Decentralized Local Stochastic Extra-Gradient for Variational Inequalities

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider distributed stochastic variational inequalities (VIs) on unbounded domains with the problem data being heterogeneous (non-IID) and distributed across many devices. We make a very general assumption on the computational network that, in particular, covers the settings of fully decentralized calculations with time-varying networks and centralized topologies commonly used in Federated Learning. Moreover, multiple local updates on the workers can be made for reducing the communication frequency between workers. We extend the stochastic extragradient method to this very general setting and theoretically analyze its convergence rate in the strongly monotone, monotone, and non-monotone setting when an Minty solution exists. The provided rates have explicit dependence on network characteristics and how it varies with time, data heterogeneity, variance, number of devices, and other standard parameters. As a special case, our method and analysis apply to distributed stochastic saddle-point problems (SPP), e.g., to training Deep Generative Adversarial Networks (GANs) for which the decentralized training has been reported to be extremely challenging. In experiments for decentralized training of GANs we demonstrate the effectiveness of our proposed approach.

1 Introduction

In large scale machine learning (ML) scenarios the training data is often is split over many client devices, such as e.g. geo-distributed datacenters or mobile devices [33]. Decentralized training methods can train a ML model to the same accuracy as if all data would be aggregated on one single server [48, 3]. Training in a non-centralized fashion can offer many advantages over traditional centralized approaches in core aspects such as data ownership, privacy, fault tolerance and scalability. A particular instance of the decentralized learning setting is Federated Learning (FL), where the training is orchestrated by a single entity that communicates with all participating clients [55, 33]. In contrast, in fully decentralized learning (FD) scenarios the devices only communicate with their neighbours in the network topology [48]. Such algorithms are important in scenarios where centralized communication is expensive or impossible.

There have been tremendous advances recently in the development, design and understanding of decentralized training schemes [64, 85, 76, 48, 74, 82, 80, 84]. In particular, aspects such as data-heterogeneity [80, 71, 49], communication efficiency (through local updates [46, 39] or compression [79, 40]), or personalization [83, 6] have been studied recently. However, all these methods have been developed for single objective loss functions (minimization objective) and are not applicable to more general problem classes. For example, the training of Generative Adversarial Networks (GANs) [23] requires the joint optimization of the generator and discriminator objective, i.e. solving a non-convex non-concave saddle-point problem (SPP). This problem structure makes

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

GANs notoriously difficult to train in the single machine setting [22, 13, 14] and in particular over decentralized data [52, 62, 73].

In this paper, we present a novel algorithm for solving decentralized SPP, and more generally, decentralized stochastic Minty variational inequalities (MVI) [57, 32]. In a decentralized stochastic MVI, the data is distributed over $M \geq 1$ devices and each device $m \in [M]$ has access to its local stochastic oracle $F_m(z, \xi_m)$ for the operator $F_m(z) := \mathbb{E}_{\xi_m \sim \mathcal{D}_m} F_m(z, \xi_m)$. The data ξ_m follows unknown distributions $\mathcal{D}_m$, different at every node $m \in [M]$. The devices are connected via a network given as a communication graph and two nodes can exchange information only if they are connected by an edge in this graph. The goal is, respecting the communication constraints, to find cooperatively a point $z^* \in \mathbb{R}^n$ such that, for all $\forall z \in \mathbb{R}^n$,

$$\tfrac{1}{M} \textstyle\sum_{m=1}^{M} \langle \mathbb{E}_{\xi_m \sim \mathcal{D}_m} F_m(z, \xi_m), z^* - z \rangle \leq 0. \tag{1}$$

A special instance of a decentralized MVI is the decentralized SPP with local objectives $f_m(x, y) := \mathbb{E}_{\xi_m \sim \mathcal{D}_m}[f_m(x, y, \xi_m)]$:

$$\min_{x \in \mathbb{R}^{n_x}} \max_{y \in \mathbb{R}^{n_y}} \left[f(x, y) := \tfrac{1}{M} \textstyle\sum_{m=1}^{M} f_m(x, y) \right]. \tag{2}$$

This can be seen by considering $z = \left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ and the gradient field $F_m(z) = \left[\begin{smallmatrix} \nabla_x f_m(x, y) \\ -\nabla_y f_m(x, y) \end{smallmatrix}\right]$. In the special case when $f(x, y)$ is convex-concave, the corresponding operator $F(z) = \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{\xi_m} F_m(z, \xi_m)$ is monotone. However, in the context of GANs—where x and y are the parameters of the generator and discriminator, respectively—in general the local $f_m(x, y)$ are possibly non-convex, non-concave in x, y and we cannot assume monotonicity of F in general [see also 18].

In this paper, we develop a novel algorithm for problems (1) or (2). Gradient descent-ascent scheme on objective (2) may diverge even in the simple convex-concave setting and $M = 1$ worker [13]. Thus, unlike [52], we use extragradient updates [43, 32, 22] as a building block and combine it with gossip-type communication protocol [88, 10] on arbitrary, possibly time-varying, network topologies. One of the main challenges due to communication constraints is a "network error" induced by impossibility of all the nodes to reach the exact consensus, i.e. to have the same information on the current iterate of an algorithm. Thus, each device stores a local variable and only partial consensus among nodes can be achieved by gossip steps [42]. Unlike other decentralized algorithms [74, 52] our method avoids multiple gossip steps per iteration that leads to better practical performance and possibility to work on time-varying networks. Moreover, our method allows multiple local updates between communication rounds making it suitable for communication- and privacy-restricted FL or fully decentralized settings [90].

Our contributions. 1) We develop a novel algorithm, based on extragradient updates, for distributed stochastic MVIs with heterogeneous data and, in particular, distributed stochastic SPP. Our scheme supports very general communication protocol that covers centralized settings as in Federated Learning, fully decentralized settings, local steps in the both centralized/decentralized settings, and time-varying topology. In particular, we are not aware of earlier works proposing or analyzing extragradient method with local steps for the fully decentralized setting or decentralized algorithms for stochastic MVIs over time-varying networks.

2) Under the very general communication protocol and in the three settings of MVIs with an operator that is strongly-monotone, monotone, or non-monotone with the Minty condition, we prove convergence for our algorithm and give an explicit dependence of the rates on the problem parameters: characteristics of the network and how it changes with time, data heterogeneity, variance of the data, number of devices, and other standard parameters. These theoretical results clearly translate to the corresponding three settings of SPP (strongly-convex-strongly-concave, convex-concave, non convex-concave under Minty condition). All our theoretical results are valid in the important heterogeneous data regime and allow judging in a quantifiable way how different properties, e.g., data heterogeneity, scale of the noise in the data, and network characteristics influence the convergence rate of the algorithm. Even for decentralized setting, our results are novel for time-varying graphs and three different settings of monotonicity. See also Table 1 that gives more details on our contribution compared to existing literature. The main challenge of our analysis is to deal with very general assumption on the communication protocol and cope with the errors caused by stochastic nature and heretogeneity of the data and limited information exchange between the nodes of the communication network. As a byproduct of independent interest, we analyze stochastic extragradient method with biased oracle on unbounded domains that was not done so far in the literature.

