Jointly Improving the Sample and Communication Complexities in Decentralized Stochastic Minimax Optimization

Xuan Zhang[1], **Gabriel Mancino-Ball**[2], **Necdet Serhat Aybat**[1], **Yangyang Xu**[2]

[1]Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, University Park, PA
[2]Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180
xxz358@psu.edu, gabriel.mancino.ball@gmail.com, nsa10@psu.edu, xuy21@rpi.edu

Abstract

We propose a novel single-loop decentralized algorithm, DGDA-VR, for solving the stochastic nonconvex strongly-concave minimax problems over a connected network of agents, which are equipped with stochastic first-order oracles to estimate their local gradients. DGDA-VR, incorporating variance reduction, achieves $\mathcal{O}(\epsilon^{-3})$ oracle complexity and $\mathcal{O}(\epsilon^{-2})$ communication complexity *without* resorting to multi-communication rounds – both are *optimal*, i.e., matching the lower bounds for this class of problems. Since DGDA-VR does not require multiple communication rounds, it is applicable to a broader range of decentralized computational environments. To the best of our knowledge, this is the first distributed method using a single communication round in each iteration to jointly optimize the oracle and communication complexities for the problem considered here.

Introduction

This paper considers a connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of M agents which cooperatively solve

$$\min_{\mathbf{x}\in\mathbb{R}^d} \max_{\mathbf{y}\in\mathbb{R}^m} f(\mathbf{x},\mathbf{y}) \triangleq \frac{1}{M}\sum_{i=1}^{M} f_i(\mathbf{x},\mathbf{y}), \tag{1}$$

where $f_i : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ is smooth and possibly *nonconvex* in $\mathbf{x} \in \mathbb{R}^d$ and are strongly-concave in $\mathbf{y} \in \mathbb{R}^m$ for $i = 1, 2, ..., M$. Furthermore, each agent-i can only access unbiased stochastic gradients $\tilde{\nabla} f_i$ rather than exact gradients ∇f_i, and we assume that $\{\tilde{\nabla} f_i\}_{i\in\mathcal{V}}$ have finite variances, uniformly bounded by some $\sigma > 0$. The set $\mathcal{V} = \{1, 2, \ldots, M\}$ indexes the M agents and $(i, j) \in \mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ only if agent i can send information to agent j. Minimax optimization has garnered recent interest due to applications in many machine learning settings such as adversarial training (Goodfellow et al. 2014; Liu et al. 2020), distributionally robust optimization (Namkoong and Duchi 2016; Xian et al. 2021), reinforcement learning (Zhang et al. 2021c), and fair machine learning (Nouiehed et al. 2019). The problem in (1) arises naturally when the data is physically distributed among many agents or is too large to store on a single computing device (Xin, Khan, and Kar 2021a). It is well known that *centralized* methods suffer from communication bottlenecks on the parameter server (Lian et al. 2017; Xian et al. 2021) and potential data privacy violations (Verbraeken et al. 2020); hence, decentralized methods have emerged as a practical alternative to overcome these issues.

In a decentralized setting, only agent-i has access to f_i and its stochastic gradient oracle; thus, in order for the M agents to collaboratively solve (1), each agent-i will make a local copy, denoted as $(\mathbf{x}_i, \mathbf{y}_i)$, of the primal-dual variable $(\mathbf{x}, \mathbf{y})$ and communicate the local variables and gradient information with its immediate (1-hop) neighbors. In this way, (1) can be reformulated equivalently into the following problem in a decentralized format:

$$\begin{aligned}&\min_{\{\mathbf{x}_i\}_{i\in\mathcal{V}}} \max_{\{\mathbf{y}_i\}_{i\in\mathcal{V}}} \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} f_i(\mathbf{x}_i,\mathbf{y}_i)\\ &\text{s.t. } \mathbf{x}_i = \mathbf{x}_j, \quad \mathbf{y}_i = \mathbf{y}_j, \ \forall\,(i,j)\in\mathcal{E}.\end{aligned} \tag{2}$$

Consensus among the agents is then enforced through the use of a *mixing matrix* encoding the topology of $\mathcal{G}$.

Although there are decentralized algorithms for stochastic nonconvex-strongly-concave minimax problems, the existing work (Liu et al. 2020; Chen, Ye, and Luo 2022) requires *multi-communication rounds* at each iteration; hence, they can be analyzed as a centralized algorithm with inexact gradients. In a multi-agent setting, methods requiring multi-communication rounds per iteration are *not* desired as they require more strict coordination among the agents while single round communication methods are much easier to implement. We will design a decentralized algorithm for (1) or equivalently (2) that only requires a single communication round per iteration. Although another recent work (Xian et al. 2021) also proposed a decentralized algorithm for the same setting with a single round of communication per iteration, we noticed that its proof has a fundamental issue and the claimed complexity results do not hold — we explain this problem in detail when we compare our results with the existing work below. In addition, the communication complexity of the algorithm in (Xian et al. 2021) is intrinsically of the same order with its oracle complexity; hence, it cannot be optimal. In contrast, the method we propose can achieve an optimal complexity result for both oracle complexity and communication complexity in terms of its dependence on a given tolerance $\epsilon > 0$ for ϵ-stationarity, defined below.

Contributions. Our contributions are two-fold. First, we propose a decentralized stochastic gradient-type method, called DGDA-VR, for solving (1) or equivalently (2). At every iteration of the method, each agent-i performs a local stochastic gradient descent step for $\mathbf{x}_i$ and a local stochastic gradient ascent step for $\mathbf{y}_i$, along a tracked (global) stochastic gradient direction. DGDA-VR needs only a single communication round per iteration among neighbors for (weighted) averaging local variables and tracking the global stochastic gradient information.

Second, we show that when each agent uses a SPIDER-type stochastic gradient estimator (Fang et al. 2018), which is a variant of SARAH (Nguyen et al. 2017a), DGDA-VR can, in a decentralized manner, generate $\{\mathbf{z}_i(\epsilon)\}_{i\in\mathcal{V}}$ with $\mathbf{z}_i(\epsilon) \triangleq (\mathbf{x}_i(\epsilon), \mathbf{y}_i(\epsilon))$ such that the local decisions $\{\mathbf{z}_i(\epsilon)\}_{i\in\mathcal{V}}$ and their average $(\bar{\mathbf{x}}_\epsilon, \bar{\mathbf{y}}_\epsilon) = \bar{\mathbf{z}}_\epsilon = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \mathbf{z}_i(\epsilon)$ have the following properties:

1. $\bar{\mathbf{x}}_\epsilon$ is an *ϵ-stationary point* of the primal function $\Phi(\cdot) \triangleq \max_{\mathbf{y}} f(\cdot, \mathbf{y})$, i.e., $\mathbf{E}[\|\nabla\Phi(\bar{\mathbf{x}}_\epsilon)\|] \le \epsilon$;
2. $\bar{\mathbf{y}}_\epsilon$ is an *$\mathcal{O}(\epsilon)$-optimal-response* to $\bar{\mathbf{x}}_\epsilon$, i.e., $\mathbf{E}[\|\bar{\mathbf{y}}_\epsilon - \mathbf{y}^*(\bar{\mathbf{x}}_\epsilon)\|] = \mathcal{O}(\epsilon)$, where $\mathbf{y}^*(\bar{\mathbf{x}}_\epsilon) = \operatorname{argmax}_{\mathbf{y}} f(\bar{\mathbf{x}}_\epsilon, \mathbf{y})$;
3. $\{\mathbf{z}_i(\epsilon)\}_{i\in\mathcal{V}}$ has *$\mathcal{O}(\epsilon)$-consensus-violation*, i.e., $\mathbf{E}[\sum_{i\in\mathcal{V}} \|\mathbf{z}_i(\epsilon) - \bar{\mathbf{z}}_\epsilon\|^2] = \mathcal{O}(\epsilon^2)$;
4. computing $\{\mathbf{z}_i(\epsilon)\}_{i\in\mathcal{V}}$ requires $\mathcal{O}((1-\rho)^{-2}\epsilon^{-2})$ communication among neighboring nodes, which employ $\mathcal{O}(\sigma(1-\rho)^{-2}\epsilon^{-3})$ stochastic oracle calls, i.e., the sampling complexity — here, $\rho \in [0, 1)$ measures the connectivity of the underlying communication network, and a smaller ρ means a more connected network. The orders $\mathcal{O}(\epsilon^{-2})$ for communication rounds and $\mathcal{O}(\epsilon^{-3})$ for stochastic gradient oracles both match with existing lower bounds (Sun and Hong 2019; Arjevani et al. 2022).

