**.** *$f_i(\boldsymbol{x}, \boldsymbol{y})$ is strongly concave in $\boldsymbol{y}$, which implies there exists a $\mu > 0$ such that $\forall \boldsymbol{x} \in \mathbb{R}^{d_x}, \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^{d_y}$ it holds that $f_i(\boldsymbol{x}, \boldsymbol{y}) \leq f_i(\boldsymbol{x}, \boldsymbol{y}') + \langle \nabla_y f_i(\boldsymbol{x}, \boldsymbol{y}'), \boldsymbol{y}' - \boldsymbol{y}\rangle - \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{y}'\|^2$.*

**Assumption 3** (Smoothness)**.** *There exists a $L > 0$ such that $\forall i \in [n], \|\nabla f_i(\boldsymbol{x}_1, \boldsymbol{y}_1) - \nabla f_i(\boldsymbol{x}_2, \boldsymbol{y}_2)\| \leq L\|(\boldsymbol{x}_1, \boldsymbol{y}_1) - (\boldsymbol{x}_2, \boldsymbol{y}_2)\|$, $\forall \boldsymbol{x} \in \mathbb{R}^{d_x}, \boldsymbol{y} \in \mathbb{R}^{d_y}$.*

**Assumption 4** (Bounded Variance)**.** *The variance of stochastic gradients computed at each local function is bounded, i.e., $\forall i \in [n], \mathbb{E}[\|\nabla_x f_i(\boldsymbol{x}, \boldsymbol{y}; \xi) - \nabla_x f_i(\boldsymbol{x}, \boldsymbol{y})\|^2] \leq \sigma^2$ and $\mathbb{E}[\|\nabla_y f_i(\boldsymbol{x}, \boldsymbol{y}; \xi) - \nabla_y f_i(\boldsymbol{x}, \boldsymbol{y})\|^2] \leq \sigma^2$.*

**Main techniques.** In our analysis, due to infrequent synchronization, the key is to bound the deviation among local and global models as defined below

$$\delta_{\boldsymbol{x}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \left\|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}^{(t)}\right\|^2, \delta_{\boldsymbol{y}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \left\|\boldsymbol{y}_i^{(t)} - \boldsymbol{y}^{(t)}\right\|^2, \tag{2}$$

where $\boldsymbol{x}^{(t)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(t)}$ and $\boldsymbol{y}^{(t)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i^{(t)}$ are (virtual) primal and dual global averages at iteration $t$, respectively. We note that virtual averages are introduced for analysis purposes and only computed at synchronization rounds.

Despite minimization problems, where we already have a solid theory to bound the deviation between global and local models [46, 9, 10, 22, 16, 14, 23, 52, 51], none of these guarantees apply to minimax problem, due to the unstable nature of primal-dual optimization. Hence the key step in our analysis is to develop a relatively tight bound for quantities introduced in (2). In the homogeneous setting, we show that under the dynamic of primal-dual algorithm, and smooth strongly convex assumption, the deviation $\delta_{\boldsymbol{x}}^{(t)} + \delta_{\boldsymbol{y}}^{(t)}$ can decrease as the rate of $O(\tau(1+(L-\mu)\eta)^{2\tau}\eta^2\sigma^2)$. By properly choosing $\tau$ and $\eta$, we can recover the rate $O(\tau\eta^2\sigma^2)$, which matches with the existing tightest deviation bounds of local SGD [16, 52]. In the heterogeneous setting, we prove that, by carefully controlling the step size, we can develop the deviation bound that depends on distance between the current iterate and the saddle point $(\boldsymbol{x}^*, \boldsymbol{y}^*)$: $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|^2 + \|\boldsymbol{y}^{(t)} - \boldsymbol{y}^*\|^2$, plus terms that capture heterogeneity.

### 4.1 Convergence in homogeneous setting

We now turn to stating the convergence rate in homogeneous setting.

**Theorem 4.1.** *Suppose each client's objective function $f_i$ satisfy Assumptions 1,2,3,4. If we use local SGDA*

*(Algorithm 1) under homogeneous data setting to optimize (1), choosing synchronization gap as* $\tau = \frac{T}{n \log T}$, *using learning rate* $\eta_x = \eta_y = \frac{4 \log T}{\mu T}$, *and by denoting* $\kappa = L/\mu$, *it holds that*

$$\mathbb{E}\left[\left\|\boldsymbol{x}^{(T)} - \boldsymbol{x}^*\right\|^2 + \left\|\boldsymbol{y}^{(T)} - \boldsymbol{y}^*\right\|^2\right] \leq \tilde{O}\left(\frac{1}{T^2} + \frac{\sigma^2}{\mu^2 nT} + \frac{\kappa^2\sigma^2}{\mu^2 nT} + \frac{\kappa^2\sigma^2}{\mu^2 nT^2}\right).$$

The proof of Theorem 4.1 is deferred to Appendix A. It can be observed that we obtain an $\tilde{O}(\frac{1}{nT})$ convergence rate with only $\tilde{O}(n)$ communication rounds. This indeed implies that we can achieve a linear speedup in terms of number of clients $n$, while significantly reducing the communication complexity from $T$ (fully synchronous SGDA) to $\tilde{O}(n)$ in strongly-convex-strongly-concave setting. We also note that the obtained bound matches the best known rate of local SGD for minimization problems [16], up to a logarithmic factor. The notable difference is that in [16], the communication rounds can be a constant, i.e., $O(n)$, but in our result, we have an extra logarithmic dependency on $T$. We leave removing this log factor as a future work.