3) We verify our theoretical results in numerical experiments and demonstrate the practical effectiveness of the proposed scheme we train DCGAN [72] architecture on the CIFAR-10 [45] dataset.

1.1 Related Work

The research on MVIs dates back at least to 1962 [57] with the classic book [36] and the recent works [53, 50, 11, 18]. VIs arise in a broad variety of settings: image denoising [20, 12], game theory and optimal control [21], robust optimization [7] or non-smooth oprimization via smooth reformulations [67, 66]. In ML, MVIs and SPP arise in training GANs [16, 13, 14], reinforcement learning [69, 31], and adversarial training [54].

Extragradient. The extragradient method (EGM) was first proposed in [43] and extended as mirror-prox method for deterministic problems in [66] and for stochastic ones with bounded variance in [32]. Yet, if the stochastic noise is not uniformly bounded, the EGM can diverge, see [13, 59].

Reference	base method	arbitrary network	time-varying	local updates	no multiple gossip steps	SM	M	NM
Liu et al. 2019 [52]	Stoch. ES	✔	✘	✘	✘	✘	✘	✔†
Beznosikov et al. 2021[9] Alg. 2	Stoch. ES	✔	✘	✘	✘	✔	✔	✘
Barazandeh et el. 2021 [4]	Stoch. ES	✔	✘	✘	✘	✘	✘	✔
Liu et al. 2019 [53]	Deter. prox	✔	✘	✘	✔	✘	✘	✔
Mukherjee and Chakraborty 2020 [62]	Deter. ES	✔	✘	✘	✔	✔	✘	✘
Tsaknakis et al. 2020 [81]	Stoch. DA	✔	✘	✘	✔	✘	✘	✔‡
Rogozin et al. 2021[73]	Deter. ES	✔	✘	✘	✔	✘	✔	✘
Xian et al. 2021 [87]	Stoch. DA	✔	✘	✘	✔	✘	✘	✔‡
Beznosikov et al. 2021 [9] Alg 3	Stoch. ES	✘	✘	✔	-§	✔	✘	✘
Deng and Mahdavi 2021[17]	Stoch. DA	✘	✘	✔	-	✔	✘	✔‡
Hou et al. 2021 [27]	Stoch. DA	✘	✘	✔	-	✔	✘	✘
Ours	Stoch. ES	✔	✔	✔	✔	✔	✔	✔

† – homogeneous case, ‡ – non-convex-concave SPP (other works use minty condition – (NM)), § – this column does not apply to centralized algorithms.

Table 1: Comparison of approaches for distributed strongly monotone (SM), monotone (M) and non-monotone (NM) VI or, respectively, strongly-convex-strongly-concave, convex-concave, non-convex-non-concave SPPs. Definition of columns: **base method**—the non-distributed algorithm that is taken as a basis for the distributed method, typically it is either an extragradient (ES) method or descent-ascent (DA); **arbitrary network**—supporting fully decentralized vs. only centralized topology; **time-varying**—decentralized method with time-varying network topology; **local updates**—method supporting local steps between communications; **no multiple gossip steps**—at one global iteration method does not use many iterations of gossip averaging to good accuracy; **SM,M,NM**—monotonicity assumption, see Assumption 3.2.

Decentralized algorithms for MVIs and SPP are the most closely related to our work and we summarize them in Table 1 and compare with our contribution, e.g. existing methods do not support arbitrary time-varying network typologies. The methods that use multiple rounds of gossip averaging (sparse communication) per iteration [52, 9, 4] can give optimal theoretical rates, but are often unstable in practice. Thus, it is preferred to have only one sparse communication per iteration [53, 62, 81, 73, 87]. The second column of the table refers to standard algorithms that are extended to distributed setting in the corresponding work. In particular, [53] require expensive proximal updates. The closest work to ours is [9], where decentralized EGM without local steps is analyzed in (strongly-)monotone setting. Unlike our more general algorithm with local steps and analysis, they require multiple gossip updates in each iteration which is not desired in practice. For the FL (centralized) setting, the same work studies EGM with local steps in the strongly-monotone setting, and [17, 27] study descent-ascent method with local steps, and all three do not consider arbitrary time-varying graphs as we do.

2 Algorithm

In this section we present and discuss the proposed algorithm (Algorithm 1) that is based on two main ideas: (i) an extragradient step, as in the classical methods for VIs [43, 66], and (ii) gossip averaging [10, 64] widely used in decentralized optimization methods.

Algorithm 1 can be divided into two phases. The local phase (lines 4–6) consists of a step of the stochastic extragradient method at each node using only local information. As in the non-distributed case, the nodes make first an extrapolation step—to "look into the future"—and then an update based on the operator value at the "future" point. This is followed by the communication phase (line 1), during which the devices average local iterates with their neighbors $\mathcal{N}_m^k$ in the current network graph

corresponding to the iteration k. The averaging process involves the weights $w_{m,i}^k$ – elements of the matrix W^k,which is called the mixing matrix:

Definition 2.1 (Mixing matrix)**.** We call a matrix $W \in [0;1]^{M\times M}$ a mixing matrix if it satisfies the following conditions: 1) W is symmetric, 2) W is doubly stochastic ($W\mathbf{1} = \mathbf{1}$, $\mathbf{1}^T W = \mathbf{1}^T$, where $\mathbf{1}$ denotes the all-one vector), 3) W is aligned with the network: $w_{ij} \neq 0$ if and only if $i = j$ or edge (i, j) is in the network graph.

Typical choices of mixing matrices are for example (i) the choice $W^k = I_M - \frac{L^k}{\lambda_{\max}(L^k)}$, where L^k denotes the Laplacian matrix of the network graph at time k and I_M the identity matrix, or (ii) local rules based on the degrees of the neighboring nodes [88]. Note that our setting allows a great flexibility as in between the iterations the topology of the communication graph can change, and the matrix W^k, that encodes the current structure of the network, changes accordingly. This is encoded in line 1, where the matrix W^k is generated by some rule $\mathcal{W}^k$ which can have different nature. Examples include deterministic choice of a sequence of matrices W^k, sampling from a time-varying probability distribution on matrices. Even local steps with no communication can be encoded with a diagonal matrix W^k. To ensure that it is possible to approach the consensus between the agents, we need the following assumption on the mixing properties of the matrix sequence W^k.

