Decentralized Local Stochastic Extra-Gradient for Variational Inequalities

Aleksandr Beznosikov
Innopolis University[*], MIPT[†], HSE University and Yandex
anbeznosikov@gmail.com

Pavel Dvurechensky
WIAS[‡]
pavel.dvurechensky@wias-berlin.de

Anastasia Koloskova
EPFL
anastasia.koloskova@epfl.ch

Valentin Samokhin
IITP RAS[§]
samohin.vyu@phystech.edu

Sebastian U. Stich
CISPA[¶]
stich@cispa.de

Alexander Gasnikov
MIPT, HSE University and IITP RAS
gasnikov@yandex.ru

Abstract

We consider distributed stochastic variational inequalities (VIs) on unbounded domains with the problem data that is 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 the 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 (when a Minty solution exists) settings. The provided rates explicitly exhibit the dependence on network characteristics (e.g., mixing time), iteration counter, 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 the training of Deep Generative Adversarial Networks (GANs) for which decentralized training has been reported to be extremely challenging. In experiments for the 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 split between many devices, such as geographically distributed datacenters or mobile devices [38]. Decentralized training methods can learn an ML model with the same accuracy as if all the data would be aggregated on one single server [54, 5]. At the same time, training in a decentralized fashion has many advantages over traditional centralized approaches in such core aspects as data ownership, privacy, fault tolerance,

[*] Research Center for Artificial Intelligence, Innopolis University

[†] Moscow Institute of Physics and Technology

[‡] Weierstrass Institute for Applied Analysis and Stochastics

[§] Institute for Information Transmission Problems RAS

[¶] CISPA Helmholtz Center for Information Security

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

and scalability. A particular instance of the decentralized learning setting is Federated Learning (FL), where the training is orchestrated by a single device or server that communicates with all the participating client devices [62, 38]. In contrast, in fully decentralized learning (FD) scenarios the devices only communicate with their neighbors in the communication network graph with possibly arbitrary topology [54]. Thus, decentralized algorithms are important in scenarios where centralized communication is expensive, not desired, or impossible.

There have been tremendous advances recently in the development, design, and understanding of decentralized training schemes [71, 95, 86, 54, 84, 92, 90, 94, 23, 81, 50]. In particular, such aspects as data-heterogeneity [90, 78, 55], communication efficiency (through local updates [52, 44] or compression [89, 45]), and personalization [93, 8] have been studied. However, all these advances were aimed at training with single-criterion loss functions leading to minimization problems, and they do not apply to more general problem classes. For example, the training of Generative Adversarial Networks (GANs) [28] requires simultaneous competing optimization of the generator and the discriminator objectives, i.e., solving a non-convex-non-concave saddle-point problem (SPP). This problem structure makes GANs notoriously difficult to train even in the single-node setting [27, 15, 16], not talking about training over decentralized datasets [58, 69, 80].

Our goal in this paper is solving decentralized stochastic SPPs, and, more generally, decentralized stochastic Minty variational inequalities (MVIs) [64, 37]. In a decentralized stochastic MVI, the data is distributed over $M \geq 1$ devices/nodes and each device $m \in [M]$ has access to its local stochastic oracle $F_m(z, \xi_m)$ for the local operator $F_m(z) := \mathbb{E}_{\xi_m \sim \mathcal{D}_m} F_m(z, \xi_m)$. The data ξ_m in the device m follows an unknown distribution $\mathcal{D}_m$ that can be different for each device $m \in [M]$. The devices are connected via a communication network forming a graph such that two devices can exchange information if and only if the corresponding nodes are connected by an edge in this graph. The goal is, while respecting the communication constraints, to find cooperatively a point $z^* \in \mathbb{R}^n$ such that, for all $z \in \mathbb{R}^n$,

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

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

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

The relation to VI can be seen by considering the variable $z = \left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ 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 training, where x and y are the parameters of the generator and the discriminator, respectively, the local losses $f_m(x, y)$ are possibly non-convex-non-concave in x, y and one can not assume the monotonicity of F in general, see also [21].