### 4.2 Convergence in heterogeneous setting

We now turn to stating the convergence rate of local SGDA for strongly-convex-strongly-concave functions in heterogeneous local data setting. To this end, we first need to decide a proper notion to capture the heterogeneity among local functions by introducing the following quantities.

**Definition 2** (Heterogeneity at Optimum)**.** *The heterogeneity at global saddle point* $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ *is defined as*

$$\Delta_x = \frac{1}{n}\sum_{i=1}^n \|\nabla_x f_i(\boldsymbol{x}^*, \boldsymbol{y}^*)\|^2,$$
$$\Delta_y = \frac{1}{n}\sum_{i=1}^n \|\nabla_y f_i(\boldsymbol{x}^*, \boldsymbol{y}^*)\|^2.$$

Definition 2 is a generalized notion borrowed from [16], where they firstly employ it in the analysis of local SGD. It characterizes the heterogeneity of each local function at the global optimums of primal and dual variables. The following theorem establishes the convergence rate of local SGDA in heterogeneous settings.

**Theorem 4.2.** *Let each client's objective* $f_i$ *satisfy Assumptions 1,2,3,4. If we use local SGDA (Algorithm 1) under heterogeneous data setting to optimize (1), choosing synchronization gap* $\tau = \sqrt{T/n}$, *using decreasing learning rate* $\eta_x = \eta_y = \eta_t = \frac{8}{\mu(t+a)}$, *where* $a = \max\left\{2048\kappa^2\tau, 1024\sqrt{2}\tau\kappa^2, 256\kappa^2\right\}$, $\kappa = L/\mu$, *then the following convergence holds:*

$$\mathbb{E}\left[\left\|\boldsymbol{x}^{(T)} - \boldsymbol{x}^*\right\|^2 + \left\|\boldsymbol{y}^{(T)} - \boldsymbol{y}^*\right\|^2\right] \leq O\left(\frac{a^3}{T^3}\right) + O\left(\frac{\sigma^2}{\mu^2 nT}\right) + O\left(\frac{\kappa^2\left(\Delta_x + \Delta_y\right)}{\mu nT}\right) + O\left(\frac{\kappa^2\sigma^2}{\mu nT}\right).$$

The proof of Theorem 4.2 is deferred to Appendix A. Here we obtain an $O\left(\frac{\kappa^2(\Delta_x+\Delta_y)}{\mu nT}\right)$ rate using $\sqrt{nT}$ communication rounds, which also enjoys the linear speedup w.r.t. the number of the nodes. This result recovers the convergence rate of local SGD or FedAvg on strongly-convex minimization problems [16]. Our result does not need bounded gradient assumption, and we recover the linear dependency on function heterogeneity at global optimum, which matches the best bound for local SGD in minimization problems [16]. The most analogous work to ours is [2], where it achieves an $\tilde{O}(\frac{1}{T})$ rate with $O(n^{1/3}T^{2/3})$ communication rounds, which is worse than our result.

## 5 Nonconvex-Strongly-Concave Case

In this section we will present the convergence of local SGDA for nonconvex-strongly-concave functions. In this setting, since objective is no longer convex, we are unable to show the convergence to global saddle point. Thus, following the standard machinery in nonconvex-concave analysis [29, 47, 41], we introduce the following envelope function which will prove useful in convergence analysis.

**Definition 3.** *We define the following envelope functions to facilitate our analysis:*

$$\Phi(\boldsymbol{x}) = F(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})), \boldsymbol{y}^*(\boldsymbol{x}) = \arg\max_{\boldsymbol{y}\in\mathbb{R}^{d_y}} F(\boldsymbol{x}, \boldsymbol{y}). \quad (3)$$

We consider the convergence rate to the first order stationary point of $\Phi(\boldsymbol{x})$, as advocated in seminal nonconvex-concave minimax literature [29, 41, 47]. Namely, we will show how fast $\|\nabla\Phi(\boldsymbol{x})\|$ vanishes. Our analysis here mainly considers heterogeneous setting, but it can be easily generalized to homogeneous setting as well. We will use the following quantity to measure heterogeneity in nonconvex-strongly-concave case.

**Definition 4** (Gradient Dissimilarity)**.** *We define the following quantities to measure the gradient dissimilarity among local functions:*

$$\zeta_x = \sup_{(\boldsymbol{x},\boldsymbol{y})\in\mathbb{R}^{d_x}\times\mathbb{R}^{d_y}} \frac{1}{n}\sum_{i=1}^n \|\nabla_x f_i(\boldsymbol{x}, \boldsymbol{y}) - \nabla_x F(\boldsymbol{x}, \boldsymbol{y})\|^2,$$
$$\zeta_y = \sup_{(\boldsymbol{x},\boldsymbol{y})\in\mathbb{R}^{d_x}\times\mathbb{R}^{d_y}} \frac{1}{n}\sum_{i=1}^n \|\nabla_y f_i(\boldsymbol{x}, \boldsymbol{y}) - \nabla_y F(\boldsymbol{x}, \boldsymbol{y})\|^2.$$

Definition 4 is also a customary notion of heterogeneity in distributed optimization [23, 51], and we will use it to quantify the data heterogeneity in the nonconvex-nonconcave case. The following theorem establishes the convergence rate.

