# PRECISION: Decentralized Constrained Min-Max Learning with Low Communication and Sample Complexities

Zhuqing Liu[+], Xin Zhang[*], Songtao Lu[°], and Jia Liu[+]
[+]Department of Electrical and Computer Engineering, The Ohio State University
[*]Department of Statistics, Iowa State University
[°]IBM Research, Thomas J. Watson Research Center

## ABSTRACT

Recently, min-max optimization problems have received increasing attention due to their wide range of applications in machine learning (ML). However, most existing min-max solution techniques are either single-machine or distributed algorithms coordinated by a central server. In this paper, we focus on the *decentralized* min-max optimization for learning with domain constraints, where multiple agents collectively solve a nonconvex-strongly-concave min-max saddle point problem without coordination from any server. Decentralized min-max optimization problems with domain constraints underpins many important ML applications, including multi-agent ML fairness assurance, and policy evaluations in multi-agent reinforcement learning. We propose an algorithm called PRECISION (proximal gradient-tracking and stochastic recursive variance reduction) that enjoys a convergence rate of $O(1/T)$, where $T$ is the maximum number of iterations. To further reduce sample complexity, we propose PRECISION$^+$ with an adaptive batch size technique. We show that the fast $O(1/T)$ convergence of PRECISION and PRECISION$^+$ to an $\epsilon$-stationary point imply $O(\epsilon^{-2})$ communication complexity and $O(m\sqrt{n}\epsilon^{-2})$ sample complexity, where $m$ is the number of agents and $n$ is the size of dataset at each agent. To our knowledge, this is the first work that achieves $O(\epsilon^{-2})$ in both sample and communication complexities in decentralized min-max learning with domain constraints. Our experiments also corroborate the theoretical results.

## CCS CONCEPTS

• **Computing methodologies** → **Machine learning**.

## KEYWORDS

Decentralized learning, optimization, algorithm design

**ACM Reference Format:**
Zhuqing Liu[+], Xin Zhang[*], Songtao Lu[°], and Jia Liu[+]. 2023. PRECISION: Decentralized Constrained Min-Max Learning with Low Communication and Sample Complexities. In *International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '23), October 23–26, 2023, Washington, DC, USA.* ACM, New York, NY, USA, 25 pages. https://doi.org/10.1145/3565287.3610267


MobiHoc '23, October 23–26, 2023, Washington, DC, USA

ACM ISBN 978-1-4503-9926-5/23/10…$15.00
https://doi.org/10.1145/3565287.3610267

## 1 INTRODUCTION

In recent years, machine learning (ML) has achieved a great success in many areas, including robotics[43], image recognition[35], natural language processing[33], recommender systems[11], to name just a few. Traditionally, the training of ML models is deployed in high-performance computer clusters co-located at large-scale data centers with easy access to big training datasets. However, with more diverse ML applications emerging, the deployment of ML has also been migrating to the edge of computing and communication networks due to the following reasons: First, in many ML applications, data are generated and collected through diverse data sources that are geographically disperse (e.g., smart mobile devices, vehicles, environmental sensors, satellite imagery). Second, because of the limited communication capabilities of the devices and data privacy concerns, it is expensive or even infeasible to send the data collected at the edge networks to the cloud for centralized processing. These real-world limitations have spawned the rapid development of *decentralized learning* over edge networks in recent years, which can leverage highly flexible peer-to-peer edge computing networks with arbitrary topologies [19, 32]. Also, thanks to the resilience to single-point-of-failure, data privacy, and simple implementations, decentralized learning has attracted growing interest recently, and has found various science and engineering applications, e.g., distributed robotics control [39, 55], network resource allocation [16, 40], dictionary learning [8], multi-agent systems [6, 55], multi-task learning [49, 53], and information retrieval [1].

From a mathematical perspective, conducting decentralized learning over a computing network amounts to solving an optimization problem *distributively* and *collaboratively* by a group of agents in the network. However, among the existing literature of decentralized learning, most works are focused on the standard loss minimization formulation, i.e., $\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$, where $f(\cdot)$ denotes the loss objective function of learning and $\mathbf{x}$ denotes the global model parameters to be learned, and $d$ is the model dimension. While this standard loss minimization formulation is sufficiently general to cover a wide range of ML applications (e.g., robotic network [17, 36, 44]), sensor network [9, 34, 38]), power network [5, 10, 12, 13]), it has become increasingly apparent that its mathematical structure is not rich enough to capture new requirements of ever-emerging ML applications. Notably, many sophisticated ML problems nowadays can be expressed as the so-called "min-max" optimization in the form of $\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x},\mathbf{y})$, where $\mathbf{x}$ and $\mathbf{y}$ are both parameters to be learned (may have different dimensionality), and $\mathcal{X}$ and $\mathcal{Y}$ are some conforming real subspaces for $\mathbf{x}$ and $\mathbf{y}$, respectively. Although min-max optimization also has a long history that dates back to 1945 [48], research on *decentralized min-max optimization* remains in its infancy so far and results in this area are surprisingly limited.