In this paper, we develop a novel algorithm for solving problems (1) and (2). Note that the gradient descent-ascent scheme for the problem (2) may diverge even in the simple convex-concave setting with $M = 1$ device [15]. Thus, unlike [58], we use extragradient updates [48, 37, 27] as a building block and combine them with a gossip-type communication protocol [98, 12] on arbitrary, possibly time-varying, network topologies. One of the main challenges due to the communication constraints is a “network error” induced by the impossibility of all the devices to reach the exact consensus, i.e., to have exactly the same information about the current iterate of the algorithm. Thus, each device stores a local variable, and only approximate consensus among the devices can be achieved by gossip steps [47]. Unlike other decentralized algorithms [84, 58], our method avoids multiple gossip steps per iteration, which leads to better practical performance and the possibility to work on time-varying networks. Moreover, our method allows for multiple local updates between communication rounds to reduce the communication overhead. This also makes our approach suitable for communication- and privacy-restricted FL or fully decentralized settings [101].

Our contributions. 1) Based on extragradient updates, we develop a novel algorithm for distributed stochastic MVIs (and, as a special case, for distributed stochastic SPPs) with heterogeneous data. Our scheme supports a very general communication protocol that covers centralized settings as in Federated Learning, fully decentralized settings, local steps in both the centralized/decentralized settings, and time-varying network topologies. In particular, we are not aware of earlier works proposing or analyzing extragradient methods 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, i.e., with an operator that is strongly-monotone, monotone, or non-monotone under the Minty condition, we prove the convergence of our algorithm and give an explicit dependence of the rates on the problem parameters: characteristics of the network (e.g., mixing time), data heterogeneity, the variance of the data, number of devices, and other standard parameters. These theoretical results translate to the corresponding three settings of SPPs (strongly-convex-strongly-concave, convex-concave, non-convex-non-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, the scale of the noise in the data, and network characteristics, influence the convergence rate of the algorithm. Even for decentralized settings, 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 the existing literature. The main challenge of our analysis is to deal with the very general assumption about the communication protocol and cope with the errors caused by the stochastic nature and heterogeneity of the data and limited information exchange between the nodes of the communication network. As a byproduct of independent interest, we analyze the stochastic extragradient method with biased oracle on unbounded domains, which 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. In particular, we train the DCGAN [79] architecture on the CIFAR-10 [51] dataset.

1.1 Related Work

The research on MVIs dates back at least to 1962 [64] with the classical book [41] and the recent works [59, 56, 13, 21]. VIs arise in a broad variety of applications: image denoising [25, 14], game theory and optimal control [26], robust optimization [9], and non-smooth oprimization via smooth reformulations [74, 73]. In ML, MVIs and SPPs arise in GANs training [19, 15, 16], reinforcement learning [76, 36], and adversarial training [60].

Extragradient. The extragradient method (EGM) was first proposed in [48], generalized as the mirror-prox method for deterministic problems in [73] and for stochastic problems with bounded variance in [37]. Yet, if the stochastic noise is not uniformly bounded, the EGM may diverge, see [15, 66].