**Theorem 5.1.** *Let each client's objective function $f_i$ satisfy Assumptions 2-4. Running Algorithm 1 under heterogeneous data setting, choosing $\tau = \frac{T^{1/3}}{n^{1/3}}$ and learning rates $\eta_x = \frac{n^{1/3}}{LT^{2/3}}$ aand $\eta_y = \frac{2}{LT^{\frac{1}{2}}}$, if we choose sufficiently large $T$ such that*

$$T \geq \max\left\{40^{3/2}, \frac{160^3}{n^2}, \left(\frac{16n^{4/3}\kappa^4 + \sqrt{16n^{4/3}\kappa^8 - 12\beta n^{1/3}/L}}{2}\right)^3\right\},$$

*holds, then we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\Phi(\boldsymbol{x}^{(t)})\right\|^2\right] \leq O\left(\frac{\kappa^4 L^2\sigma^2}{(nT)^{1/3}} + \frac{L^2\zeta_x}{T^{2/3}} + \frac{L^2\zeta_y}{n^{2/3}T^{1/3}}\right),$$

*where $\kappa = L/\mu$, $\beta = L + \kappa L$.*

The proof of Theorem 5.1 is deferred to Appendix B. We note that when we assume local data distributions are homogeneous, the above rate still holds but the terms $\zeta_x$ and $\zeta_y$ that correspond to heterogeneity will disappear. Theorem 5.1 shows that local SGDA converges in the rate of $O\left(\frac{1}{(nT)^{1/3}}\right)$ with $O\left(n^{\frac{1}{3}}T^{\frac{2}{3}}\right)$ communication rounds. Also, local SGDA enjoys linear speedup in the number of workers $n$. The most analogue work to ours in this setting is [29], where they study the convergence of *centralized* SGDA (single machine) for noncovex-strongly-concave objectives, and achieve an $O(\frac{1}{\sqrt{T}})$ convergence rate. However, their algorithm requires that the mini-batch size of stochastic gradients to be very large, i.e., $O(\frac{1}{\epsilon^2})$ to reach an $\epsilon$-stationary point, therefore, requiring more computation budget per iteration. In our case, the batch size can be a constant, which avoids expensive large batch evaluations. We also note that as pointed out in [29], due to the nonsymmetric nature of the nonconvex-(strongly)-concave problem, we need different step sizes for primal and dual variables. In fact, since objective is strongly-concave in dual variable, we can choose a larger dual step size as stated in Theorem 5.1.

## 6 Local SGDA+

In this section, we proceed to an even harder seting where the objective is nonconvex in primal variable $\boldsymbol{x}$ and nonconcave in dual parameter $\boldsymbol{y}$. Nonconvex-nonconcave minimax optimization is an active research area due to the rise of GANs [8], and a few recent studies have proposed efficient algorithms for optimizing nonconvex-nonconcave objectives [26, 12, 50, 39]. However, these algorithms are all double loop: they require solving the maximization problem to get a $\epsilon$-accurate solution, and then go back to solve minimization problem. The drawbacks will be two-fold: first, they introduce a new hyperparameter $\epsilon$, which needs to be pre-tuned; second, the implementation will be more complicated, and is not straightforward to be extended to distributed setting. In this section, we propose a variant of local SGDA, dubbed as local SGDA+, aimed at solving nonconvex-nonconcave minimax problems in distributed setting with reduced communication overhead.

**Algorithm 2:** `Local SGDA+`

**Input:** Synchronization gap $\tau$, Snapshot gap $S$, Number of iterations $T$, Initial local models $\boldsymbol{x}_i^{(0)}$, $\boldsymbol{y}_i^{(0)}$ for $i \in [n]$.

**parallel for** $i = 1, ..., n$ **do**
  **for** $t = 0, ..., T-1$ **do**
    $\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} - \eta_x \nabla_x f_i\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{y}_i^{(t)}; \xi_i^t\right)$
    $\boldsymbol{y}_i^{(t+1)} = \boldsymbol{y}_i^{(t)} + \eta_y \nabla_y f_i\left(\tilde{\boldsymbol{x}}, \boldsymbol{y}_i^{(t)}; \xi_i^t\right)$
    **if** $t+1$ *divides* $\tau$ **then**
      all nodes send their local model $\boldsymbol{x}_i^{(t)}$ and $\boldsymbol{y}_i^{(t+1)}$ to server.
      $\boldsymbol{x}^{(t+1)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(t+1)}$;
      $\boldsymbol{y}^{(t+1)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i^{(t+1)}$;
      send $\boldsymbol{x}^{(t+1)}$, $\boldsymbol{y}^{(t+1)}$ to all nodes to update their local models.
      each client initializes its local models: $\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}^{(t+1)}$ and $\boldsymbol{y}_i^{(t+1)} = \boldsymbol{y}^{(t+1)}$.
    **end**
    **if** $t+1$ *divides* $S$ **then**
      all nodes send their local model $\boldsymbol{x}_i^{(t+1)}$ to server.
      take snapshot: $\tilde{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(t+1)}$;
      send $\tilde{\boldsymbol{x}}$ to all nodes.
    **end**
  **end**
**end**