Algorithm 1 Extra Step Time-Varying Gossip Method

parameters:stepsize $\gamma > 0$, $\{\mathcal{W}^k\}_{k\geq 0}$ – rules or distributions for mixing matrix in iteration k.
initialize:$z^0 \in \mathcal{Z}, \forall m : z_m^0 = z^0$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Sample matrix W^k from $\mathcal{W}^k$
3: **for** each machine m **do**
4: Generate independently $\xi_m^k \sim \mathcal{D}_k$, $\xi_m^{k+1/3} \sim \mathcal{D}_k$
5: $z_m^{k+1/3} = z_m^k - \gamma F_m(z_m^k, \xi_m^k)$
6: $z_m^{k+2/3} = z_m^k - \gamma F_m(z_m^{k+1/3}, \xi_m^{k+1/3})$
7: $z_m^{k+1} = \sum_{i\in\mathcal{N}_m^k} w_{m,i}^k z_i^{k+2/3}$
8: **end for**
9: **end for**

Assumption 2.2 (Expected Consensus Rate)**.** We assume that there exist two constants $p \in (0, 1]$ and integer $\tau \geq 1$ such that for all matrices $Z \in \mathbb{R}^{d\times M}$ and all integers $l \in \{0, \ldots, T/\tau\}$,

$$\mathbb{E}_W[\|ZW_{l,\tau} - \bar{Z}\|_F^2] \leq (1-p)\|Z - \bar{Z}\|_F^2, \tag{3}$$

where $W_{l,\tau} = W^{l\tau} \cdot \ldots \cdot W^{(l+1)\tau-1}$, we use matrix notation $Z = [z_1, \ldots, z_M]$, $\bar{Z} = [\bar{z}, \ldots, \bar{z}]$ with $\bar{z} = \frac{1}{M}\sum_{m=1}^M z_m$, and the expectation $\mathbb{E}_W$ is taken over distributions of W^t and indices $t \in \{l\tau, ..., (l+1)\tau - 1\}$.

This assumption ensures that after τ steps of the gossip algorithm with such time-varying matrices we improve the averaging between nodes by the factor of $\frac{1}{1-p}$. It is important that in this case some matrices W^k can be, for example, the identity matrix (which corresponds to performing local steps only in this round).

Assumption 2.2 was recently quite popular in the literature on distributed optimization methods [65, 39, 44]. Moreover, it is very general and covers many special cases of decentralized and centralized algorithms. For example, if we fix $W^k = W$ for some connected graph we get a decentralized algorithm on a constant topology. If at the same time we set the matrix $W = \frac{1}{M}\mathbf{1}\mathbf{1}^T$, then it is easy to make sure that then we get an analogue of centralized learning with averaging over all nodes in one communication step. If we take the matrix $W^k = W$ for some connected graph at every τth iteration, and in other cases use $W^k = I_M$, we have a decentralized (or centralized) algorithm with local iterations [77, 25, 39] and communication once in τ iterations. Generic Assumption 2.2 covers also many other settings of time-varying decentralized topologies, e.g. random topologies, cliques, B-connected graphs [30, 63]. Below we provably show that under an appropriate choice of the stepsize our extragradient method provably converges under such a general assumption that covers centralized, decentralized settings, local steps in the both centralized/decentralized settings, and changing topology. Even for decentralized setting, this is novel for time-varying graphs and three different settings of monotonicity.

3 Setting and assumptions

In this section we introduce necessary assumptions that are used to analyze the proposed algorithm.

Assumption 3.1 (Lipschitzness)**.** For all m, the operator $F_m(z)$ is Lipschitz with constant L, i.e.,

$$\|F_m(z_1) - F_m(z_2)\| \leq L\|z_1 - z_2\|, \quad \forall z_1, z_2. \tag{L}$$

This is a standard assumption that is used in the analysis of all the methods displayed in Table 1.

Assumption 3.2. We consider three scenarios for the operator F, namely when F is strongly monotone, monotone and non-monotone, but with an additional assumption:
(SM) Strong monotonicity. There exists $\mu > 0$ such that for all z_1, z_2

$$\langle F(z_1) - F(z_2), z_1 - z_2\rangle \geq \mu\|z_1 - z_2\|^2. \tag{SM}$$

(M) Monotonicity. For all z_1, z_2, it holds that:

$$\langle F(z_1) - F(z_2), z_1 - z_2\rangle \geq 0. \tag{M}$$

(NM) Non-monotonicity (Minty). There exists z^* such that for all z:

$$\langle F(z), z - z^*\rangle \geq 0. \tag{NM}$$

Assumptions (SM), (M) and (L) are standard classical assumptions in the literature on VIs. Assumption (NM) is sometimes referred to as the *Minty or Variational Stability condition* and is widely used recently by the community as a structured variant of non-monotonicity [15, 29, 56, 53, 34, 28, 18], particularly being appropriate for GANs training [51, 52, 19, 4].

The next assumption is standard for the stochastic setting.

Assumption 3.3 (Bounded noise)**.** $F_m(z, \xi)$ is unbiased and has bounded variance, i.e. for all z

$$\mathbb{E}[F_m(z, \xi)] = F_m(z), \ \mathbb{E}[\|F_m(z, \xi) - F_m(z)\|^2] \leq \sigma^2. \tag{4}$$

Our last assumption reflects the variability of the local operators compared to their mean and is usually called D-heterogeneity. This assumption is widely used in the analysis of local-steps (and not only) algorithms for minimization problems [35, 86, 25, 75, 2, 26]. Moreover, [17, 27] use this assumption for the analysis of centralized local-steps methods for SPPs. The authors of [52] assume $D = 0$ for decentralized training of GANs. Even in this case algorithms' analysis may be challenging.

Assumption 3.4 (D-heterogeneity.)**.** The values of the local operator have bounded variablility, i.e., for all z

$$\|F_m(z) - F(z)\|^2 \leq D^2. \tag{5}$$

4 Main results

In this section, we present the convergence results for the proposed method under different settings of Assumption 3.2. To present the main result, we introduce notation $\bar{z}^k := \frac{1}{M}\sum_{m=1}^M z_m^k$, $\bar{z}^{k+1/3} := \frac{1}{M}\sum_{m=1}^M z_m^{k+1/3}$ for the averaged among the devices iterates and $\widehat{z}^k = \frac{1}{k+1}\sum_{i=0}^k \bar{z}^{i+1/3}$ for the averaged among the devices and iterates sequence, a.k.a. ergodic sequence. Finally, we denote $\Delta = \frac{\tau}{p}\left(\frac{D^2\tau}{p} + \sigma^2\right)$ which plays a role of the consensus error—the error that is due to impossibility to reach the exact consensus between the agents. Note that the data heterogeneity appears in the convergence rates only through the quantity Δ.