Reference	base method	arbitrary network	time-varying	local updates	no multiple gossip steps	SM	M	NM
Liu et al. 2019 [58]	Stoch. ES	✔	✘	✘	✘	✘	✘	✔[†]
Beznosikov et al. 2021[11] Alg. 2	Stoch. ES	✔	✘	✘	✘	✔	✔	✘
Barazandeh et el. 2021 [6]	Stoch. ES	✔	✘	✘	✘	✘	✘	✔
Liu et al. 2019 [59]	Deter. prox	✔	✘	✘	✔	✘	✘	✔
Mukherjee and Chakraborty 2020 [69]	Deter. ES	✔	✘	✘	✔	✔	✘	✘
Tsaknakis et al. 2020 [91]	Stoch. DA	✔	✘	✘	✔	✘	✘	✔[‡]
Rogozin et al. 2021[80]	Deter. ES	✔	✘	✘	✔	✘	✔	✘
Xian et al. 2021 [97]	Stoch. DA	✔	✘	✘	✔	✘	✘	✔[‡]
Beznosikov et al. 2021 [11] Alg 3	Stoch. ES	✘	✘	✔	-[§]	✔	✘	✘
Deng and Mahdavi 2021[20]	Stoch. DA	✘	✘	✔	-	✔	✘	✔[‡]
Hou et al. 2021 [32]	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) VIs or, respectively, strongly-convex-strongly-concave, convex-concave, non-convex-non-concave SPPs. Definitions of columns: **base method** — the non-distributed algorithm that is taken as the basis for the distributed method, typically it is either the extragradient method (EGM) or the descent-ascent (DA); **arbitrary network** — supporting fully decentralized vs. only centralized topology; **time-varying** — decentralized method supporting time-varying network topology; **local updates** — method supporting local steps between communications; **no multiple gossip steps** — at one global iteration the method does not use many iterations of gossip averaging to reach a good consensus accuracy; **SM, M, NM** — monotonicity assumption, see Assumption 3.2.

Decentralized algorithms for MVIs and SPPs are the most closely related to our work. In Table 1, we summarize their features and make a comparison with our algorithm, showing that, 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 [58, 11, 6] can give near-optimal theoretical rates, but are often unstable in practice. Thus, it is preferable to have only one sparse communication per iteration [59, 69, 91, 80, 97]. The second column of the table refers to standard algorithms that are extended to distributed settings in the corresponding work. In particular, the algorithm of [59] requires expensive proximal updates. The closest work to ours is [11], where a decentralized EGM without local steps is analyzed in the (strongly-)monotone setting. Unlike our more general algorithm with local steps, theirs require multiple gossip updates in each iteration which is not desired in practice. For the FL, i.e., centralized, setting, [11] studies the EGM with local steps in the strongly-monotone setting, and [20, 32] study the descent-ascent method with local steps. Yet, all three works do not consider arbitrary time-varying graphs as in our work.

2 Algorithm

In this section, we present and discuss the proposed algorithm (Algorithm 1) that is based on two main ideas: (i) the extragradient step, as in the classical methods for VIs [48, 73], and (ii) the gossip averaging [12, 71] widely used in decentralized optimization methods and in the literature on diffusion strategies in distributed learning [82, 83, 99, 2, 61]. Unlike these papers that propose algorithms for optimization problems by exploiting gradient descent, our algorithm is based on the extragradient method and is designed to solve VIs and SPPs. Moreover, unlike the mentioned works, our method also allows for local steps in-between the communication rounds and for time-varying networks and has non-asymptotic theoretical convergence rate guarantees.

Each step of 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 first make 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 (gossip step) (line 7), during which the nodes share and average local iterates with their neighbors $\mathcal{N}_m^k$ in the communication network graph corresponding to the iteration k. The averaging process involves the weights $w_{m,i}^k$ which are the elements of the matrix W^k 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 vector of all ones), 3) W is aligned with the network: $w_{ij} \neq 0$ if and only if $i = j$ or the edge (i, j) is in the communication network graph.