Notation and definitions. Throughout the paper, we use bold lower-case letters $\mathbf{x}, \mathbf{y}, \ldots$ to denote vectors and upper-case letters $X, Y, \ldots$ to denote matrices. $\|\cdot\|$ denotes the *Euclidean norm* for a vector. $\|\cdot\|_F$ and $\|\cdot\|_2$ denote the *Frobenius norm*, and the *spectral norm* of a matrix, respectively. The symbols $\mathbf{I}$ and $\mathbf{1}$ denote the identity matrix and the column vector with all elements 1, respectively. The symbol $\mathbf{E}$ is used for expectation. W represents a mixing matrix and $\Pi \triangleq \frac{1}{M}\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{M\times M}$ the averaging matrix. We let $\mathbb{N}^+ \triangleq \mathbb{N}/\{0\}$. Given $M \in \mathbb{N}^+$, $[M]$ denotes the integer set $\{1, 2, .., M\}$. Given a random variable ξ, for any $i \in [M]$, $\tilde{\nabla} f_i(\mathbf{x}, \mathbf{y}; \xi)$ denotes an unbiased estimator of $\nabla f_i(\mathbf{x}, \mathbf{y})$, of which properties are stated in Assumptions 4 and 5. We interchangeably use $\mathbb{R}^d \times \mathbb{R}^m = \mathbb{R}^{d+m}$ when it is convenient to define the inputs to f_i as a single vector. We will compactly use matrix variables for the formulation in (2):

$$X \triangleq [\mathbf{x}_1, \ldots, \mathbf{x}_M]^\top,\ Y \triangleq [\mathbf{y}_1, \ldots, \mathbf{y}_M]^\top,\ Z \triangleq [X, Y].$$

Organization. We first briefly discuss the previous work on decentralized minimax problems related to ours. After we give some important definitions and state our assumptions, we describe our proposed method and main results in detail. Finally, we test our method against the SOTA methods employing variance reduction on a game problem and two different robust machine learning problems.

Related Work

We provide a brief literature review on *decentralized optimization* methods (specifically for nonconvex and stochastic problems), and discuss both centralized and decentralized methods for minimax problems.

Decentralized optimization. D-PSGD (Lian et al. 2017) first advocated for the use of decentralized methods and provided convergence analysis for a stochastic gradient-type method. D^2 (Tang et al. 2018) improved the analysis of D-PSGD to allow for data heterogeneity. More recently, gradient tracking has been utilized to further enhance the convergence rate of new methods; see (Lu et al. 2019; Zhang and You 2020; Koloskova, Lin, and Stich 2021; Xin, Khan, and Kar 2021b) for further discussions. Variance reduction methods that mimic updates from the SARAH (Nguyen et al. 2017b) and SPIDER (Wang et al. 2019) methods provide optimal gradient complexity results at the expense of large batch computations; examples include D-SPIDER-SFO (Pan, Liu, and Wang 2020), D-GET (Sun, Lu, and Hong 2020), GT-SARAH (Xin, Khan, and Kar 2022), DESTRESS (Li, Li, and Chi 2022). To avoid the large batch requirement of these methods, the STORM (Cutkosky and Orabona 2019; Xu and Xu 2023) and Hybrid-SGD (Tran-Dinh et al. 2022a) methods have also been adapted to the *decentralized setting*; see GT-STORM (Zhang et al. 2021b) and GT-HSGD (Xin, Khan, and Kar 2021a). Both types of variance reduction have recently been extended to include a proximal term in ProxGT-SR-O/E (Xin et al. 2021) and DEEPSTORM (Mancino-Ball et al. 2023). There are many other decentralized methods which handle various problem settings, but an exhaustive discussion is beyond the scope of this work; we refer interested readers to the references in the above works for more details.

Minimax optimization. Before discussing purely decentralized minimax optimization methods, we first provide a brief overview of minimax optimization methods in the *centralized setting*. In recent years, a significant amount of work has been proposed (Chen, Ma, and Zhou 2021; Jin, Netrapalli, and Jordan 2020; Lin, Jin, and Jordan 2020; Lin, Jin, and Jordan 2020; Lu et al. 2020; Ostrovskii, Lowy, and Razaviyayn 2021; Thekumparampil et al. 2019; Zhang, Aybat, and Gürbüzbalaban 2021; Yang et al. 2022). Moreover, the lower complexity bounds have also been studied for centralized minimax algorithms in (Zhang, Hong, and Zhang 2019; Zhang et al. 2021a; Li et al. 2021). Additionally, more methods employing *variance reduction* have been considered to improve the performance of the stochastic minimax algorithms, e.g., see (Xu et al. 2020; Huang, Wu, and Huang 2021; Luo et al. 2020; Zhang, Aybat, and Gurbuzbalaban 2022). In this paper, to control the noise accumulation, we propose DGDA-VR, a decentralized method employing the SPIDER *variance reduction* technique (Fang et al. 2018), a variant of SARAH (Nguyen et al. 2017a).

For the *decentralized setting*, we summarize some representative work for solving the minimax problem in Table 1. The method GT-DA (Tsaknakis, Hong, and Liu 2020) is proposed for a slightly modified version of (2) in the *deterministic* setting; this method only enforces consensus on $\mathbf{x}_i$ variables and as such requires the $\mathbf{y}$-subproblem to be solved to an increasing accuracy at each iteration. GT-SRVR (Zhang et al. 2021c) is closely related to DGDA-VR, our proposed

Method	P	U	Oracle Comp.	Comm. Comp.	Requirement
GT-DA (Tsaknakis, Hong, and Liu 2020)[†]	FS	D	$\tilde{\mathcal{O}}\left(\frac{n\kappa^{a_s}}{(1-\rho)^{b_s}\varepsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{\kappa^{a_c}}{(1-\rho)^{b_c}\varepsilon^2}\right)$	mult. $\mathbf{y}$-update
GT-SRVR (Zhang et al. 2021c)	FS	S	$\mathcal{O}\left(n+\frac{\sqrt{n}\kappa^{c_s}}{(1-\rho)^{d_s}\varepsilon^2}\right)^{\diamond}$	$\mathcal{O}\left(\frac{\kappa^{c_c}}{(1-\rho)^{d_c}\varepsilon^2}\right)$	✗
DSGDA (Gao 2022)	FS	S	$\mathcal{O}\left(\frac{\sqrt{n}L\kappa^3}{(1-\rho)^2\varepsilon^2}\right)^{\P}$	$\mathcal{O}\left(\frac{L\kappa^3}{(1-\rho)^2\varepsilon^2}\right)$	✗
DPOSG (Liu et al. 2020)	S	S	$\mathcal{O}\left(\frac{\sigma^2}{(1-\rho^t)^2\varepsilon^{12}}\right)^{\ddagger}$	$\mathcal{O}\left(\frac{\sigma^2}{(1-\rho^t)^2\varepsilon^{12}}\right)^{\ddagger}$	mult. comm.
DREAM (Chen, Ye, and Luo 2022)	S	S	$\mathcal{O}\left(\frac{L\kappa^3\sigma}{\varepsilon^3}\right)$	$\mathcal{O}\left(\frac{L\kappa^2}{\sqrt{1-\rho}\varepsilon^2}\right)$	mult. comm.
This Paper (`DGDA-VR`)	S	S	$\mathcal{O}\left(\frac{L\kappa^3\sigma}{\min\{1/\kappa,(1-\rho)^2\}\epsilon^3}\right)$	$\mathcal{O}\left(\frac{L\kappa^2}{\min\{1/\kappa,(1-\rho)^2\}\varepsilon^2}\right)$	✗
Lower Bounds[○]	S	S	$\Omega\left(L\sigma\epsilon^{-3}\right)$	$\Omega\left(\frac{L}{\sqrt{1-\rho}\epsilon^2}\right)$	✗