Theorem 4.1 (Main theorem)**.** *Let Assumptions 2.2, 3.1, 3.3, 3.4 hold and the sequences $\bar{z}^k$, $\widehat{z}^k$ be generated by Algorithm 1 that is run for $K > 0$ iterations. Then,*

- ***Strongly-monotone case:*** *under Assumption 3.2(SM), with* $\gamma = \mathcal{O}\left(\min\left\{\frac{p}{\tau L}, \frac{\ln\left(\mu MK\|z^0 - z^*\|^2/(20\sigma^2)\right)}{\mu K}\right\}\right)$ *it holds that*

$$\mathbb{E}\left[\|\bar{z}^{K+1} - z^*\|^2\right] = \tilde{\mathcal{O}}\left(\|z^0 - z^*\|^2 \cdot \exp\left(-\frac{\mu K p}{240 L\tau}\right) + \frac{\sigma^2}{\mu^2 MK} + \frac{L^2\Delta}{\mu^4 K^2}\right); \tag{6}$$

• ***Monotone case:*** *under Assumption 3.2(M), for any convex compact $\mathcal{C}$ s.t. $z^0, z^* \in \mathcal{C}$ and $\max_{z,z'\in\mathcal{C}} \|z - z'\| \leq \Omega_{\mathcal{C}}$, with $\gamma = \mathcal{O}\left(\min\left\{\frac{1}{L}, \left(\frac{\Omega_{\mathcal{C}}^2 M}{K\sigma^2}\right)^{\frac{1}{2}}, \left(\frac{\Omega_{\mathcal{C}}^2}{K^2L^2\Delta}\right)^{\frac{1}{4}}\right\}\right)$ it holds that*

$$\sup_{z\in\mathcal{C}} \mathbb{E}\left[\left\langle F(z), \widehat{z}^K - z\right\rangle\right] = \mathcal{O}\Big(\frac{L\Omega_{\mathcal{C}}^2}{K} + \frac{\sigma\Omega_{\mathcal{C}}}{\sqrt{MK}} + \frac{\sqrt{L\Omega_{\mathcal{C}}^3\sqrt{\Delta}}}{\sqrt{K}} + \sqrt{\frac{(\Delta+L^2\Omega_{\mathcal{C}}^2)\Omega_{\mathcal{C}}\sqrt{\Delta}}{KL}}\Big) \tag{7}$$

Under the additional assumption that, for all k, $\|\bar{z}^k\| \leq \Omega$, with $\gamma = \mathcal{O}\left(\min\left\{\frac{1}{L}, \left(\frac{\Omega_{\mathcal{C}}^2 M}{K\sigma^2}\right)^{\frac{1}{2}}, \left(\frac{\Omega_{\mathcal{C}}^2}{K^2L^2\Delta}\right)^{\frac{1}{4}}, \left(\frac{\Omega_{\mathcal{C}}^2}{K((\Omega+\Omega_{\mathcal{C}})L\sqrt{\Delta}+\Delta)}\right)^{\frac{1}{2}}\right\}\right)$, we have that

$$\sup_{z\in\mathcal{C}} \mathbb{E}\left[\left\langle F(z), \widehat{z}^K - z\right\rangle\right] = \mathcal{O}\Big(\frac{L\Omega_{\mathcal{C}}^2}{K} + \frac{\sigma\Omega_{\mathcal{C}}}{\sqrt{MK}} + \frac{\sqrt{L\Omega_{\mathcal{C}}^3\sqrt{\Delta}}}{K^{3/4}} + \sqrt{\frac{((\Omega+\Omega_{\mathcal{C}})L\sqrt{\Delta}+\Delta)\Omega_{\mathcal{C}}^2}{K}}\Big); \tag{8}$$

• ***Non-monotone case:*** *under Assumption 3.2(NM) and if $\|z^0\| \leq \Omega, \|z^*\| \leq \Omega$, with $\gamma = \mathcal{O}\left(\min\left\{\frac{1}{L}, \left(\frac{\Omega^2}{K^2L^2\Delta}\right)^{\frac{1}{4}}\right\}\right)$:*

$$\mathbb{E}\Big[\frac{1}{K+1}\sum_{k=0}^K \|F(\bar{z}^k)\|^2\Big] = \mathcal{O}\Big(\frac{L^2\Omega^2}{K} + \frac{\sigma^2}{M} + L\Omega\sqrt{\Delta} + \frac{\sqrt{L\Omega\Delta^{3/4}}}{\sqrt{K}}\Big). \tag{9}$$

Under an additional assumption that, for all k, $\|\bar{z}^k\| \leq \Omega$, with $\gamma = \mathcal{O}\left(\min\left\{\frac{1}{L}, \left(\frac{\Omega^2}{KL\Delta}\right)^{\frac{1}{3}}\right\}\right)$, we have that

$$\mathbb{E}\Big[\frac{1}{K+1}\sum_{k=0}^K \|F(\bar{z}^k)\|^2\Big] = \mathcal{O}\Big(\frac{L^2\Omega^2}{K} + \frac{\sigma^2}{M} + \frac{(L\Omega\Delta)^{2/3}}{K^{1/3}} + L\Omega\sqrt{\Delta}\Big). \tag{10}$$

The proof of the theorem is given in the supplementary material, where one can also find explicit dependence of the rates on the stepsize γ before it is chosen in an optimal way. We underline that the standard analysis [32] does not apply for the following reasons. Firstly, unlike [32], in our problem (1), the feasible set is not bounded, which is especially important for the analysis in the monotone and non-monotone settings. Secondly, our algorithm has an additional communication step (Step 1) between the computational nodes, which leads to impossibility for all the nodes to have the same information about the global operator $F(z)$ and about the current iterate z. This, in order, leads to biased oracle that, unlike existing works, has to be analyzed in the setting of unbounded feasible set, which is quite challenging. To analyze our variant of the extragradient method we successfully handle this challenge. Our key steps are to bound the bias (see, e.g. the last two terms in the r.h.s. of Lemma C.7 that are caused by the network errors), prove the boundedness in expectation of the sequence of the iterates for monotone (see Section C.3.1 of the supplementary material) and non-monotone (see Section C.4.1 of the supplementary material) cases, which may be of independent interest and which we have not seen in the literature, even in the non-distributed setting with biased stochastic oracles. Proving the boundedness is challenging due to noise caused by stochasticity and heterogeneity of the data and network effects due to the imperfect exchange of information. Surprisingly, in the end we still manage to analyze our algorithm under the very general Assumption 1 and we are not aware of any results with similar generality of the settings: different networks topologies (including time-varying), distributed architectures, different monotonicity assumptions.