Reasonable choices of mixing matrices are, for example, (i) $W^k = I_M - \frac{L^k}{\lambda_{\max}(L^k)}$, where L^k is the Laplacian matrix of the network graph at the step k and I_M is the identity matrix, or (ii) using some local rules in the graph, based on the degrees of the neighboring nodes [98]. Note that our setting has a great flexibility since in-between the iterations the topology of the communication graph is allowed to change, and the matrix W^k, that encodes the current structure of the network, changes accordingly. This is encoded in line 2, 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 without communication can be encoded with a diagonal matrix 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 $k = 0, 1, 2, \dots$ do
2:     Sample matrix $W^k$ from $\mathcal{W}^k$
3:     for each node $m$ 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
```

To ensure that it is possible to approach the consensus between the nodes, we need the following assumption on the mixing properties of the matrix sequence W^k.

Assumption 2.2 (Expected Consensus Rate)**.** We assume that there exist a constant $p \in (0,1]$ and an integer $\tau \geq 1$ such that, after K iterations, for all matrices $Z \in \mathbb{R}^{d\times M}$ and all integers $l \in \{0,\dots,K/\tau\}$,

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

where $W_{l,\tau} = W^{l\tau} \cdot \ldots \cdot W^{(l+1)\tau-1}$, we use the matrix notation $Z = [z_1,\dots,z_M]$, $\bar{Z} = [\bar{z},\dots,\bar{z}]$ 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 gossip steps with such time-varying matrices, we improve the consensus between the nodes, i.e., how close each z_m is to $\bar{z}$, by the factor of $\frac{1}{1-p}$. Importantly, in this case, some matrices W^k can be, for example, the identity matrix (which corresponds to performing only local steps in iteration k).

Assumption 2.2 has been recently quite popular in the literature on distributed optimization methods [72, 44, 49]. 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 fixed connected graph, we get a decentralized algorithm on this graph. If, at the same time, we set the matrix $W = \frac{1}{M}\mathbf{1}\mathbf{1}^T$, then it is easy to see that we get an analog of the centralized setting with the averaging over all nodes performed in each communication step. If we take $W^k = W$ for some fixed connected graph at every τ-th step and in other steps use $W^k = I_M$, we have a decentralized (and, in particular, centralized) algorithm with local steps [87, 30, 44] and communications after each τ iterations. Generic Assumption 2.2 covers also many other settings of time-varying decentralized topologies, e.g., random topologies, cliques, B-connected graphs [35, 70]. Below we show that, under an appropriate choice of the stepsize, our extragradient method provably converges under such a general assumption that covers centralized and decentralized settings, local steps in both centralized and decentralized settings, and changing topologies of the communication graph. Even for decentralized settings, this is novel for time-varying graphs and three different settings of monotonicity which we consider.

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 and classical assumptions in the literature on VIs. Assumption (NM) is sometimes called the *Minty or Variational Stability condition* and it has been widely used recently by the community as a structured variant of non-monotonicity [18, 34, 63, 59, 39, 33, 21], particularly, since it is appropriate in GANs training [57, 58, 22, 6].

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 [40, 96, 30, 85, 4, 1, 31, 17, 24]. Moreover, [20, 32] use this assumption for the analysis of centralized local-steps methods for SPPs. The authors of [58] assume $D = 0$ for the decentralized training of GANs. Even in this case algorithms' analysis can 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 rate 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 average. Finally, we denote $\Delta = \frac{\tau}{p}\big(\frac{D^2\tau}{p} + \sigma^2\big)$ which plays the role of the consensus error, i.e., the error caused by the impossibility of reaching the exact consensus between the nodes. 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 $\gamma = \tilde{\mathcal{O}}\left(\min\left\{\frac{p}{\tau L}, \frac{1}{\mu K}\right\}\right)$ it holds that*

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

- ***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*

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

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

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

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

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

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

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