**Our proposal: snapshot iterate and stale gradients.** Before introducing our algorithm, let us first discuss the single machine setting to illustrate our main ideas. In the vanilla single loop (S)GDA, we query the gradient based on current iterate $(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)})$. It posts difficulty to prove the convergence since under nonconcavity assumption, we do not know how close $\boldsymbol{y}^{(t)}$ is to $\boldsymbol{y}^*(\boldsymbol{x}^{(t)})$ (as elaborated in [29], in nonconcave case, $\boldsymbol{y}^*(\cdot)$ is not even Lipschitz). As a result, the existing methods mainly follow a double loop schema: at outer

loop, we update $\boldsymbol{x}^{(t-1)}$ using SGD or its variants to get $\boldsymbol{x}^{(t)}$, and then, we fix $\boldsymbol{x}^{(t)}$, and run few steps of stochastic gradient ascent to solve inner maximization problem: $\max_{\boldsymbol{y}\in\mathbb{R}^{d_y}} F(\boldsymbol{x}^{(t)}, \boldsymbol{y})$ to get an $\epsilon$-accurate approximation of $\boldsymbol{y}^*(\boldsymbol{x}^{(t)})$, where $\epsilon$ is the predetermined level of accuracy. This is a successful algorithm, but due the two weaknesses we mentioned before, we prefer a single loop algorithm is distributed setting. In order to alleviate the need for the inner loop, we propose to update $\boldsymbol{y}$ with **stale gradients** evaluated on some past **snapshot iterate** $\tilde{\boldsymbol{x}}$. To be more specific, each local worker will perform following update:

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} - \eta_x \nabla_x f_i\left(\boldsymbol{x}_i^{(t)}, \boldsymbol{y}_i^{(t)}; \xi_i^t\right),$$
$$\boldsymbol{y}_i^{(t+1)} = \boldsymbol{y}_i^{(t)} + \eta_y \nabla_y f_i\left(\tilde{\boldsymbol{x}}, \boldsymbol{y}_i^{(t)}; \xi_i^t\right).$$

The update for primal $\boldsymbol{x}_i^{(t)}$ is identical to what we did in local SGDA, however, when we update the dual model $\boldsymbol{y}_i^{(t)}$, instead of evaluating gradient on $\boldsymbol{x}_i^{(t)}$, we query gradient evaluated on a snapshot iterate $\tilde{\boldsymbol{x}}$, which will be updated every $S$ iterations. This updating scheme can guarantee that we can optimize on $\boldsymbol{y}$ for fixed $\boldsymbol{x}$ but without actually *locking* the update of $\boldsymbol{x}$. This algorithm will no longer need the inner loop hyperparameter $\epsilon$ and it is easy to be implemented in a distributed fashion. The detailed steps of local SGDA+ are provided in Algorithm 2. We note that by choosing a small primal learning rate, $\tilde{\boldsymbol{x}}$ will not drift far away from current iterate $\boldsymbol{x}_i^{(t)}$, and hence its convergence is guaranteed.

### 6.1 Convergence of local SGDA+

We now establish the convergence of local SGDA+ for a class of nonconvex-nonconcave function. We consider two function class: (i) $F(\boldsymbol{x}, \boldsymbol{y})$ is nonconvex in $\boldsymbol{x}$, and satisfies PL-condition in $\boldsymbol{y}$. (ii) $F(\boldsymbol{x}, \boldsymbol{y})$ is nonconvex in $\boldsymbol{x}$, and one-point concave in $\boldsymbol{y}$. To do so, we make the following assumptions on the objective.

**Assumption 5** (Polyak-Łojasiewicz Condition)**.** *$F(\boldsymbol{x}, \boldsymbol{y})$ is said to satisfy Polyak-Łojasiewicz (PL) condition in $\boldsymbol{y}$ if $\forall \boldsymbol{x} \in \mathbb{R}^{d_x}$, the following holds: $\frac{1}{2}\left\|\nabla_y F(\boldsymbol{x}, \boldsymbol{y})\right\|^2 \geq \mu\left(F(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) - F(\boldsymbol{x}, \boldsymbol{y})\right), \forall \boldsymbol{y} \in \mathbb{R}^{d_y}$.*

**Assumption 6** (Lipschitz Continuity in $\boldsymbol{x}$)**.** *$F(\boldsymbol{x}, \boldsymbol{y})$ is said to be $G_x$-Lipschitz in $\boldsymbol{x}$ if the following holds: $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{d_x}$: $\left\|F(\boldsymbol{x}, \boldsymbol{y}) - F(\boldsymbol{x}', \boldsymbol{y})\right\| \leq G_x \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|$.*

The following theorem establishes the convergence rate of local SGDA+ on nonconvex-PL objectives.