The provided convergence rates have explicit dependence on the problem parameters: network that is characterized by mixing time τ and mixing factor p, data heterogeneity D (these three quantities appear in the convergence rates only through the quantity Δ), variance σ^2 of the noise in the data, Lipschitz constant L and strong monotonicity parameter μ, number of devices M. Thus, our rates allow judging how different properties, e.g., data heterogeneity, amount of noise, network characteristics influence the convergence rates. This opens an opportunity for meta-optimization process if we can design the network by changing M, τ, p to achieve faster convergence.

We now discuss the convergence results obtained in the theorem, and also compare them with the already existing algorithms (see Table 1) and their guarantees. Firstly, all the estimates have similar there-component structure. The first term corresponds to the deterministic setting and is similar to existing methods for smooth VIs in non-distributed setting. Only in the strongly convex case there is an additional factor τ/p that increases the condition number L/μ of the problem. The second (stochastic) term is also standard for non-distributed setting and corresponds to the stochastic nature

of the problem. Note that, for a very general distributed setting we have managed to obtain the corresponding terms similar to non-distributed setting. Moreover, we can see the benefit of exploiting distributed computations: the leading stochastic term depends on σ^2/M that decreases as the number M of nodes increases. The other terms correspond to the consensus error Δ and are caused by imperfect communications between the agents, i.e. that it is impossible for the agents to have exactly the same information about the current iterate. Importantly, in all the cases this error does not make the overall convergence worse since the dependence on K is no worse for these terms than the dependence on K in the stochastic term. In the experimental section we illustrate that the network error is not an artefact of the analysis, but indeed is present in practice.

Theorem 4.1 is formulated for a fixed budget of iterations K and the corresponding stepsizes γ that depend on K, which is pretty standard in the literature [32, 78, 8], where many algorithms fix the stepsize depending on the budget of the iterations. In Appendix D, we present a simple restarting procedure that allows to extend the results of Theorem 4.1 to any-time convergence without a-priori fixing the number K. The idea is to set $K_t = 2^t$ for $t = 0, 1, \ldots$ and restart the algorithm after each K_t iterations. We next make refined comments for each particular setting.

• **Strongly-monotone case:** In the centralized setting with local updates our rate is slightly better than in [9]. Unlike our algorithm, centralized algorithms with local steps for SPP in [17, 27] are based on gradient descent-ascent that may diverge in the stochastic setting even for bilinear problems. Moreover, their analysis implies a very small stepsize $\gamma \sim \frac{\mu p}{L^2 \tau}$ (cf. ours $\gamma \sim \frac{p}{L\tau}$) that greatly slows down the convergence of the algorithm.

For the decentralized setting [9] propose an optimal algorithm with the rate matching the lower bound which they give. Our rate is worse probably because of the generality of the Assumption 2.2. On the other hand, we do not rely on many gossip iterations at once that should be avoided in practice. Also, our algorithm is more general, allowing us to work with time-varying topology and local steps even in the decentralized setting.

• **Monotone case:** The quantity $\sup_{z\in\mathcal{C}} \mathbb{E}\left[\left\langle F(z), \widehat{z}^K - z\right\rangle\right]$ in the estimates reflects the stochastic nature of the problem and is a counterpart of the standard restricted gap (or merit) function [68]: $\text{Gap}_{\mathcal{C}}(u) := \sup_{z\in\mathcal{C}} [\langle F(z), u - z\rangle]$. When F is a monotone operator, if $\text{Gap}_{\mathcal{C}}(\hat{u}) = 0$ and $\mathcal{C}$ contains a neighborhood of $\hat{u}$, then [68, 1] $\hat{u}$ is a solution to (1) and even more: it is a strong solution to the corresponding variational inequality, i.e., for all z, $\langle F(\hat{u}), \hat{u} - z\rangle \leq 0$. Thus, $\text{Gap}_{\mathcal{C}}(u)$ is an appropriate measure of suboptimality in this setting and (7) guarantees that after a sufficient number of iterations, we obtain an approximate solution in expectation. Importantly, for (7), neither z nor $\bar{z}^k$ are assumed to be bounded. As in the previous works on non-distributed algorithms for MVIs [68, 1], we use $\text{Gap}_{\mathcal{C}}(u)$ for an arbitrary compact set $\mathcal{C}$ that contains z^0 and z^* (this can be a large set). Further, (8) is a refined version of the general result (7) under additional assumptions of boundedness. If boundedness does not hold, we still have (7). Moreover, (7) and (8) hold for the same method and to run the algorithm, there is no need to know in advance whether the generated sequence is bounded or not.

Only [9, 73] consider MVIs with monotone operator in distributed setting. Our algorithm is more general than theirs: our algorithm supports time-varying networks and local steps in between communications. The algorithm in [9] uses multiple gossip steps between the updates of the iterates. On the one hand, this allows decrease the consensus error Δ. On the other hand, this leads to an additional factor in the number of communications compared to our estimates: the first term in their bound is $\sqrt{\chi}$ times larger than ours, where $\chi > 1$ is some condition number of the mixing matrix. Moreover, multiple gossip steps may be impractical if the communication is performed through unstable channels or is expensive by some reason. The paper [73] considers only deterministic setting.

• **Non-monotone case:** Here the same as in the previous case remark on the boundedness of $\bar{z}^k$, z^* assumed to obtain (10) can be repeated. Further, in this setting the convergence is guaranteed up to some accuracy that is governed by the stochastic nature of the problem (the σ^2 term) and by the distributed nature of the problem (Δ terms). With this respect the results are similar to non-distributed stochastic extragradient method [5] and distributed method in the homogeneous case [52]. To the best of our knowledge, convergence up to arbitrarily small accuracy can be guaranteed only for deterministic distributed methods [53], i.e. in a much simpler setting than ours. Moreover, the methods of [53] are not the most robust since they require evaluating the proximal operator of a function and it is assumed that this can be done in closed-form, which is computationally expensive and may not hold in practice.

Note that based on our result it is possible to achieve convergence up to arbitrarily small accuracy if one considers the homogeneous case ($D = 0$). Indeed, choosing the right batch size, for example, $\sim K^\alpha$ with $\alpha > 0$, one can replace σ^2 by $\frac{\sigma^2}{K^\alpha}$ in (9) and (10) and get convergence.