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 optimally. We underline that the standard analysis [37] does not apply for the following reasons. Firstly, unlike [37], 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 (line 7) between the computational nodes, which leads to the 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 a biased oracle that, unlike existing works, has to be analyzed in the setting of an 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.8 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 the noise caused by the 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 2.2 and we are not aware of any results with similar generality of the settings: different network topologies (including time-varying), distributed architectures, different monotonicity assumptions.

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

We now discuss the convergence results obtained in the theorem, and also compare them with already existing algorithms (see Table 1) and their guarantees. Firstly, all the estimates have a similar several-term structure. The first term corresponds to the deterministic setting and is similar to existing methods for smooth VIs in the 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 the 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 the 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 the nodes increases. The other terms correspond to the consensus error Δ and are due to the imperfect communications between the nodes, i.e., that all the nodes can't 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 in these terms than the dependence on K in the stochastic term. In the experimental section, we illustrate that the network error is not an artifact of the analysis but is indeed 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 [37, 88, 10], where many algorithms fix the stepsize depending on the budget of the iterations. In Section D of the supplementary material, we present a simple restarting procedure that allows to extend the results of Theorem 4.1 to any-time convergence without a-priori fixing K. The idea is to set $K_t = 2^t$ for $t = 0, 1, \dots$ and restart the algorithm after each K_t iterations. We next make refined comments for each particular setting of monotonicity.

• **Strongly-monotone case:** In the centralized setting with local updates, our rate is slightly better than in [11]. Unlike our algorithm, centralized algorithms with local steps for SPPs in [20, 32] are based on the gradient descent-ascent method 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 $\gamma \sim \frac{p}{L\tau}$), which greatly slows down the convergence of the algorithm.

For the decentralized setting, [11] propose an optimal algorithm with the rate matching the lower bound which they also give. Our rate is worse probably because of the generality of the Assumption 2.2. On the other hand, our algorithm is more practical since it avoids using multiple gossip steps at each iteration. Also, our algorithm is more general, allowing us to work with time-varying topologies 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 convergence rates estimates reflects the stochastic nature of the problem and is a counterpart of the standard restricted gap (or merit) function [75]: $\text{Gap}_{\mathcal{C}}(u) := \sup_{z\in\mathcal{C}}\left[\langle F(z), u - z\rangle\right]$. When F is a monotone operator, if $\text{Gap}_{\mathcal{C}}(\hat{u}) = 0$ and $\mathcal{C}$ contains a neighborhood of $\hat{u}$, then [75, 3] $\hat{u}$ 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 [75, 3], we use $\text{Gap}_{\mathcal{C}}(u)$ with 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 the additional assumption of the boundedness of the averaged iterates. If the 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 [11, 80] consider MVIs with monotone operator in distributed setting. Our algorithm is more general than theirs: our algorithm supports time-varying networks and local steps between communications. The algorithm in [11] uses multiple gossip steps between the updates of the iterates. On the one hand, this allows decreasing 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 for some reason. The paper [80] considers only deterministic setting.

• **Non-monotone case:** The same as in the previous case remark on the boundedness of $\bar{z}^k$, z^* assumed to obtain (10) applies in this case. 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 (the Δ-terms). With this respect, the results are similar to the non-distributed stochastic extragradient method [7] and the distributed method [58] analyzed in the homogeneous case $D = 0$. To the best of our knowledge, convergence up to arbitrarily small accuracy can be guaranteed only for deterministic distributed methods [59], i.e., in a much simpler setting than ours. Moreover, the methods of [59] 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 a 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 with $D = 0$. Indeed, choosing the right batch size, for example, proportionally to K^α with $\alpha > 0$, one can replace σ^2 by $\frac{\sigma^2}{K^\alpha}$ in (9) and (10) and get convergence guarantees.