**Theorem 6.1** (Nonconvex-PL)**.** *Let objective function $F$ satisfies Assumption 5 and local functions satisfy Assumptions 3 and 6. Also assume $F$ is $G_x$ Lipschitz in $\boldsymbol{x}$. Running Algorithm 2 under heterogeneous data setting, by choosing $\tau = T^{1/3}$, $S = T^{2/3}$, $\eta_x = \frac{n^{1/3}}{LT^{2/3}}$, $\eta_y = \frac{n^{1/3}}{LT^{1/2}}$, $\tau = \frac{T^{1/3}}{n^{2/3}}$, and $S = \frac{T^{1/3}}{n^{2/3}}$, if we set*

$$T \geq \max\left\{(8\kappa^2)^6, \right.$$
$$\left. O\left(\frac{\beta n^{1/3}}{2L} + \sqrt{\frac{8L(L+\beta)n^{1/3}}{\mu^2} + \frac{4L^2 n^{2/3}}{\mu}}\right)^{3/2}\right\},$$

*then it holds that*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\Phi(\boldsymbol{x}^{(t)})\right\|^2\right]$$
$$\leq O\left(\frac{\beta\sigma^2}{(nT)^{1/3}} + \frac{\kappa^2 L^2 \zeta_y}{n^{2/3}T^{1/3}} + \frac{\kappa^2 L^2 \zeta_x}{n^{2/3}T} + \frac{\kappa^2 L^2 G_x^2}{T}\right),$$

*where $\kappa = L/\mu$, $\beta = L + \kappa L$.*

The proof of Theorem 6.1 is deferred to Appendix C. Again, if we assume local functions are homogeneous, this rate also holds but the terms $\zeta_x$ and $\zeta_y$ will disappear. Here we obtain an $O\left(\frac{1}{(nT)^{1/3}}\right)$ rate with only $O(T^{2/3})$ communication rounds, as good as what we get in nonconvex-strongly-concave case. The most analogous work is [43], where they prove the convergence rate of vanilla local SGDA on nonconvex-PL game. Their work shows that, the vanilla local SGDA can still converge under nonconvex-PL condition. However, their analysis does not generalize to nonconvex-one-point-concave setting, but we develop the convergence theory of local SGDA+, as we will present in the next theorem. Another similar work is [39], where they study the single machine algorithm in nonconvex-PL setting. They propose a double loop gradient descent ascent, and achieve and $\tilde{O}\left(\frac{1}{T^{1/2}}\right)$ convergence rate under their convergence measure, which is recognized as the first analysis for nonconvex-PL game, to our best knowledge.

Now, we proceed to an even harder case: the objective is nonconvex in $\boldsymbol{x}$ and one-point concave in $\boldsymbol{y}$. One point convexity/concavity property has been shown to hold under the dynamic of SGD on optimizing neural networks [24, 17, 56], which has been demonstrated both theoretically and empirically. In addition, some works on minimax optimization also adapt similar assumption [35, 32, 31, 11]. Since our objective is no longer strongly-concave or PL in $\boldsymbol{y}$, then it will be difficult to analyze the dynamic of $\Phi$ directly, because $\Phi$ is not smooth any more. Instead, we study the *Moreau envelope* of $\Phi$, in order to analyze the convergence, as suggested in several recent studies [5, 29, 41].

**Definition 5** (Moreau Envelope)**.** *A function $\Phi_p(\boldsymbol{x})$ is the $p$-Moreau envelope of a function $\Phi$ if $\Phi_p(\boldsymbol{x}) := \min_{\boldsymbol{x}'\in\mathbb{R}^{d_x}}\left\{\Phi(\boldsymbol{x}') + \frac{1}{2p}\left\|\boldsymbol{x}' - \boldsymbol{x}\right\|^2\right\}$.*

We will use $1/2L$-Moreau envelope of $\Phi$, following the setting in [29, 41], and state the convergence rate in terms of $\|\nabla\Phi_{1/2L}(\boldsymbol{x})\|$.

**Assumption 7** (One Point Concavity). *$f_i(\boldsymbol{x},\boldsymbol{y})$ is said to satisfy one point concavity in $\boldsymbol{y}$ if we fix $\boldsymbol{x}$, then $\forall \boldsymbol{y}\in\mathbb{R}^{d_y}$, the following holds: $\langle\nabla_y f_i(\boldsymbol{x},\boldsymbol{y}), \boldsymbol{y}-\boldsymbol{y}^*(\boldsymbol{x})\rangle \le f_i(\boldsymbol{x},\boldsymbol{y}) - f_i(\boldsymbol{x},\boldsymbol{y}^*(\boldsymbol{x}))$.*

**Theorem 6.2** (Nonconvex-One-Point-Concave). *Let local functions satisfy Assumptions 3, 6 and 7, and $\|\boldsymbol{y}^{(t)}\|^2 \le \frac{D}{2}$, $\|\boldsymbol{y}^*(\tilde{\boldsymbol{x}})\|^2 \le \frac{D}{2}$ for all $t$ and $\tilde{\boldsymbol{x}}$ during iterating. Also assume $F$ is $G_x$ Lipschitz in $\boldsymbol{x}$. Running Algorithm 2 under heterogeneous data setting, by choosing $\eta_x = \frac{1}{LT^{\frac{5}{6}}}$, $\eta_y = \frac{1}{4LT^{\frac{1}{2}}}$, $\tau = T^{\frac{1}{3}}/n^{\frac{1}{6}}$, $S = T^{\frac{2}{3}}$, it holds that:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\Phi_{1/2L}(\boldsymbol{x}^{(t)})\right\|^2\right] \le O\left(\frac{L\sigma^2}{T^{1/6}}\right) + O\left(\frac{L^2\sigma^2+LG_x^2}{(nT)^{1/3}} + \frac{L^2\zeta_x}{n^{1/3}T} + \frac{L^2\zeta_y}{(nT)^{1/3}}\right) + O\left(\frac{D}{T^{1/6}}\right).$$