5 Experiments

We present two set of experiments to validate the performance of Algorithm 1. In Section 5.1 we verify the proven convergence guarantee on a strong-monotone and on a monotone bilienar problem, and in Section 5.2 we explore the non-monotone case with application of GAN training. Extended details about the experimental setup can be found in the supplementary material.

5.1 Verifying Theoretical Convergence Rate

First, we focus on verifying whether the Algorithm 1 behaviour is predicted by the theoretical convergence rate (Theorem 4.1).

Setup. We consider a distributed bi-linear SPP (2) with objective functions $f_m(x, y) = \frac{a}{2}\|x\|^2 + \frac{b}{2}x^\top y - \frac{a}{2}\|y\|^2 + c_m^\top x$, where $x, y, c_m \in \mathbb{R}^n$, $a, b \in \mathbb{R}$ and $m \in \{1, \dots, M\}$. This set of functions satisfy Assumptions 3.1, 3.2, 3.4 with constants $\mu = a$, $L = a^2 + \frac{b^2}{4}$, $D = \max_m \|c_m - \bar{c}\|$. In this section we use a ring topology on $M = 9$ nodes, with uniform averaging weights, and we set the dimension $n = 2$, $a = b = 1$ and we set $D = 1$ and keep $\tau = 1$. The value of the parameter p in this setting is approximately 0.288 [41, Table 1]. To satisfy Assumption 3.3, we generate stochastic gradients by adding to the real gradients unbiased Gaussian noise with variance σ^2.

Convergence Behaviour. In Figure 1 we show the convergence of Algorithm 1 with a fixed stepsize on a strongly-monotone ($a = 1$) and monotone ($a = 0$) instance. In the strongy-convex case we see linear convergence up to the level of the heterogenity parameter and the noise. The convergence on the non-strongly monotone problem is stronger affected by the noise, but interestingly we also see linear convergence (with oscillations) when there is no noise. Note that convergence to some limit accuracy is expected since when a constant stepsize is used in stochastic optimization/stochastic variational inequalities with strong convexity/monotonicity, the algorithm is usually guaranteed to converge to a vicinity of the solution, see, e.g., Theorem 2 in [59]. This is also in accordance to Theorem 4.1 which for a fixed stepsize gurantees convergence to some non-zero limit accuracy and says that the error to drop to zero one needs to choose a decreasing stepsize. We additionally validate in the Appendix A.2 that with decreasing stepsize, algorithm can converge to zero error.

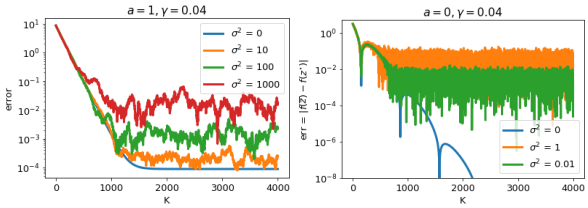


Figure 1: Convergence of Algorithm 1 with constant stepsize in the presence of stochastic noise in strongly monotone (top) and monotone (bottom) cases. We observe linear convergence up to an error floor depending on the noise variance and problem parameters (cf. Theorem 4.1). In Appendix A.2 we show convergence to arbitrary accuracy with decreasing stepsizes.

Depencence on the Heterogenity parameter D. In a second set of experiments we aim to verify the dependence on the data heterogeneity parameter D. Therefore, we consider the setting when $\sigma^2 = 0$. From our theory, equation (6), we predict that the most significant term in the convergence rate (when $\sigma^2 = 0$) scales as $\mathcal{O}\big(\frac{D^2}{p^2K^2}\big)$ (since the primary goal of this experiment is to study the dependence on p, D, K, we omit all the other fixed parameters for simplicity). We repeat experiments for different $a > 0$ (strongly-monotone case) with number of iterations needed until the error $\frac{1}{M}\sum_{m=1}^{M}\|z_m^k - z^*\|^2 < \epsilon$, for different ϵ. In all these experiments the step size is tuned individually.

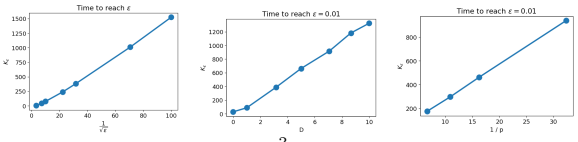


Figure 2: Verifying the $\mathcal{O}\big(\frac{D^2}{p^2K^2}\big)$ convergence for the strongly monotone noiseless ($\hat{\sigma}^2 = 0$) case.

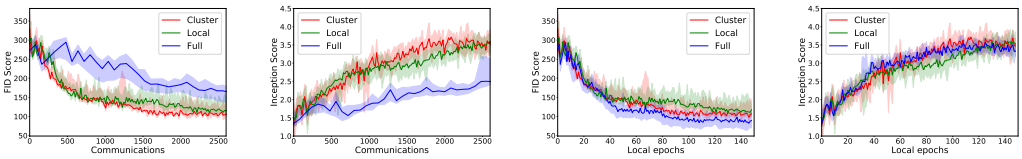
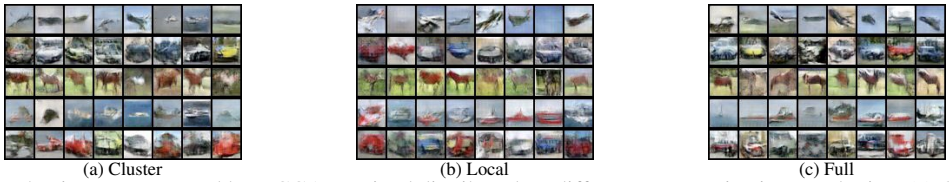


Figure 3: Comparison of three topological approaches in DCGAN distributed decentralized learning on CIFAR-10. FID Score and Inception Score in terms of the number of communications (two left), and Scores in terms of local epochs (two right). The experiment was repeated 5 times on different data random splitting, the maximum and minimum deviations are depicted in the plots by shade.



(a) Cluster (b) Local (c) Full

Figure 4: Pictures generated by DCGAN trained distributed on different communication topologies: (a) Cluster, (b) Local, (c) Full.