5 Experiments

In this section, we present two sets of experiments to validate the performance of Algorithm 1. In Section 5.1, we verify the obtained convergence guarantees on two examples: a strongly-monotone and a monotone bilinear problems, and in Section 5.2, we explore the non-monotone case with an application to GANs training. Extended details of the experimental setup can be found in the supplementary material.

5.1 Verifying Theoretical Convergence Rate

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

Setup. We consider a distributed bilinear SPP (2) with the objective functions $f_m(x,y) = \frac{a}{2}\|x\|^2 + bx^\top y - \frac{a}{2}\|y\|^2 + c_m^\top x$, where $x, y, c_m \in \mathbb{R}^n$, $a, b \in \mathbb{R}$ and $m \in \{1, \ldots, M\}$. This set of functions satisfy Assumptions 3.1, 3.2, 3.4 with constants $\mu = a$, $L^2 = a^2 + b^2$, $D = \max_m \|c_m - \bar{c}\|$. In this section, we use a ring topology on $M = 20$ nodes with uniform averaging weights, and we set the dimension $n = 5$, $b = 1$, $D \approx 3$, and keep $\tau = 1$. The value of the parameter p in this setting is approximately 0.288 [46, Table 1]. To satisfy Assumption 3.3, we generate stochastic gradients by adding to the exact gradients unbiased Gaussian noise with variance σ^2.

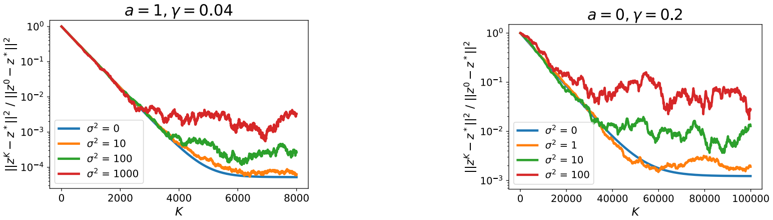

Figure 1: Convergence of Algorithm 1 with constant stepsize in the presence of stochastic noise in strongly-monotone (left) and monotone (right) cases. We observe linear convergence up to an error floor depending on the noise variance and problem parameters (cf. Theorem 4.1). In Section A.2 of the supplementary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

Convergence Behaviour. In Figure 1, we show the convergence of Algorithm 1 with a fixed stepsize on the strongly-monotone ($a = 1$) and monotone ($a = 0$) instances. In the strongly-monotone case, we see a linear convergence up to some level defined by the heterogeneity parameter and the noise. The convergence for the non-strongly-monotone problem is slower, but we also see a linear convergence up to some level (for bilinear problems this behavior is expected from the theoretical point of view [48]). Note that the convergence to some limiting accuracy is expected since when a constant stepsize is used in stochastic optimization/stochastic variational inequalities with strong convexity/monotonicity, algorithms are usually guaranteed to converge only to a vicinity of the solution, see, e.g., Theorem 2 in [66]. This is also in accordance with Theorem 4.1 that, for a fixed stepsize, guarantees the convergence to some non-zero limit accuracy and says that, to achieve the zero error, one needs to choose a decreasing stepsize. We additionally validate in Section A.2 of the supplementary material that with a decreasing stepsize, the algorithm can converge to the zero error.

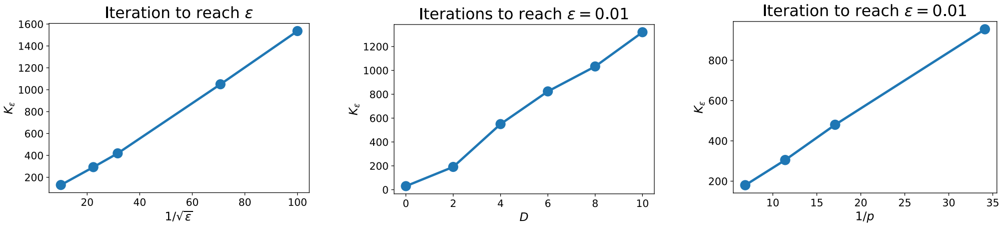

Figure 2: Verifying the $\mathcal{O}\big(\frac{D^2}{p^2K^2}\big)$ convergence rate for the strongly-monotone noiseless ($\sigma^2 = 0$) case.

Dependence on the Heterogeneity parameter D. In the 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 take $b = 1$, $a = 1$ and conduct experiments with the number of iterations needed to achieve the error $\frac{1}{M}\sum_{m=1}^{M}\|z_m^k - z^*\|^2 < \varepsilon$, for different ε. In all these experiments, the stepsize is tuned individually.

First, we verify the power of K in the bounds. For this experiment, we keep D, p constant and vary the accuracy ε. As we can see from the leftmost subplot in Figure 2, the number of iterations scales as $K \propto \frac{1}{\sqrt{\varepsilon}}$, confirming the predicted $\mathcal{O}\big(\frac{1}{K^2}\big)$ dependency of the error on K. Next, we measure the number of iterations sufficient to reach the error $\varepsilon = 0.01$ while varying D. The middle plot shows that the number of iterations scales proportionally to D (showing $D \propto K$). Lastly, we depict the number of iterations to reach $\varepsilon = 0.01$ while changing the graph parameter p and again observe $\frac{1}{p} \propto K$. Summarizing, these experiments verify the $\mathcal{O}\big(\frac{D^2}{p^2K^2}\big)$ term in the convergence rate.

5.2 Training GANs

Our algorithm allows combining in the distributed learning setting different communication graph topologies, as well as local steps. Thus, our goal in this section is to illustrate this empirically with the experiments on GANs training. In Section A.1 of the supplementary material, we discuss to what extent our theoretical results hold for GANs training.