The proof of Theorem 6.2 is deferred to Appendix D. Local SGDA+ is guaranteed to find the first order stationary point of $\Phi_{1/2L}(\boldsymbol{x})$ at the rate of $O(\frac{1}{T^{1/6}})$ with $T^{2/3}$ communication rounds. Again, if we assume local functions are homogeneous, this rate also holds but the terms $\zeta_x$ and $\zeta_y$ will disappear. The most similar work to ours is [29], where they analyze the single machine SGDA on nonconvex-concave setting, and established a rate of $O(\frac{1}{T^{1/4}})$. In contrast, we consider a more difficult concave setting, and their analysis technique does not apply here directly.

## 7 Experiments

In this section, we empirically examine the convergence of the proposed algorithms local SGDA and local SGDA+. We use two datasets, MNIST and a synthetic dataset and develop our code using `distributed` API of `PyTorch`. For the Algorithm 1, to have a strongly convex-strongly concave loss function, we consider the robust linear regression problem, and for the Algorithm 2, to construct a nonconvex-nonconcave problem, we consider the robust neural network training, and use a 2-layer MLP model with a cross entropy loss function. First, we explain the generation of non-iid datasets and then turn into the experimental results.

**Datasets.** To generate a synthetic non-iid dataset, we follow the steps from [21]. In here, we only use the parameter to control the divergence between local datasets, while the true models for data generation of each node is coming from the same distribution. Hence, for each node we generate a weight matrix $\boldsymbol{W}_i \in \mathbb{R}^{m\times 1}$ and a bias $\boldsymbol{b}\in\mathbb{R}^c$, where the output for the $i$th client is $y_i = \boldsymbol{W}_i^\top \boldsymbol{x}_i + b$. The model is generated based on a Gaussian distribution $\boldsymbol{W}_i \sim \mathcal{N}(0,1)$ and $\boldsymbol{b}_i \sim \mathcal{N}(0,1)$. The input data $\boldsymbol{x}_i \in \mathbb{R}^m$ has $m$ features and is drown from a Gaussian distribution $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}_i \sim \mathcal{N}(M_i, 1)$ and $M_i \sim \mathcal{N}(0,\alpha)$. Also the variance $\boldsymbol{\Sigma}$ is a diagonal matrix with value of $\boldsymbol{\Sigma}_{k,k} = k^{-1.2}$. In this process, by changing $\alpha$ we can control the divergence between local input data of different nodes. We create 3 different datasets by changing this parameter for the regression task, namely, Synthetic (0.0), Synthetic (0.25), and Synthetic (0.5). For the MNIST dataset and for the classification task, we follow the same procedure in [34], where we allocate data from only 2 classes per node. This way, the data is distributed heterogeneously among nodes.

**Robust Linear Regression.** In this experiments the model and loss function is defined as

$$\min_{\boldsymbol{w}}\max_{\|\boldsymbol{\delta}\|^2\le 1}\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{w}^\top(\boldsymbol{x}_i+\boldsymbol{\delta})-y_i)^2 + \frac{1}{2}\|\boldsymbol{w}\|^2,$$

For the convergence measure, we can use the robust loss. Given a model $\hat{\boldsymbol{w}}$, its robust loss is defined as

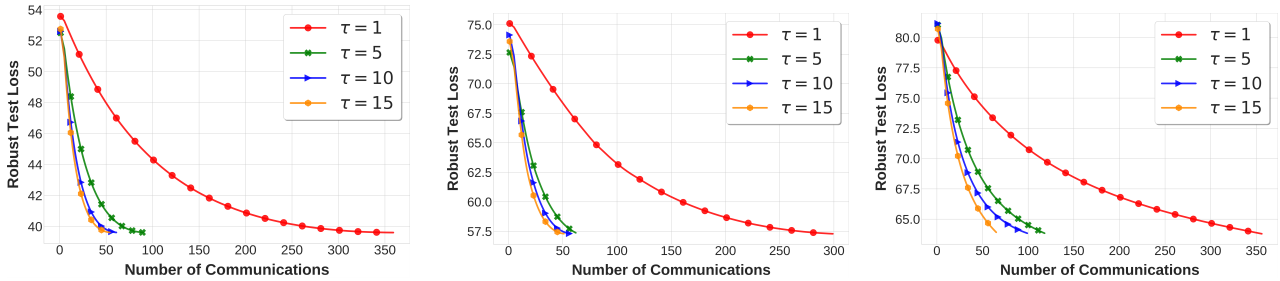
$$\ell(\hat{\boldsymbol{w}}) = \max_{\|\boldsymbol{\delta}\|^2\le 1}\frac{1}{n}\sum_{i=1}^{n}(\hat{\boldsymbol{w}}^\top(\boldsymbol{x}_i+\boldsymbol{\delta})-y_i)^2 + \frac{1}{2}\|\hat{\boldsymbol{w}}\|^2,$$

so each time to evaluate a node's robust loss, we have to solve above maximization problem. One way to do it is to run few steps of gradient ascent to get a estimated $\hat{\delta}$.