Table 1: The P column shows the problem setting: finite-sum (FS) or stochastic (S) –for FS setting, n denotes the number of component functions. The U column indicates whether stochastic (S) or deterministic (D) gradients are used. Some works do not explicitly state the dependence upon the spectral gap or condition number –we use constants $a, b, c, d > 0$ with subscripts s and c indicating that these unknowns are related to the *sample* or *communication* complexities, respectively. An ✗ in the final column indicates there is no special requirement for the theoretical results to hold. Table notes: [†]GT-DA considers a slightly different problem than (2) as consensus is only enforced on $\{\mathbf{x}_i\}$ or $\{\mathbf{y}_i\}$; additionally, GT-DA performs deterministic updates; hence, σ does not appear in the complexity results. [◇]GT-SRVR can remove the dependence upon σ by computing a full gradient periodically. [¶]DSGDA uses a variance reduction technique which removes the bounded variance assumption (hence σ does not appear); however, it is unclear whether this technique can be extended to the stochastic setting. [‡]DPOSG considers the nonconvex-*nonconcave* problem, hence κ is undefined for this setting; additionally, $t > 1$ represents the required number of communications per iteration. [○] (Arjevani et al. 2022) considers centralized nonconvex minimization problems defined by functions with Lipshitz gradients and assumes that their stochastic oracles are unbiased and have bounded variance. Similarly, (Sun, Lu, and Hong 2020) considers a deterministic distributed nonconvex minimization problem under the same conditions. The oracle complexity of distributed methods cannot be less than that of centralized methods, and the communication complexity of minimax problems cannot be less than that of minimization problems; therefore, their lower bounds apply here.

algorithm; that said, the analysis for GT-SRVR is only provided for the finite-sum problem, and the dependence upon important parameters such as κ and ρ is unclear. Similarly, DSGDA (Gao 2022) is proposed for the finite-sum setting, and employs a stochastic gradient estimator from (Li, Hanzely, and Richtárik 2021), for which it is unclear on how to theoretically extend to the general stochastic setting. For the purely stochastic case, DPSOG (Liu et al. 2020) is a general method that solves the nonconvex-nonconcave problem, however, its oracle complexity is sub-optimal. Furthermore, DPSOG requires multiple communications per iteration in order to guarantee the convergence to a stationary point.

Comparison with DM-HSGD and DREAM. We provide a detailed comparison of `DGDA-VR` to two closely related methods: DM-HSGD (Xian et al. 2021) and DREAM (Chen, Ye, and Luo 2022). The recent DM-HSGD (Xian et al. 2021) algorithm adapts the STORM-type update to the decentralized minimax setting; however, there are several critical errors in their proof which impact their results. First, their equation (28) does not hold with the given choice of θ. In fact, θ must depend on L, for which it is not clear whether their convergence analysis will go through if one chooses $\theta = \Theta(1/L)$ to make their equation (28) valid, e.g., in this scenario the coefficient of $\mathbf{E}\|\bar{u}^t\|$ becomes positive and cannot be dropped from the final bound while their convergence analysis requires this term to be dropped. Second, the algorithm is claimed to solve the minimax problem in (2) such that $\mathbf{y} \in \mathcal{Y}$ for a convex set $\mathcal{Y} \subseteq \mathbb{R}^m$; however, equation (29) in their Lemma 5 cannot hold unless $\mathcal{Y} \triangleq \mathbb{R}^m$ which means that at best, their analysis is only applicable to (2) without simple constraint sets. The recent DREAM (Chen, Ye, and Luo 2022) is similar to our method in terms of the variance reduction technique used to reduce the oracle complexity. However, their proof *requires* the use of multi-communication rounds, i.e., rather than using a mixing matrix W (satisfying Assumption 6), each iteration of DREAM uses W^K for $K = \mathcal{O}(\log(M)/(1-\rho))$ which exhibits the typical behavior of a *centralized* method.[1] Our proof technique removes such a requirement while ensuring the convergence of `DGDA-VR` for any connected network.

Preliminaries

Throughout the paper, for notational convenience, we define $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{d+m}$ to be the concatenation of the $\mathbf{x}$ and $\mathbf{y}$ variables. We start with some basic definitions.

Definition 1. A differentiable function r is L-smooth if $\exists L > 0$ such that $\|\nabla r(\mathbf{z}) - \nabla r(\mathbf{z}')\| \leq L\|\mathbf{z} - \mathbf{z}'\|, \forall \mathbf{z}, \mathbf{z}'$.

Since only stochastic estimates of $\{\nabla f_i\}$ are available to the agents, we introduce the concept of a stochastic oracle

[1] Indeed, for W satisfying Assumption 6, as $k \to \infty$, W^k converges *linearly* to the averaging matrix $\frac{1}{M}\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{M\times M}$.

and state our assumptions on the oracle below.

Definition 2. For all $i \in [M]$, given a random sample ξ, we define the stochastic oracle of $\nabla f_i(\mathbf{x}, \mathbf{y})$ at $(\mathbf{x}, \mathbf{y})$ to be $\tilde{\nabla} f_i(\mathbf{x}, \mathbf{y}; \xi)$. Additionally, given a set of random samples $\mathcal{B}$,

$$G_i(\mathcal{B}) \triangleq \frac{1}{|\mathcal{B}|} \sum_{\xi \in \mathcal{B}} \tilde{\nabla} f_i(\mathbf{x}_i, \mathbf{y}_i; \xi) \tag{3}$$

is the averaged stochastic estimator for $\nabla f_i(\mathbf{x}_i, \mathbf{y}_i)$ with random samples $\mathcal{B}$. $G_i^t(\mathcal{B})$ denotes (3) evaluated at $(\mathbf{x}_i^t, \mathbf{y}_i^t)$.

Below we state our assumptions on the functions $\{f_i\}_{i \in \mathcal{V}}$ and their stochastic gradient oracles and also, assumptions on the primal objective $\Phi(\cdot)$ and the mixing matrix W.

Assumption 1. There exists $L > 0$ such that $f_i : \mathbb{R}^{d+m} \to \mathbb{R}$ is L-smooth for all $i \in [M]$.

Assumption 2. There exists $\mu > 0$ such that $f_i(\mathbf{x}, \cdot)$ is μ-strongly concave for all fixed $\mathbf{x}$ and $i \in [M]$.

Remark 1. Assumptions 1 and 2 imply that f is L-smooth and $f(\mathbf{x}, \cdot)$ is μ-strongly concave for all $\mathbf{x}$.

Definition 3. The condition number of (1) is $\kappa \triangleq L/\mu$. The primal function is defined as $\Phi(\cdot) \triangleq \max_{\mathbf{y}} f(\cdot, \mathbf{y})$.

Assumption 3. Φ is lower bounded, i.e., $\inf_{\mathbf{x}} \Phi(\mathbf{x}) > -\infty$.

Assumptions 1, 2, and 3 are standard in the minimax literature, e.g., see (Li et al. 2021). For all $i \in [M]$, we make the following assumptions for the stochastic oracles $\tilde{\nabla} f_i(\mathbf{x}, \mathbf{y}; \xi)$ (see Definition 2).

Assumption 4. The stochastic gradients are unbiased and have finite variance. Namely, there exists $\sigma > 0$ such that for all $i \in [M]$ and for any $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{d+m}$, the stochastic gradient $\tilde{\nabla} f_i(\mathbf{z}; \xi)$ satisfies the conditions:

1. $\mathbf{E}\big[\tilde{\nabla} f_i(\mathbf{z}; \xi) \mid \mathbf{z}\big] = \nabla f_i(\mathbf{z})$;
2. $\mathbf{E}\big[\|\tilde{\nabla} f_i(\mathbf{z}; \xi) - \nabla f_i(\mathbf{z})\|^2 \mid \mathbf{z}\big] \leq \sigma^2$.

Assumption 4 is common in the literature, e.g., (Can, Gurbuzbalaban, and Aybat 2022; Fallah, Ozdaglar, and Pattathil 2020; Yang et al. 2022), and satisfied when gradients are estimated from randomly sampled data points with replacement. We also make the following assumption on f_i.

Assumption 5. Given random ξ, for any $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^{d+m}$, we assume $\mathbf{E}[\|\tilde{\nabla} f_i(\mathbf{z}; \xi) - \tilde{\nabla} f_i(\mathbf{z}'; \xi)\|^2] \leq L^2 \mathbf{E}[\|\mathbf{z} - \mathbf{z}'\|^2]$.