Data and model. We consider the CIFAR-10 [51] dataset containing 60000 images, equally distributed over 10 classes. We increased the size of the dataset by 4 times using transformations and adding noise. We simulate a distributed setup of 16 nodes on two GPUs and use Ray [67]. 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 [79], conditioned by class labels, similarly to [65] (the network architecture can be found in Section A.1). We chose Adam [42] as the optimizer. We make one local Adam step and then one gossip averaging step with time-varying matrices W^k—similar to Algorithm 1.

Setting. We compare the following three topologies (and the corresponding matrices W^k):
- **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 communication rounds in an epoch.

Note that the communication budget of the first approach is 5 times larger.

We use the same learning rate equal to 0.002 for the generator and discriminator. The rest of the parameters and features of the architecture can be found in the supplementary material.

Results. The results of the experiment are presented in Figure 3 and Figure 6 (Section A.3). In terms of the number of local epochs, all the methods converged quite close to each other and produced similar images. In terms of communications, Local and Cluster topologies lead to much better results, and the Cluster topology is slightly better than the Local.

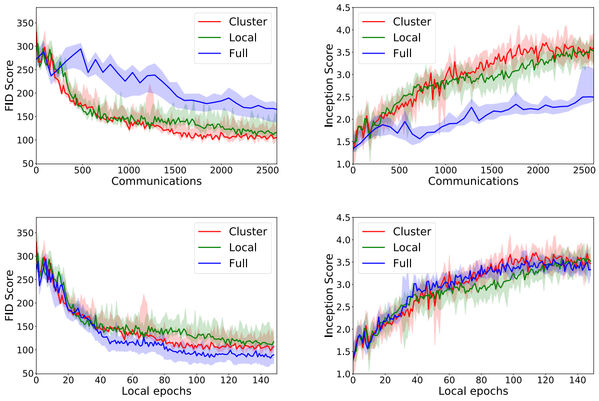

Figure 3: Comparison of the three network topologies in DCGAN distributed decentralized learning on CIFAR-10. FID Score and Inception Score vs the number of communications (two top), and the same Scores vs the number of local epochs (two bottom). The experiment was repeated 5 times on different random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

6 Conclusion

We propose a novel efficient algorithm for solving decentralized stochastic MVIs and SPPs under a very general assumption on the network topology and communication constraints. In particular, our method is the first decentralized extragradient method with local steps for time-varying network topologies. Moreover, for the proposed algorithm, we prove the convergence rate theorem in the SM, M and NM cases. In the numerical experiments, we verify that the dependence of our rates on the data heterogeneity parameter D is tight in the SM case, and cannot be further improved in general. By training DCGAN on a decentralized topology, we demonstrate that our method is effective on practical DL tasks. As a future work it would be interesting to generalize such algorithms for infinite-dimensional problems.

Acknowledgments

This research of A. Beznosikov has been supported by The Analytical Center for the Government of the Russian Federation (Agreement No. 70-2021-00143 dd. 01.11.2021, IGK 000000D730321P5Q0002). The work by P. Dvurechensky in Section C.3 was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689).

References

[1] Artem Agafonov, Pavel Dvurechensky, Gesualdo Scutari, Alexander Gasnikov, Dmitry Kamzolov, Aleksandr Lukashevich, and Amir Daneshmand. An accelerated second-order method for distributed stochastic optimization. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2407–2413, 2021.

[2] Sulaiman A Alghunaim and Kun Yuan. A unified and refined convergence analysis for non-convex decentralized learning. *IEEE Transactions on Signal Processing*, 2022.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] Amir Daneshmand, Gesualdo Scutari, Pavel Dvurechensky, and Alexander Gasnikov. Newton method over networks is fast up to the statistical precision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2398–2409. PMLR, 18–24 Jul 2021.

[18] 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.

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

[20] 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.

[21] 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.

[22] 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.

[23] Pavel Dvurechensky, Darina Dvinskikh, Alexander Gasnikov, César A. Uribe, and Angelia Nedić. Decentralize and randomize: Faster algorithm for Wasserstein barycenters. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, NeurIPS 2018, pages 10783–10793. Curran Associates, Inc., 2018.

[24] Pavel Dvurechensky, Dmitry Kamzolov, Aleksandr Lukashevich, Soomin Lee, Erik Ordentlich, César A. Uribe, and Alexander Gasnikov. Hyperfast second-order local solvers for efficient statistically preconditioned distributed optimization. *EURO Journal on Computational Optimization*, page 100045, 2022. (accepted), arXiv:2102.08246.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

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

[33] 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.