In the first set of experiments, we run the training procedure proposed in Algorithm 1 on synthetic datasets that introduced before. We set the input dimension to 60 and each node has between 400 to 500 samples. We generate data for 100 nodes, and drawn 20% of each node's data for the test dataset to make it the average distribution among all nodes. We use the same learning rates for both dual and primal variables, and use a decaying mechanism to decrease it by 5% every iteration. The initial learning rate for all the experiments is set to 0.001. The results of this experiment is depicted in Figure 1, where we compare the local SGDA ($\tau \in \{5, 10, 15\}$) with normal SGDA ($\tau = 1$). It is clear that to achieve certain level of the robust loss, local SGDA needs significantly fewer number of communication rounds, compared to vanilla SGDA, hence it achieves communication efficiency.

**Robust Neural Network Training.** Similar to the setting of Robust Linear Regression in [39], here we just replace the model with a DNN and optimize

$$\min_{\boldsymbol{W}}\max_{\|\boldsymbol{\delta}\|^2\le 1}\frac{1}{n}\sum_{i=1}^{n}\ell(h_{\boldsymbol{W}}(\boldsymbol{x}_i+\boldsymbol{\delta}), y_i).$$




(a) Synthetic (0.0) (b) Synthetic (0.25) (c) Synthetic (0.5)

Figure 1: Linear regression with synthetic datasets using local SGDA ($\tau > 1$) comparing to SGDA ($\tau = 1$). Local SGDA can achieve the same robust loss with fewer number of communication rounds than SGDA.

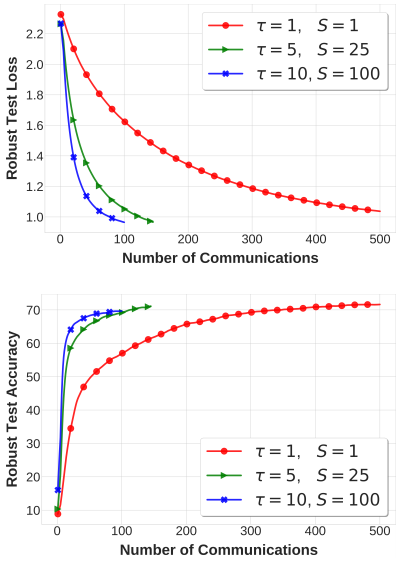



Figure 2: Comparing local SGDA+ with normal SGDA ($\tau = 1$, $S = 1$) on training a 2-layer MLP on heterogeneous MNIST dataset over 100 nodes. Local SGDA+ acn converge to the same accuracy as SGDA with fewer rounds of communication between nodes and the server.

For this experiment, to evaluate Algorithm 2, we use a 2-layer MLP, each with 200 neurons followed by ReLU activation and a cross entropy loss function at the end. We divide the MNIST dataset among 100 nodes, each with only having access to 2 classes to introduce heterogeneity among local data shards. The test dataset is a pool of all classes, hence, it is the average dataset over all nodes. We use the same decaying learning rate scheme as the linear regression, where the initial learning rate is set to 0.01. In this experiment, we set the snapshot gap $S = \tau^2$, as suggested in Theorem 6.1. The convergence measure is the robust accuracy, and we compute it similarly to robust loss as in robust linear regression. The results of these experiments are shown in Figure 2, where compared to normal SGDA ($\tau = 1$, $S = 1$), the proposed local SGDA+ can converge faster in terms of number of communications.

## 8 Conclusions and Path Forward

In this paper we proposed a communication efficient distributed method to solve minimax optimization problems and establish its convergence rate for strongly-convex-strongly-concave and nonconvex-strongly-concave objectives in both homogeneous and heterogeneous data distribution settings. We also proposed a single loop variant of proposed algorithm to address nonconvex-noncancave problems that arises in learning GANs. The present work is the first to study local SGD method in minimax setting and leaves many interesting directions as future work. We believe some of the obtained rates can be tightened. Investigating the achievable rates via local methods in minimax setting also remains open. Another future work will be the exploration of faster algorithm to match the known lower bound of first order minimax algorithm obtained in [40].

## Acknowledgement

We would like to thank Mohammad Mahdi Kamani for his help on conducting the experiments. This work has been done using the Extreme Science and Engineering Discovery Environment (XSEDE) resources, which is supported by National Science Foundation under grant number ASC200045.

### References

[1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.

[2] Aleksandr Beznosikov, Valentin Samokhin, and Alexander Gasnikov. Local sgd for saddle-point problems. *arXiv preprint arXiv:2010.13112*, 2020.

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

[4] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. In *International Conference on Learning Representations (ICLR 2018)*, 2018.

[5] Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

[6] Simon S Du and Wei Hu. Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 196–205, 2019.

[7] Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. *arXiv preprint arXiv:1802.10551*, 2018.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[9] Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. In *Advances in Neural Information Processing Systems*, pages 11080–11092, 2019.

[10] Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

[11] Alfredo N Iusem, Alejandro Jofré, Roberto Imbuzeiro Oliveira, and Philip Thompson. Extragradient method with variance reduction for stochastic variational inequalities. *SIAM Journal on Optimization*, 27(2):686–724, 2017.

[12] Chi Jin, Praneeth Netrapalli, and Michael I Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? *arXiv preprint arXiv:1902.00618*, 2019.