Indeed, Assumptions 4 and 5 imply Assumption 1 holds, see section 2.2 in (Tran-Dinh et al. 2022b). Finally, we state our assumptions on the mixing matrix $W \in \mathbb{R}^{M \times M}$.

Assumption 6. Consider a connected network $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of M agents and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. An ordered pair $(i, j) \in \mathcal{E}$ if agent i can directly communicate with agent j. Let $W \triangleq (w_{ij}) \in \mathbb{R}^{M \times M}$ be a matrix with non-negative entries such that

1. (Decentralized property) If $(i, j) \notin \mathcal{E}$, then $w_{ij} = 0$;
2. (Doubly stochastic property) $W\mathbf{1} = \mathbf{1}$ and $W^\top \mathbf{1} = \mathbf{1}$;
3. (Spectral property) $\rho \triangleq \|W - \Pi\|_2 \in [0, 1)$;

where $\Pi \triangleq \frac{1}{M}\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{M \times M}$ denotes the average operator.

Algorithm 1: DGDA-VR

```
Input: $Z^0$, $\{\eta_x, \eta_y\}$, $\{S_1, S_2, q, T\}$
for $t = 0, 1, 2, ..., T-1$ do
    $X^{t+1} = WX^t - \eta_x D_x^t$
    $Y^{t+1} = WY^t + \eta_y D_y^t$
    if $\mathrm{mod}(t, q) = 0$ then
        Let $\mathcal{C}_i^{t+1}$ be random samples with $|\mathcal{C}_i^{t+1}| = S_1$
        $\mathbf{v}_i^{t+1} = G_i^{t+1}(\mathcal{C}_i^{t+1}),\ \forall i \in [M]$
    else
        Let $\mathcal{B}_i^{t+1}$ be random samples with $|\mathcal{B}_i^{t+1}| = S_2$
        $\mathbf{v}_i^{t+1} = G_i^{t+1}(\mathcal{B}_i^{t+1}) - G_i^t(\mathcal{B}_i^{t+1}) + \mathbf{v}_i^t,\ \forall i \in [M]$
    end if
    $D_x^{t+1} = W(D_x^t + V_x^{t+1} - V_x^t)$
    $D_y^{t+1} = W(D_y^t + V_y^{t+1} - V_y^t)$
end for
Output:$(X^\tau, Y^\tau)$, where $\tau$ is selected from
    $\{0, \ldots, T-1\}$ uniformly at random
```

Notice that W is not assumed to be symmetric; hence, Assumption 6 covers both strongly-connected weight-balanced directed networks and undirected ones (Xin, Khan, and Kar 2021a). This is a weaker assumption compared to some related papers (Liu et al. 2020; Zhang et al. 2021b; Chen, Ye, and Luo 2022), which require a symmetric W and hence are only theoretically applicable to undirected networks.

Indeed, the main problem in (1) is equivalent to $\min_{\mathbf{x}} \Phi(\mathbf{x})$. Moreover, the norm of the gradient of the primal function $\Phi(\mathbf{x})$, i.e., $\|\nabla \Phi(\mathbf{x})\|$, is widely used as the convergence metric in the algorithmic analysis for nonconvex minimax problems in the literature. Given that we solve (2), we quantify the consensus errors among the agents related to the average point $\bar{\mathbf{z}} = (\bar{\mathbf{x}}, \bar{\mathbf{y}})$ and also $\|\nabla \Phi(\bar{\mathbf{x}})\|$.

DGDA-VR Method

We introduce our proposed Decentralized Gradient Decent Ascent - Variance Reduction, DGDA-VR, method in Algorithm 1 for solving (2). Specifically, through local computations and communicating with neighboring agents, each agent-i for $i \in [M]$ iteratively updates its local variable $\mathbf{z}_i \triangleq (\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^{d+m}$ – its value at iteration $t \in \mathbb{N}$ is denoted by $\mathbf{z}_i^t \triangleq (\mathbf{x}_i^t, \mathbf{y}_i^t) \in \mathbb{R}^{d+m}$. For notational convenience, we define the following terms.

Definition 4. $(X^t, Y^t), D^t, V^t \in \mathbb{R}^{M \times (d+m)}$ such that

$$X^t \triangleq [\mathbf{x}_1^t, \ldots, \mathbf{x}_M^t]^\top,\ Y^t \triangleq [\mathbf{y}_1^t, \ldots, \mathbf{y}_M^t]^\top,$$
$$V^t \triangleq [\mathbf{v}_1^t, \ldots, \mathbf{v}_M^t]^\top, D^t \triangleq [\mathbf{d}_1^t, \ldots, \mathbf{d}_M^t]^\top,$$

where (X^t, Y^t) denotes the iterates of DGDA-VR displayed in Algorithm 1, $\mathbf{d}_i^t = (\mathbf{d}_{x,i}^t, \mathbf{d}_{y,i}^t)$ denotes the gradient-tracking term, and $\mathbf{v}_i^t = (\mathbf{v}_{x,i}^t, \mathbf{v}_{y,i}^t)$ denotes the SPIDER-type stochastic gradient estimates of agent-i at iteration $t \in \mathbb{N}$. Let $Z^t = \big(X^t, Y^t\big) \in \mathbb{R}^{M \times (d+m)}$ for $t \in \mathbb{N}$.

Definition 5. For $t \geq 0$, given a matrix $X^t \in \mathbb{R}^{M \times d}$, we define $\bar{X}^t \triangleq \Pi(X^t)$, i.e., let

$$\bar{\mathbf{x}}^t \triangleq \frac{1}{M} \sum_{i=1}^{M} \mathbf{x}_i^t, \quad \bar{X}^t = \mathbf{1}\bar{\mathbf{x}}^{t\top}, \quad X_\perp \triangleq X^t - \bar{X}^t, \tag{4}$$

and $\{\bar{Y}^t, Y^t_\perp, \bar{Z}^t, Z^t_\perp, \bar{D}^t, D^t_\perp, \bar{V}^t, V^t_\perp\}$ is defined similarly.

Notice that under Assumption 6, Algorithm 1 implies that

$$\begin{aligned} \bar{\mathbf{x}}^{t+1} &= \bar{\mathbf{x}}^t - \eta_x \bar{\mathbf{d}}^t_x, & \bar{\mathbf{d}}^{t+1}_x &= \bar{\mathbf{d}}^t_x + \bar{\mathbf{v}}^{t+1}_x - \bar{\mathbf{v}}^t_x, \\ \bar{\mathbf{y}}^{t+1} &= \bar{\mathbf{y}}^t + \eta_y \bar{\mathbf{d}}^t_y, & \bar{\mathbf{d}}^{t+1}_y &= \bar{\mathbf{d}}^t_y + \bar{\mathbf{v}}^{t+1}_y - \bar{\mathbf{v}}^t_y, \end{aligned} \tag{5}$$

hold for all $t \geq 0$. Moreover, when $\bar{\mathbf{d}}^0 = \bar{\mathbf{v}}^0$, it holds that $\bar{\mathbf{d}}^t = \bar{\mathbf{v}}^t$ for $t \in \mathbb{N}$; thus, in such scenarios, we have

$$\bar{\mathbf{x}}^{t+1} = \bar{\mathbf{x}}^t - \eta_x \bar{\mathbf{v}}^t_x, \quad \bar{\mathbf{y}}^{t+1} = \bar{\mathbf{y}}^t + \eta_y \bar{\mathbf{v}}^t_y, \quad \forall\, t \in \mathbb{N}. \tag{6}$$