[34] 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.

[35] 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.

[36] 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.

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

[38] 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.

[39] 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.

[40] 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.

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

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

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

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

[49] 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.

[50] Roman Krawtschenko, César A. Uribe, Alexander Gasnikov, and Pavel Dvurechensky. Distributed optimization with quantization for computing wasserstein barycenters. *arXiv:2010.14325*, 2020. WIAS preprint 2782.

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

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

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

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] Alexey S. Matveev, Mostafa Almodarresi, Romeo Ortega, Anton Pyrkin, and Siyu Xie. Diffusion-based distributed parameter estimation through directed graphs with switching topology: Application of dynamic regressor extension and mixing. *IEEE Transactions on Automatic Control*, 67(8):4256–4263, 2022.

[62] 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.

[63] 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.

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

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

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

[67] 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.

[68] 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.

[69] 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.

[70] 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.

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

[72] 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.

[73] 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.

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

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

[76] 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.

[77] 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.

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

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

[80] 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.

[81] Alexander Rogozin, Mikhail Bochko, Pavel Dvurechensky, Alexander Gasnikov, and Vladislav Lukoshkin. An accelerated method for decentralized distributed stochastic optimization over time-varying graphs. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 3367–3373, 2021.

[82] Ali H. Sayed. Chapter 9 - diffusion adaptation over networks*the work was supported in part by nsf grants eecs-060126, eecs-0725441, ccf-0942936, and ccf-1011918*. In Abdelhak M. Zoubir, Mats Viberg, Rama Chellappa, and Sergios Theodoridis, editors, *Academic Press Library in Signal Processing: Volume 3*, volume 3 of *Academic Press Library in Signal Processing*, pages 323–453. Elsevier, 2014.

[83] Ali H Sayed, Sheng-Yuan Tu, Jianshu Chen, Xiaochuan Zhao, and Zaid J Towfic. Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior. *IEEE Signal Processing Magazine*, 30(3):155–171, 2013.

[84] 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.

[85] 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.

[86] 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.

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

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

[89] 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.

[90] 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.

[91] 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.

[92] 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.

[93] 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.

[94] 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.

[95] 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.

[96] 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.

[97] 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.

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

[99] Siyu Xie and Lei Guo. Analysis of distributed adaptive filters based on diffusion strategies over sensor networks. *IEEE Transactions on Automatic Control*, 63(11):3643–3658, 2018.

[100] 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.

[101] 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] , Supplementary Material
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] , Section 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] , Section 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, we 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]