[13] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

[14] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

[15] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Better communication complexity for local sgd. *arXiv preprint arXiv:1909.04746*, 2019.

[16] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *AISTAT*, 2020.

[17] Robert Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does sgd escape local minima? *arXiv preprint arXiv:1802.06175*, 2018.

[18] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

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

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

[21] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

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

[23] Xiang Li, Wenhao Yang, Shusen Wang, and Zhihua Zhang. Communication efficient decentralized training with multiple local updates. *arXiv preprint arXiv:1910.09126*, 2019.

[24] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. *Advances in neural information processing systems*, 30:597–607, 2017.

[25] Tengyuan Liang and James Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 907–915, 2019.

[26] Qihang Lin, Mingrui Liu, Hassan Rafique, and Tianbao Yang. Solving weakly-convex-weakly-concave saddle-point problems as weakly-monotone variational inequality. *arXiv preprint arXiv:1810.10207*, 2018.

[27] Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local sgd. *ICLR*, 2019.

[28] Tianyi Lin, Chi Jin, Michael Jordan, et al. Near-optimal algorithms for minimax optimization. *arXiv preprint arXiv:2002.02417*, 2020.

[29] Tianyi Lin, Chi Jin, and Michael I Jordan. On gradient descent ascent for nonconvex-concave minimax problems. *arXiv preprint arXiv:1906.00331*, 2019.

[30] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017.

[31] Mingrui Liu, Youssef Mroueh, Jerret Ross, Wei Zhang, Xiaodong Cui, Payel Das, and Tianbao Yang. Towards better understanding of adaptive gradient algorithms in generative adversarial nets. *arXiv preprint arXiv:1912.11940*, 2019.

[32] Mingrui Liu, Youssef Mroueh, Wei Zhang, Xiaodong Cui, Tianbao Yang, and Payel Das. Decentralized parallel algorithm for training generative adversarial nets. *arXiv preprint arXiv:1910.12999*, 2019.

[33] David Mateos-Núnez and Jorge Cortés. Distributed subgradient methods for saddle-point problems. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 5462–5467. IEEE, 2015.

[34] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[35] Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. *arXiv preprint arXiv:1807.02629*, 2018.

[36] Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. Convergence rate of o (1/k) for optimistic gradient and extra-gradient methods in smooth convex-concave saddle point problems. *arXiv preprint arXiv:1906.01115*, 2019.

[37] Arkadi Nemirovski. 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, 2004.

[38] Yurii Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2-3):319–344, 2007.

[39] Maher Nouiehed, Maziar Sanjabi, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *arXiv preprint arXiv:1902.08297*, 2019.

[40] Yuyuan Ouyang and Yangyang Xu. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Mathematical Programming*, pages 1–35, 2019.

[41] Hassan Rafique, Mingrui Liu, Qihang Lin, and Tianbao Yang. Non-convex min-max optimization: Provable algorithms and applications in machine learning. *arXiv preprint arXiv:1810.02060*, 2018.

[42] Mohammad Rasouli, Tao Sun, and Ram Rajagopal. Fedgan: Federated generative adversarial networks for distributed data. *arXiv preprint arXiv:2006.07228*, 2020.

[43] Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. Robust federated learning: The case of affine distribution shifts. *Advances in Neural Information Processing Systems*, 33, 2020.

[44] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[45] Kunal Srivastava, Angelia Nedić, and Dušan Stipanović. Distributed min-max optimization in networks. In *2011 17th International Conference on Digital Signal Processing (DSP)*, pages 1–8. IEEE, 2011.

[46] Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

[47] Kiran K Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. Efficient algorithms for smooth minimax optimization. In *Advances in Neural Information Processing Systems*, pages 12659–12670, 2019.

[48] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 1, 2008.

[49] Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

[50] Yuanhao Wang, Guodong Zhang, and Jimmy Ba. On solving minimax optimization locally: A follow-the-ridge approach. *arXiv preprint arXiv:1910.07512*, 2019.

[51] Blake Woodworth, Kumar Kshitij Patel, and Nathan Srebro. Minibatch vs local sgd for heterogeneous distributed learning. *arXiv preprint arXiv:2006.04735*, 2020.

[52] Blake Woodworth, Kumar Kshitij Patel, Sebastian U Stich, Zhen Dai, Brian Bullins, H Brendan McMahan, Ohad Shamir, and Nathan Srebro. Is local sgd better than minibatch sgd? *arXiv preprint arXiv:2002.07839*, 2020.

[53] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd for non-convex optimization with faster convergence and less communication. *arXiv preprint arXiv:1807.06629*, 2018.

[54] Honglin Yuan and Tengyu Ma. Federated accelerated stochastic gradient descent. *arXiv preprint arXiv:2006.08950*, 2020.

[55] Hantian Zhang, Jerry Li, Kaan Kara, Dan Alistarh, Ji Liu, and Ce Zhang. Zipml: Training linear models with end-to-end low precision, and a little bit of deep learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4035–4043. JMLR. org, 2017.

[56] Yi Zhou, Junjie Yang, Huishuai Zhang, Yingbin Liang, and Vahid Tarokh. Sgd converges to global minimum in deep learning via star-convex path. *arXiv preprint arXiv:1901.00451*, 2019.