First, we verify the power of K. For this experiment, we keep D, L, μ, p constant and vary the accuracy ϵ. We can seen from the leftmost subplot in Figure 2 that the number of iterations scale as $K \propto \frac{1}{\sqrt{\epsilon}}$, confirming the predicted $T = \mathcal{O}\big(\frac{1}{K^2}\big)$ dependency. Next, we measure the number of iteration it takes to reach error $\epsilon = 0.01$ while varying D. The middle plot shows that the number of iterations scales proportional to D (showing $D \propto K$). Lastly, we depict the time to reach $\epsilon = 0.01$ while changing the graph parameter p and again observe $\frac{1}{p} \propto K$. All together, these experiments verify the $\mathcal{O}\big(\frac{D^2}{p^2 K^2}\big)$ term in the convergence rate.

5.2 GANs

Our algorithm allows you to combine different communication topologies of devices, as well as local steps in distributed learning. This is what we want to compare in the next experiment with GAN (in Appendix A.1, we discuss how close the theory is to GANs.).

Data and model. We consider the CIFAR-10 [45] dataset. It contains 60000 images (but we increased the size of data by 4 times due to transformations and adding noise), equally distributed over 10 classes. We simulate a distributed setup of 16 nodes on two GPUs, we are using Ray [60]. To emulate the heterogeneous setting, we partition the dataset into 16 subsets. For each subset, we select a major class that forms 20% of the data, while the rest of the data split is filled uniformly by the other classes. As a basic architecture we choose DCGAN [72], conditioned by class labels, similarly to [58] (the network architecture can be found in Appendix A.1). We chose Adam [37] as the optimizer. We make one local Adam step, and then one gossip averaging step with time-varying matrix W—similar to how it works in Algorithm 1.

Setting. We compare the following three topologies:
• **Full.** Full graph at the end of each epoch, otherwise local steps. This means that we make 120 communication rounds (by communication round we mean the exchange of information between a pair of devices) in an epoch.
• **Local.** Full graph at the end of each 5th epoch, otherwise local steps. This means that we make 24 communication rounds in an epoch (in average: 4 epochs without communications and 1 epoch with 120 rounds).
• **Clusters.** At the end of each epoch, clique clusters of size 4 are randomly formed (in total 4 cliques). This means that we make 24 rounds of communication in an epoch.

It turns out that the communication budget of the first approach is 5 times higher.

We use the same learning rate for the generator and discriminator equal to 0.002. The rest of the parameters and features of the architecture are contained in the Appendix.

Results. The results of the experiment can be found on Figure 3 and Figure 4. Note that all methods from the point of view of local epochs worked approximately the same and produced similar pictures. But from the point of view of communications, Local and Cluster topologies are much better. In turn, it can also be noted that the Cluster topology is slightly ahead of Local.

References

[1] Kimon Antonakopoulos, Veronica Belmega, and Panayotis Mertikopoulos. An adaptive mirror-prox method for variational inequalities with singular operators. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 8455–8465. Curran Associates, Inc., 2019.

[2] Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. *arXiv preprint arXiv:1506.01900*, 2015.

[3] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Michael Rabbat. Stochastic gradient push for distributed deep learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*. PMLR, 2019.

[4] Babak Barazandeh, Tianjian Huang, and George Michailidis. A decentralized adaptive momentum method for solving a class of min-max optimization problems. *Signal Processing*, 189:108245, 2021.

[5] Babak Barazandeh, Davoud Ataee Tarzanagh, and George Michailidis. Solving a class of non-convex min-max games using adaptive momentum methods. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3625–3629, 2021.

[6] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi. Personalized and private peer-to-peer machine learning. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84, pages 473–481. PMLR, 2018.

[7] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

[8] Aleksandr Beznosikov, Eduard Gorbunov, Hugo Berard, and Nicolas Loizou. Stochastic gradient descent-ascent: Unified theory and new efficient methods. *arXiv preprint arXiv:2202.07262*, 2022.

[9] Aleksandr Beznosikov, Valentin Samokhin, and Alexander Gasnikov. Distributed saddle-point problems: Lower bounds, optimal algorithms and federated GANs. *arXiv preprint arXiv:2010.13112*, 2021.

[10] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.

[11] Brian Bullins and Kevin A. Lai. Higher-order methods for convex-concave min-max optimization and monotone variational inequalities. *arXiv preprint arXiv:2007.04528*, 2020.

[12] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[13] Tatjana Chavdarova, Gauthier Gidel, François Fleuret, and Simon Lacoste-Julien. Reducing noise in GAN training with variance reduced extragradient. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019.

[14] Tatjana Chavdarova, Matteo Pagliardini, Sebastian U. Stich, Francois Fleuret, and Martin Jaggi. Taming GANs with lookahead-minmax. In *International Conference on Learning Representations (ICLR)*, 2021.

[15] Cong D Dang and Guanghui Lan. On the convergence properties of non-Euclidean extragradient methods for variational inequalities with generalized monotone operators. *Computational Optimization and Applications*, 60(2):277–310, 2015.

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

[17] Yuyang Deng and Mehrdad Mahdavi. Local stochastic gradient descent ascent: Convergence analysis and communication efficiency. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1387–1395. PMLR, 2021.

[18] Jelena Diakonikolas, Constantinos Daskalakis, and Michael Jordan. Efficient methods for structured nonconvex-nonconcave min-max optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2746–2754. PMLR, 2021.

[19] Zehao Dou and Yuanzhi Li. On the one-sided convergence of adam-type algorithms in non-convex non-concave min-max optimization. *arXiv preprint arXiv:2109.14213*, 2021.

[20] Ernie Esser, Xiaoqun Zhang, and Tony F Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM Journal on Imaging Sciences*, 3(4):1015–1046, 2010.

[21] F. Facchinei and J.S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2007.

[22] Gauthier Gidel, Hugo Berard, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial nets. In *International Conference on Learning Representations (ICLR)*, 2019.

[23] 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 (NIPS)*, 2014.

[24] Eduard Gorbunov, Pavel Dvurechensky, and Alexander Gasnikov. An accelerated method for derivative-free smooth stochastic convex optimization. *arXiv preprint arXiv:1802.09022*, 2018.

[25] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. Local sgd: Unified theory and new efficient methods. In *International Conference on Artificial Intelligence and Statistics*, pages 3556–3564. PMLR, 2021.

[26] Hadrien Hendrikx, Lin Xiao, Sebastien Bubeck, Francis Bach, and Laurent Massoulie. Statistically preconditioned accelerated gradient method for distributed optimization. In *International Conference on Machine Learning*, pages 4203–4227. PMLR, 2020.

[27] Charlie Hou, Kiran K Thekumparampil, Giulia Fanti, and Sewoong Oh. Efficient algorithms for federated saddle point optimization. *arXiv preprint arXiv:2102.06333*, 2021.

[28] Yu-Guan Hsieh, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. Explore aggressively, update conservatively: Stochastic extragradient methods with variable stepsize scaling. *Advances in Neural Information Processing Systems*, 33:16223–16234, 2020.