Main Results

In a multi-agent system, for any $\epsilon > 0$, our aim for each agent-i is to compute $\mathbf{z}_i(\epsilon) = \Big(\mathbf{x}_i(\epsilon), \mathbf{y}_i(\epsilon)\Big)$ such that

$$\mathbf{E}\Big[\|\nabla\Phi(\bar{\mathbf{x}}_\epsilon)\|\Big] \leq \epsilon, \tag{7a}$$

$$\mathbf{E}\Big[\|\bar{\mathbf{y}}_\epsilon - \mathbf{y}^*(\bar{\mathbf{x}}_\epsilon)\|^2\Big] = \mathcal{O}(\epsilon^2), \tag{7b}$$

$$\mathbf{E}\Big[\sum_{i=1}^{M} \|\mathbf{z}_i(\epsilon) - \bar{\mathbf{z}}_\epsilon\|^2\Big] = \mathcal{O}(\epsilon^2), \tag{7c}$$

where $\bar{\mathbf{z}}_\epsilon = (\bar{\mathbf{x}}_\epsilon, \bar{\mathbf{y}}_\epsilon) \triangleq \frac{1}{M}\sum_{i=1}^M \big(\mathbf{x}_i(\epsilon), \mathbf{y}_i(\epsilon)\big)$ and

$$\mathbf{y}^*(\cdot) \triangleq \operatorname*{argmax}_{\mathbf{y}} f(\cdot, \mathbf{y}) \tag{8}$$

denotes the best-response function. We show that DGDA-VR can indeed generate $\{\mathbf{z}_i(\epsilon)\}_{i\in\mathcal{V}}$ such that (7) holds. More importantly, in the decentralized optimization context, let T_ϵ denote the minimum number of communication rounds required to compute $\{\mathbf{x}_i(\epsilon)\}_{i\in\mathcal{V}}$ satisfying (7) in a decentralized manner — in each communication round, each agent-i transmits two vectors of size $(d+m)$ to its neighbors, i.e., $\mathbf{z}^t_i$ and $\mathbf{d}^t_i$. According to DGDA-VR, T_ϵ communication rounds require each agent-i to make $C_\epsilon \triangleq \lceil \frac{T_\epsilon}{q} \rceil S_1 + T_\epsilon S_2$ calls to its stochastic oracle $\tilde{\nabla} f_i$. Our aim is to provide bounds on the expected communication and oracle complexities, i.e., T_ϵ and C_ϵ. Moreover, we will provide precise bounds on the dual suboptimality as in (7b) and on the consensus violation (the deviation from the average) for $\{\mathbf{z}_i(\epsilon)\}_{i\in\mathcal{V}}$ as in (7c). The result below shows our guarantee on (7a).

Theorem 1. *Suppose Assumptions 1-6 hold, and* $\{\eta_x, \eta_y\}$ *and* $\{S_1, S_2, q\}$ *are chosen such that*

$$\begin{aligned} &\eta_y = \Theta\left(\frac{1}{L}\min\left\{(1-\rho)^2, \frac{1}{\kappa}\right\}\right),\ \eta_x = \Theta\left(\frac{\eta_y}{\kappa^2}\right), \\ &S_1 = \Theta\left(\frac{\kappa^2\sigma^2}{\epsilon^2}\right),\ S_2 \geq q, \quad q \geq 1. \end{aligned} \tag{9}$$

Given $\epsilon > 0$, *there exists* $T_\epsilon \in \mathbb{N}$ *such that*

$$T_\epsilon = \mathcal{O}\Big(\max\left\{\frac{1}{\eta_x}, \frac{L\kappa}{\eta_y}, \frac{L^2\kappa^2}{(1-\rho)M}\right\}\epsilon^{-2}\Big),$$

and $\{X^t\}_{t=0}^T$ *generated by* DGDA−VR *satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbf{E}\Big[\|\nabla\Phi(\bar{\mathbf{x}}^t)\|\Big] \leq \epsilon, \quad \forall\, T \geq T_\epsilon, \tag{10}$$

where $\bar{\mathbf{x}}^t$ *is defined in* (4).

Remark 2. Without loss of generality, $L \geq 1$. Indeed, Assumption 5 holds for all $\hat{L}$ such that $\hat{L} \geq L$; therefore,

$$T_\epsilon = \mathcal{O}\Big(\frac{\max\{L\kappa^2, L^2\kappa\}}{\min\{1/\kappa, (1-\rho)^2\}} \cdot \frac{1}{\epsilon^2}\Big).$$

Given $T = T_\epsilon, S_1$ and $S_2 \geq q$, the optimal $q = \Theta(\sqrt{S_1}) = \Theta\Big(\frac{\kappa\sigma}{\epsilon}\Big)$ and $S_2 = \Theta(q)$, so $TS_2 + TS_1/q \sim 2TS_2 = \mathcal{O}\left(\frac{\max\{L\kappa^3, L^2\kappa^2\}\sigma}{\min\{1/\kappa, (1-\rho)^2\}\epsilon^3}\right)$.

The result below shows our guarantee on (7b) and (7c).

Theorem 2. *Under the premise of Theorem 1,*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|Z^t_\perp\|^2_F] = \mathcal{O}\Big(M\epsilon^2/(L^2\kappa^2)\Big), \quad \forall\, T \geq T_\epsilon, \tag{11}$$

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\mathbf{y}^t - \mathbf{y}^*(\bar{\mathbf{x}}^t)\|^2] = \mathcal{O}(\epsilon^2/L^2), \quad \forall\, T \geq T_\epsilon, \tag{12}$$

where $\Lambda_0 \triangleq \max\{\|Z^0_\perp\|^2_F, \|D^0_\perp\|^2_F\}$.

Remark 3. Let τ_ϵ be a random variable with a uniform distribution over $\{0, \ldots, T_\epsilon - 1\}$. Then (10) implies that $\mathbf{E}\Big[\|\nabla\Phi(\bar{\mathbf{x}}^{\tau_\epsilon})\|\Big] \leq \epsilon$. Furthermore, we also have $\mathbb{E}[\|Z^{\tau_\epsilon}_\perp\|_F] = \mathcal{O}(\epsilon)$ and $\mathbb{E}[\|\mathbf{y}^{\tau_\epsilon} - \mathbf{y}^*(\bar{\mathbf{x}}^{\tau_\epsilon})\|] = \mathcal{O}(\epsilon)$.

Remark 4. Since the final complexity bound depends on the choice of $\eta_x, \eta_y, S_1, S_2, q$, we evaluate the tightness of our results by comparing these parameters with those in related work. Our selection of the VR parameters $S_1 = \mathcal{O}(\kappa^2\epsilon^{-2}), S_2 = \mathcal{O}(\kappa\epsilon^{-1}), q = \mathcal{O}(\kappa\epsilon^{-1})$ is consistent with the optimal choice in single-loop centralized VR methods, e.g., (Luo et al. 2020). The time-scale ratio $\eta_y/\eta_x = \kappa^2$ aligns with the ratios used in existing works on GDA methods (Lin, Jin, and Jordan 2020). To adapt the GDA to the decentralized setting, we have introduced a factor of $\frac{1}{\kappa}$ into the selection of η_y. DREAM can set $\eta_y = \frac{1}{L}$ but requires multi-communication rounds. It is not yet clear if this cost can be further reduced, and whether $\frac{1}{\kappa}$ represents the optimal adjustment – nevertheless, our analysis seems to be tight when compared to the existing results.

Numerical Experiments

We test our proposed method on three problems: a quadratic minimax problem, robust non-convex linear regression, and robust neural network training. For the first and third problem, we let $M = 8$ such that each agent is represented by an NVIDIA Tesla V100 GPU. For the second problem, we test methods in a serial manner to facilitate more general reproducibility; here, we let $M = 20$. In all cases, we use a ring (cycle) graph with equal weights on edges including self loops, i.e., $w_{i,i-1} = w_{i,i} = w_{i,i+1} = 1/3$ for all $i \in [M]$. The learning rates for all tests are chosen such that $\eta_y \in \{10^{-1}, 10^{-2}, 10^{-3}\}$ and we tune the ratio $\frac{\eta_x}{\eta_y} \in \{1, 10^{-1}, 10^{-2}, 10^{-3}\}$. We test our proposed method against 3 methods: DPSOG (Liu et al. 2020), DM-HSGD (Xian et al. 2021), and the deterministic GT/DA (Tsaknakis, Hong, and Liu 2020). The code is made available at https://github.com/gmancino/DGDA-VR.

A Polyak-Lojasiewicz game

We consider a slightly modified version of the two-player Polyak-Lojasiewicz game from (Chen, Yao, and Luo 2022). Namely, we make the problem decentralized by letting each agent $i \in [M]$ contain a dataset of triples

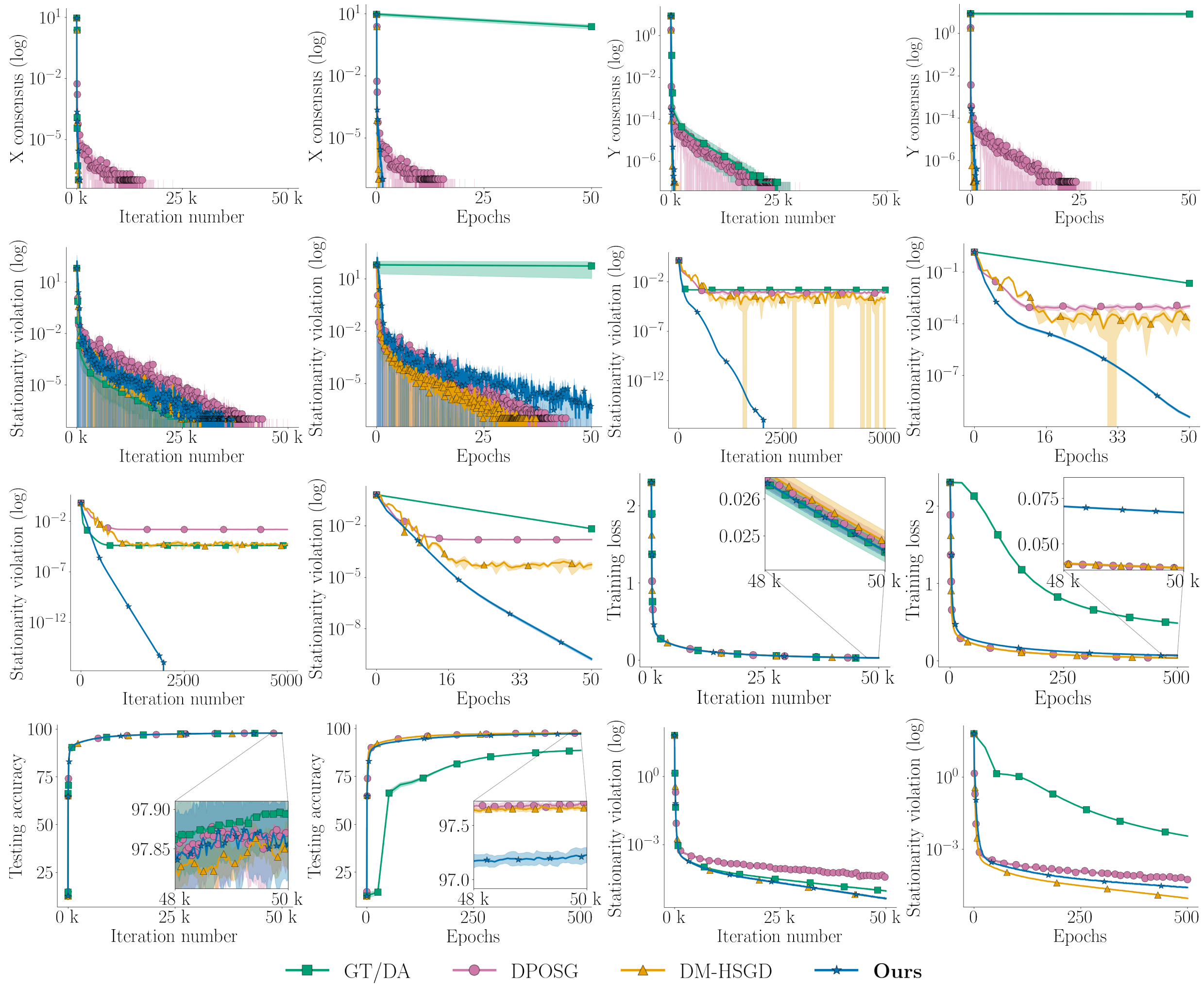

Figure 1: Pictures 1-6 are for the PL game (13). Pictures 7-10 for the robust non-convex linear regression model (15); the first two correspond to the a9a dataset, while the last two correspond to the ijcnn1 dataset. Pictures 11-16 for the robust neural network training problem (16). The arrangement of these pictures follows a left-to-right, then top-to-bottom order.

$\{(\mathbf{p}_{ij}, \mathbf{q}_{ij}, \mathbf{r}_{ij})\}_{j=1}^n$ where each vector lies in $\mathbb{R}^d$. For all $i \in [M]$, let $f_i : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ such that

$$f_i(\mathbf{x}_i, \mathbf{y}_i) = \frac{1}{2}(\mathbf{x}_i)^\top \mathbf{P}_i \mathbf{x}_i - \frac{1}{2}(\mathbf{y}_i)^\top \mathbf{Q}_i \mathbf{y}_i + (\mathbf{x}_i)^\top \mathbf{R}_i \mathbf{y}_i, \quad (13)$$

where $\mathbf{P}_i = \frac{1}{n}\sum_{j=1}^n \mathbf{p}_{ij}\mathbf{p}_{ij}^\top$, $\mathbf{Q}_i = \frac{1}{n}\sum_{j=1}^n \mathbf{q}_{ij}\mathbf{q}_{ij}^\top + \alpha\mathbf{I}$, $\mathbf{R}_i = \frac{1}{n}\sum_{j=1}^n \mathbf{r}_{ij}\mathbf{r}_{ij}^\top$ for some $\alpha > 0$ which guarantees the problem is strongly-concave in $\mathbf{y}$; we choose $\alpha = 1$ for these experiments. Data is generated in the same manner as in (Chen, Yao, and Luo 2022)[2] to guarantee that $\mathbf{P}_i$ is singular; hence, the problem is not strongly-convex in $\mathbf{x}$. Here, $n = 1{,}000$ and we fix the mini-batch size for all methods to be 1 (besides GT/DA). For our proposed method, we set $q = S_1 = 100$. We run each algorithm for 50,000 iterations and plot the results of 50 epochs (one pass over the whole dataset through sampling is an epoch) for each method. We measure the stationarity violation as $\|\sum_{i=1}^M \nabla_{\mathbf{x}} f_i(\bar{\mathbf{x}}, \mathbf{y}^{(*)})\|_2^2 + \|\mathbf{X}_\perp\|_F^2 + \|\mathbf{Y}_\perp\|_F^2$, where $\mathbf{y}^{(*)} \triangleq \arg\max_{\mathbf{y}} \sum_{i=1}^M f_i(\bar{\mathbf{x}}, \mathbf{y})$ for $\bar{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^M \mathbf{x}_i$. Results shown in Figure 1 demonstrate that DGDA-VR is competitive against SOTA for computing a stationary point.

Sensitivity Analysis To assess the influence of *graph connectivity*, we compared DGDA-VR against DM-HSGD on random connected graphs, generated such that there is an edge between any two nodes with probability $p \in \{0.05, 0.95\}$ — corresponding to low and high connectivity scenarios, respectively. For each p, we generate 15 random graphs of size $M \in \{8, 20\}$ – the average value of ρ over 15 realizations is 0.94, 0.97, 0.16, 0.1 for (p, M) combinations $(0.05, 8)$, $(0.05, 20)$, $(0.95, 8)$, $(0.95, 20)$, respectively. The first two plots in Fig. 2 report the sum of squared norms of the $\mathbf{x}$ and $\mathbf{y}$ consensus violations against the oracle complexity. In addition, we generate 15 random graphs for $M = 8$

[2] See https://github.com/TrueNobility303/SPIDER-GDA/blob/main/code/GDA/pl_data_generator.m

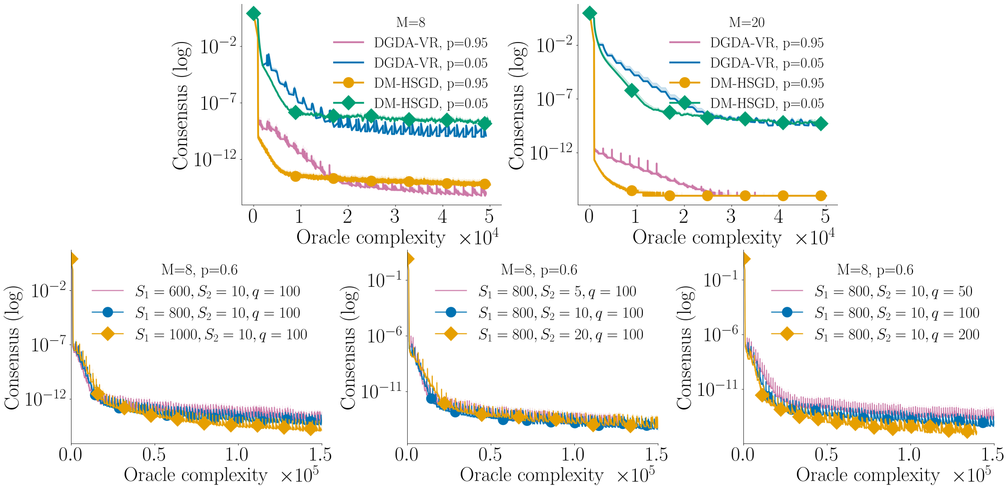

Figure 2: Sensitivity analysis results for the PL game (13). The first two plots show the sensitivity analysis in terms of graph connectivity ρ, while the last three show the sensitivity analysis in terms of the batchsizes S_1, S_2 and the frequency q. Here, oracle complexity refers to data points visited.

and $p = 0.6$, i.e., moderate connectivity with $\rho \approx 0.63$, to test DGDA-VR using low, moderate, high levels for each parameter S_1, S_2, q while fixing the other two at the moderate level. Results are reported in the last three plots of Fig. 2 which show that our method is not sensitive to the choice of hyper-parameters S_1, S_2, q.