[29] 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.

[30] A. Jadbabaie, Jie Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[31] Yujia Jin and Aaron Sidford. Efficiently solving MDPs with stochastic mirror descent. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pages 4890–4900. PMLR, 2020.

[32] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[33] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song,

Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[34] Aswin Kannan and Uday V Shanbhag. Optimal stochastic extragradient schemes for pseudomonotone stochastic variational inequality problems and their variants. *Computational Optimization and Applications*, 74(3):779–820, 2019.

[35] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 4519–4529, 2020.

[36] David Kinderlehrer and Guido Stampacchia. *An Introduction to Variational Inequalities and Their Applications*. Society for Industrial and Applied Mathematics, 2000.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[38] Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does sgd escape local minima? In *International Conference on Machine Learning*, pages 2698–2707. PMLR, 2018.

[39] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U Stich. A unified theory of decentralized sgd with changing topology and local updates. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*. PMLR, 2020.

[40] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 3478–3487. PMLR, 2019.

[41] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3478–3487. PMLR, 09–15 Jun 2019.

[42] Lingjing Kong, Tao Lin, Anastasia Koloskova, Martin Jaggi, and Sebastian U. Stich. Consensus control for decentralized deep learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR, 2021.

[43] Galina Korpelevich. The extragradient method for finding saddle points and other problems. *Eknomika i Matematicheskie Metody*, 12:747–756, 1976.

[44] Dmitry Kovalev, Elnur Gasanov, Peter Richtárik, and Alexander Gasnikov. Lower bounds and optimal algorithms for smooth and strongly convex decentralized optimization over time-varying networks. *arXiv preprint arXiv:2106.04469*, 2021.

[45] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). `http://www.cs.toronto.edu/~kriz/cifar.html`, 2009.

[46] Guanghui Lan, Soomin Lee, and Yi Zhou. Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming*, Dec 2018.

[47] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. *arXiv preprint arXiv:1705.09886*, 2017.

[48] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 5330–5340. Curran Associates, Inc., 2017.

[49] Tao Lin, Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[50] Tianyi Lin, Chi Jin, and Michael I. Jordan. Near-optimal algorithms for minimax optimization. In *Proceedings of Thirty Third Conference on Learning Theory (COLT)*, volume 125, pages 2738–2779. PMLR, 2020.

[51] 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.

[52] Mingrui Liu, Wei Zhang, Youssef Mroueh, Xiaodong Cui, Jerret Ross, Tianbao Yang, and Payel Das. A decentralized parallel algorithm for training generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[53] Weijie Liu, Aryan Mokhtari, Asuman Ozdaglar, Sarath Pattathil, Zebang Shen, and Nenggan Zheng. A decentralized proximal point-type method for saddle point problems. *arXiv preprint arXiv:1910.14380*, 2019.

[54] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[55] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016.

[56] 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.

[57] George J. Minty. Monotone (nonlinear) operators in Hilbert space. *Duke Mathematical Journal*, 29(3):341 – 346, 1962.

[58] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[59] Konstantin Mishchenko, Dmitry Kovalev, Egor Shulgin, Peter Richtárik, and Yura Malitsky. Revisiting stochastic extragradient. *arXiv preprint arXiv:1905.11373*, 2019.

[60] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, pages 561–577, 2018.

[61] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, and Ion Stoica. Ray: A distributed framework for emerging ai applications, 2018.

[62] Soham Mukherjee and Mrityunjoy Chakraborty. A decentralized algorithm for large scale min-max problems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 2967–2972, 2020.

[63] Angelia Nedic, Alex Olshevsky, Asuman Ozdaglar, and John N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, 2009.

[64] Angelia Nedić and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[65] Angelia Nedich, Alex Olshevsky, and Wei Shi. A geometrically convergent method for distributed optimization over time-varying graphs. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1023–1029. IEEE, 2016.

[66] 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.

[67] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

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

[69] Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P. How, and John Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pages 2681–2690. PMLR, 2017.

[70] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037. Curran Associates, Inc., 2019.

[71] S. Pu and A. Nedić. Distributed stochastic gradient tracking methods. *Math. Program.*, 2020.

[72] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[73] Alexander Rogozin, Alexander Beznosikov, Darina Dvinskikh, Dmitry Kovalev, Pavel Dvurechensky, and Alexander Gasnikov. Decentralized distributed optimization for saddle point problems. *arXiv preprint arXiv:2102.07758*, 2021.

[74] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[75] Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1000–1008, Bejing, China, 22–24 Jun 2014. PMLR.

[76] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[77] Sebastian U Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations (ICLR)*, 2019.

[78] Sebastian U Stich. Unified optimal analysis of the (stochastic) gradient method. *arXiv preprint arXiv:1907.04232*, 2019.

[79] Hanlin Tang, Xiangru Lian, Shuang Qiu, Lei Yuan, Ce Zhang, Tong Zhang, and Ji Liu. Deepsqueeze: Decentralization meets error-compensated compression. *arXiv preprint arXiv:1907.07346*, 2019.

[80] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D^2: Decentralized training over decentralized data. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 4848–4856. PMLR, 2018.

[81] Ioannis Tsaknakis, Mingyi Hong, and Sijia Liu. Decentralized min-max optimization: Formulations, algorithms and applications in network poisoning attack. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5755–5759, 2020.

[82] César A Uribe, Soomin Lee, and Alexander Gasnikov. A dual approach for optimal algorithms in distributed optimization over networks. *arXiv preprint arXiv:1809.00710*, 2018.

[83] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. Decentralized collaborative learning of personalized models over networks. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 509–517. PMLR, 2017.

[84] 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.

[85] E. Wei and A. Ozdaglar. Distributed alternating direction method of multipliers. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 5445–5450, 2012.

[86] Blake Woodworth, Kumar Kshitij Patel, and Nathan Srebro. Minibatch vs local SGD for heterogeneous distributed learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

[87] Wenhan Xian, Feihu Huang, Yanfu Zhang, and Heng Huang. A faster decentralized algorithm for nonconvex minimax problems. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25865–25877. Curran Associates, Inc., 2021.

[88] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[89] 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.

[90] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J. Smola. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems 23*, pages 2595–2603. Curran Associates, Inc., 2010.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] , mostly theoretical paper
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] , Section 3
   (b) Did you include complete proofs of all theoretical results? [Yes] , Appendix
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] , Appendix A.1
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] , Section 5.2
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] , Appendix A
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] , Section 5.2, cite CIFAR and DCGAN
   (b) Did you mention the license of the assets? [N/A] , use open assets
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]