Robust Machine Learning

We consider two robust machine learning problems: nonconvex linear regression with tabular data and neural network training with image data. Let each agent $i \in [M]$ contain a dataset of points and labels denoted by $\{(\mathbf{a}_{ij}, b_{ij})\}_{j=1}^n$ where b_{ij} is the class label of data point $\mathbf{a}_{ij}$. For these problems, $\mathbf{y}_i^{(*)} \triangleq \arg\max_{\mathbf{y}_i} f_i(\mathbf{x}, \mathbf{y}_i)$ is not easily computable; as a proxy, we report the stationarity violation using

$$\|\sum_{i=1}^M \nabla f_i(\bar{\mathbf{x}}, \bar{\mathbf{y}})\|_2^2 + \|\mathbf{X}_\perp\|_F^2 + \|\mathbf{Y}_\perp\|_F^2. \qquad (14)$$

Robust Non-convex Linear Regression We consider training a robust version of the non-convex linear regression classifier from (Sun, Lu, and Hong 2020). For $i \in [M]$, let

$$f_i(\mathbf{x}_i, \mathbf{y}_i) = \frac{1}{n}\sum_{j=1}^n \ln\left(\left(b_{ij} - \mathbf{x}_i^\top(\mathbf{a}_{ij} + \mathbf{y}_i)\right)^2/2 + 1\right) - \frac{\alpha}{2}\|\mathbf{y}_i\|_2^2 \qquad (15)$$

where $b_{ij} \in \{-1, +1\}$ and $\alpha > 0$ is a penalty term which guarantees that f_i is strongly-concave in $\mathbf{y}_i$ –we set $\alpha = 1$ for these experiments. The $\mathbf{y}$ variable acts as a perturbation to the data; hence, we seek to minimize the loss on the worst-case data perturbation. We test DGDA-VR on two datasets: a9a and ijcnn1[3]. We fix the mini-batch to be 32 for all methods beside GT/DA and set $S_1 = 1{,}000, q = 32$ for our method. We run each method for 5,000 iterations and plot the results of 50 epochs for each method. Results shown in Figure 1 demonstrate that in contrast to DGDA-VR, the main bottleneck for other methods is to achieve consensus among agents.

Robust Neural Network Training We consider a slightly modified version of the robust neural network training problem from (Deng and Mahdavi 2021; Sharma et al. 2022). For all $i \in [M]$, let

$$f_i(\mathbf{x}_i, \mathbf{y}_i) = \frac{1}{n}\sum_{j=1}^n \ell\left(g_{\mathbf{x}_i}(\mathbf{a}_{ij} + \mathbf{y}_i), b_{ij}\right) - \frac{\alpha}{2}\|\mathbf{y}_i\|_2^2, \qquad (16)$$

where $g_{\mathbf{x}_i}$ is a neural network parameterized by $\mathbf{x}_i$, ℓ is the cross-entropy loss function, and $\alpha > 0$ is a penalty parameter which guarantees that f_i is strongly-concave in $\mathbf{y}_i$ –we set $\alpha = 1$ for these experiments. Inspired by (Deng and Mahdavi 2021), we adopt $g_{\mathbf{x}_i}$ corresponding to a two-layer network (200 hidden units) with a tanh activation function, and we use the MNIST (LeCun 1998) dataset for training. We fix the mini-batch size for all methods to be 100 (besides GT/DA). For DGDA-VR, we set $q = 100$ and $S_1 = 7{,}500$. We run each algorithm to 50,000 iterations and plot the results of 500 epochs for each method. Results shown in Figure 1 verify that DGDA-VR is competitive against the stochastic methods and still outperforms the deterministic method in terms of data passes required to compute a near stationary point.

Conclusion

In this work, we proposed a Decentralized Gradient Decent Ascent - Variance Reduction method, DGDA-VR, for solving the stochastic nonconvex strongly-concave minimax problem over a connected network of M computing agents. Under the assumption that the computing agents only have access to stochastic first-order oracles, our method incorporates variance reduction and gradient tracking to jointly optimize the sample and communication complexities to be $\mathcal{O}\left(\epsilon^{-3}\right)$ and $\mathcal{O}\left(\epsilon^{-2}\right)$, respectively, for reaching an ϵ-accurate solution. For the class of problems considered here, this is the first work which does not require multiple coordinated communications in each iteration to achieve these optimal complexities.

[3] See: https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/

Acknowledgements

This work is partly supported by NSF Grant DMS-2208394 and the ONR grants N00014-21-1-2271 and N00014-22-1-2573, and also by the Rensselaer-IBM AI Research Collaboration, part of the IBM AI Horizons Network.

References

Arjevani, Y.; Carmon, Y.; Duchi, J. C.; Foster, D. J.; Srebro, N.; and Woodworth, B. 2022. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 1–50.

Can, B.; Gurbuzbalaban, M.; and Aybat, N. 2022. A Variance-Reduced Stochastic Accelerated Primal Dual Algorithm. *arXiv e-prints*, arXiv:2202.09688.

Chen, L.; Yao, B.; and Luo, L. 2022. Faster Stochastic Algorithms for Minimax Optimization under Polyak-Lojasiewicz Condition. In *36th NeurIPS*.

Chen, L.; Ye, H.; and Luo, L. 2022. A Simple and Efficient Stochastic Algorithm for Decentralized Nonconvex-Strongly-Concave Minimax Optimization. *arXiv preprint arXiv:2212.02387*.

Chen, Z.; Ma, S.; and Zhou, Y. 2021. Accelerated Proximal Alternating Gradient-Descent-Ascent for Nonconvex Minimax Machine Learning. *arXiv preprint arXiv:2112.11663*.

Cutkosky, A.; and Orabona, F. 2019. Momentum-Based Variance Reduction in Non-Convex SGD. In *33th NeurIPS*.

Deng, Y.; and Mahdavi, M. 2021. Local Stochastic Gradient Descent Ascent: Convergence Analysis and Communication Efficiency. In *AISTATS-24*.

Fallah, A.; Ozdaglar, A.; and Pattathil, S. 2020. An optimal multistage stochastic gradient method for minimax problems. In *59th IEEE CDC*.

Fang, C.; Li, C. J.; Lin, Z.; and Zhang, T. 2018. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *32th NeurIPS*.

Gao, H. 2022. Decentralized Stochastic Gradient Descent Ascent for Finite-Sum Minimax Problems. *arXiv preprint arXiv:2212.02724*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Nets. In *27th NeurIPS*.

Huang, F.; Wu, X.; and Huang, H. 2021. Efficient mirror descent ascent methods for nonsmooth minimax problems. *35th NeurIPS*.

Jin, C.; Netrapalli, P.; and Jordan, M. 2020. What is local optimality in nonconvex-nonconcave minimax optimization? In *ICML*, 4880–4889. PMLR.

Koloskova, A.; Lin, T.; and Stich, S. U. 2021. An Improved Analysis of Gradient Tracking for Decentralized Machine Learning. In *35th NeurIPS*.

LeCun, Y. 1998. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Li, B.; Li, Z.; and Chi, Y. 2022. DESTRESS: Computation-Optimal and Communication-Efficient Decentralized Nonconvex Finite-Sum Optimization. *SIAM Journal on Mathematics of Data Science*, 4(3): 1031–1051.

Li, H.; Tian, Y.; Zhang, J.; and Jadbabaie, A. 2021. Complexity Lower Bounds for Nonconvex-Strongly-Concave Min-Max Optimization. *arXiv preprint arXiv:2104.08708*.

Li, Z.; Hanzely, S.; and Richtárik, P. 2021. ZeroSARAH: Efficient Nonconvex Finite-Sum Optimization with Zero Full Gradient Computation. *arXiv preprint arXiv:2103.01447*.

Lian, X.; Zhang, C.; Zhang, H.; Hsieh, C.-J.; Zhang, W.; and Liu, J. 2017. Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent. In *31th NeurIPS*.

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

Lin, T.; Jin, C.; and Jordan, M. I. 2020. Near-Optimal Algorithms for Minimax Optimization. *arXiv e-prints*, arXiv:2002.02417.

Liu, M.; Zhang, W.; Mroueh, Y.; Cui, X.; Ross, J.; Yang, T.; and Das, P. 2020. A Decentralized Parallel Algorithm for Training Generative Adversarial Nets. In *34th NeurIPS*.

Lu, S.; Tsaknakis, I.; Hong, M.; and Chen, Y. 2020. Hybrid block successive approximation for one-sided non-convex min-max problems: algorithms and applications. *IEEE Transactions on Signal Processing*, 68: 3676–3691.

Lu, S.; Zhang, X.; Sun, H.; and Hong, M. 2019. GNSD: a Gradient-Tracking Based Nonconvex Stochastic Algorithm for Decentralized Optimization. In *2019 IEEE DSW*.

Luo, L.; Ye, H.; Huang, Z.; and Zhang, T. 2020. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. In *34th NeurIPS*.

Mancino-Ball, G.; Miao, S.; Xu, Y.; and Chen, J. 2023. Proximal stochastic recursive momentum methods for nonconvex composite decentralized optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 9055–9063.

Namkoong, H.; and Duchi, J. C. 2016. Stochastic Gradient Methods for Distributionally Robust Optimization with f-divergences. In *30th NeurIPS*.

Nguyen, L. M.; Liu, J.; Scheinberg, K.; and Takáč, M. 2017a. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *ICML*, 2613–2621. PMLR.

Nguyen, L. M.; Liu, J.; Scheinberg, K.; and Takáč, M. 2017b. SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient. In Precup, D.; and Teh, Y. W., eds., *ICML*, volume 70 of *PMLR*, 2613–2621. International Convention Centre, Sydney, Australia: PMLR.

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 *33th NeurIPS*.

Ostrovskii, D. M.; Lowy, A.; and Razaviyayn, M. 2021. Efficient search of first-order nash equilibria in nonconvex-concave smooth min-max problems. *SIOPT*, 31(4): 2508–2538.

Pan, T.; Liu, J.; and Wang, J. 2020. D-SPIDER-SFO: A Decentralized Optimization Algorithm with Faster Convergence Rate for Nonconvex Problems. In *AAAI-20*.

Sharma, P.; Panda, R.; Joshi, G.; and Varshney, P. 2022. Federated Minimax Optimization: Improved Convergence Analyses and Algorithms. In *ICML*, 19683–19730. PMLR.

Sun, H.; and Hong, M. 2019. Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms. *IEEE Transactions on Signal processing*, 67(22): 5912–5928.

Sun, H.; Lu, S.; and Hong, M. 2020. Improving the Sample and Communication Complexity for Decentralized Non-Convex Optimization: Joint Gradient Estimation and Tracking. In *ICML*, 9217–9228. PMLR.

Tang, H.; Lian, X.; Yan, M.; Zhang, C.; and Liu, J. 2018. D^2: Decentralized Training over Decentralized Data. In *ICML*, 4848–4856. PMLR.

Thekumparampil, K. K.; Jain, P.; Netrapalli, P.; and Oh, S. 2019. Efficient algorithms for smooth minimax optimization. *arXiv preprint arXiv:1907.01543*.

Tran-Dinh, Q.; Pham, N. H.; Phan, D. T.; and Nguyen, L. M. 2022a. A hybrid stochastic optimization framework for composite nonconvex optimization. *Mathematical Programming*, 191(2): 1005–1071.

Tran-Dinh, Q.; Pham, N. H.; Phan, D. T.; and Nguyen, L. M. 2022b. A hybrid stochastic optimization framework for composite nonconvex optimization. *Mathematical Programming*, 191(2): 1005–1071.

Tsaknakis, I.; Hong, M.; and Liu, S. 2020. Decentralized Min-Max Optimization: Formulations, Algorithms and Applications in Network Poisoning Attack. In *IEEE ICASSP 2020*, 5755–5759.

Verbraeken, J.; Wolting, M.; Katzy, J.; Kloppenburg, J.; Verbelen, T.; and Rellermeyer, J. S. 2020. A Survey on Distributed Machine Learning. *ACM Comput. Surv.*, 53(2).

Wang, Z.; Ji, K.; Zhou, Y.; Liang, Y.; and Tarokh, V. 2019. SpiderBoost and Momentum: Faster Variance Reduction Algorithms. In *33th NeurIPS*. Curran Associates, Inc.

Xian, W.; Huang, F.; Zhang, Y.; and Huang, H. 2021. A Faster Decentralized Algorithm for Nonconvex Minimax Problems. In *35th NeurIPS*.

Xin, R.; Das, S.; Khan, U. A.; and Kar, S. 2021. A Stochastic Proximal Gradient Framework for Decentralized Non-Convex Composite Optimization: Topology-Independent Sample Complexity and Communication Efficiency. *arXiv preprint, arXiv:2110.01594*.

Xin, R.; Khan, U.; and Kar, S. 2021a. A Hybrid Variance-Reduced Method for Decentralized Stochastic Non-Convex Optimization. In *ICML*, 11459–11469. PMLR.

Xin, R.; Khan, U. A.; and Kar, S. 2021b. An Improved Convergence Analysis for Decentralized Online Stochastic Non-Convex Optimization. *IEEE Transactions on Signal Processing*, 69: 1842–1858.

Xin, R.; Khan, U. A.; and Kar, S. 2022. Fast Decentralized Nonconvex Finite-Sum Optimization with Recursive Variance Reduction. *SIOPT*, 32(1): 1–28.

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

Xu, Y.; and Xu, Y. 2023. Momentum-Based Variance-Reduced Proximal Stochastic Gradient Method for Composite Nonconvex Stochastic Optimization. *Journal of Optimization Theory and Applications*, 196(1): 266–297.

Yang, J.; Orvieto, A.; Lucchi, A.; and He, N. 2022. Faster single-loop algorithms for minimax optimization without strong concavity. In *AISTATS*, 5485–5517. PMLR.

Zhang, J.; Hong, M.; and Zhang, S. 2019. On Lower Iteration Complexity Bounds for the Saddle Point Problems. *arXiv preprint arXiv:1912.07481*.

Zhang, J.; and You, K. 2020. Decentralized Stochastic Gradient Tracking for Non-convex Empirical Risk Minimization. *arXiv preprint arXiv:1909.02712*.

Zhang, S.; Yang, J.; Guzmán, C.; Kiyavash, N.; and He, N. 2021a. The complexity of nonconvex-strongly-concave minimax optimization. In *Uncertainty in Artificial Intelligence*, 482–492. PMLR.

Zhang, X.; Aybat, N.; and Gürbüzbalaban, M. 2021. Robust Accelerated Primal-Dual Methods for Computing Saddle Points. *arXiv preprint arXiv:2111.12743*.

Zhang, X.; Aybat, N.; and Gurbuzbalaban, M. 2022. SAPD+: An Accelerated Stochastic Method for Nonconvex-Concave Minimax Problems. In *36th NeurIPS*.

Zhang, X.; Liu, J.; Zhu, Z.; and Bentley, E. S. 2021b. GT-STORM: Taming Sample, Communication, and Memory Complexities in Decentralized Non-Convex Learning. *ACM Proceedings of MobiHoc*.

Zhang, X.; Liu, Z.; Liu, J.; Zhu, Z.; and Lu, S. 2021c. Taming Communication and Sample Complexities in Decentralized Policy Evaluation for Cooperative Multi-Agent Reinforcement Learning. In *35th NeurIPS*.