In this paper, rather than studying the unstructured general decentralized min-max problems as in [22, 23], we focus on a subclass of interesting decentralized min-max optimization, where multiple agents collectively solve a *domain-constrained* nonconvex-strongly-concave (NCX-SCV) min-max problem. The constraint frequently emerges in various scenarios, such as autonomous driving [3], and safety-constrained[27, 28], etc. The decentralized constrained NCX-SCV min-max problem is important because it arises naturally from many recently emerging multi-agent ML applications, such as multi-agent fairness constraints in adversarial training [51], policy evaluation in multi-agent reinforcement learning (MARL) [37], and multi-agent fairness assurance in ML [2, 41] (see Section 2 for more in-depth discussions).

However, designing effective and efficient algorithms for solving decentralized constrained NCX-SCV min-max problems is highly non-trivial due to the following technical challenges: First, min-max optimization tackles a composition of an inner maximization problem and an outer minimization problem. This tightly coupled inner-outer mathematical structure, together with the decentralized nature and the non-convexity of the outer problem, render the design and theoretical analysis of the algorithms rather difficult. Moreover, the constrained structures in both the inner and outer problems impose yet another layer of challenges in the algorithmic design for decentralized constrained NCX-SCV min-max problems. Second, the decentralization over edge computing networks faces two fundamentally *conflicting* performance metrics. On one hand, due to the high dimensionality of deep learning models and large datasets, it is infeasible to exploit information beyond first-order stochastic gradients to determine search directions in algorithm design. Although the variance of stochastic gradients can be reduced by increasing the number of training samples in mini-batches, doing so incurs higher computational costs for the stochastic gradients. On the other hand, if one uses fewer training samples in each iteration to trade for a lower computational cost, the larger variance in the stochastic gradients inevitably leads to more communication rounds to reach a certain training accuracy (i.e., slower convergence). The high communication complexity is particularly problematic in wireless edge networks, where communication connections could be low-speed and highly unreliable. Third, constrained decentralized min-max optimization presents a significantly greater challenge than its unconstrained counterpart. This is primarily due to the non-smooth nature of the domain constraints and the intricate coupling between these constraints and the min-max problem structure.

The major contribution of this paper is that we propose a series of new algorithmic techniques to address the challenges above and achieve low sample and communication complexities in decentralized constrained NCX-SCV min-max problems. Our main technical results and their significance are summarized as follows:

- We propose a decentralized constrained min-max optimization algorithm called PRECISION (proximal gradient-tracking and stochastic recursive variance reduction) and show that, to achieve an $\epsilon$-stationary point, PRECISION enjoys a convergence rate of $O(1/T)$ ($T$ is the maximum number of iterations). This result further implies an $[O(m\sqrt{n}\epsilon^{-2}), O(\epsilon^{-2})]$ sample-communication complexity scalings, where $m$ is the number of agents, and $n$ is the size of the local dataset at each agent.
- To relax the full gradient evaluation requirement in PRECISION, we propose an enhanced algorithm called PRECISION$^+$ , which is based on an adaptive batch size technique. PRECISION$^+$ further reduces the sample complexity of PRECISION, while retaining the same $[O(m\sqrt{n}\epsilon^{-2}), O(\epsilon^{-2})]$ sample-communication complexity scaling laws as those of PRECISION. Moreover, a lower sample complexity can be obtained in PRECISION$^+$ by slightly trading off its communication complexity (the trade-off is only reflected in the hidden Big-O constants).
- We note that both PRECISION and PRECISION$^+$ algorithms integrate two proximal operators for both the inner and outer constraints (on $\mathbf{x}$ and $\mathbf{y}$), variance reduction techniques for both inner and outer updates, and gradient-tracking-based updates in both inner and outer variables. In this sense, both PRECISION-based algorithms can be viewed as a triple hybrid approach, which necessitates new performance analysis and proof techniques. It is also worth pointing out that the proposed algorithmic and proof techniques in PRECISION could be of independent interest in decentralized min-max learning theory in general.

The rest of the paper is organized as follows. In Section 2, we first provide the preliminaries of the decentralized min-max optimization problems and discuss related works. In Section 3, we propose two stochastic variance reduced algorithms, namely PRECISION and PRECISION$^+$. The convergence rate, communication complexity, and sample complexity of PRECISION and PRECISION$^+$ are also provided in Section 3. Section 4 provides numerical results to verify our theoretical findings, and Section 5 concludes this paper.

## 2 PRELIMINARIES AND RELATED WORK

To facilitate subsequent technical discussions, in Section 2.1, we first provide the basics of decentralized min-max optimization and its consensus formulation. Then, we formally define the notions of sample and communication complexities of the consensus form of decentralized min-max optimization problems. Next, in Section 2.2, we provide an overview of related work of existing optimization algorithms for solving min-max learning problems and their performance in terms of their sample and communication complexities, thus putting our work in comparative perspectives.

### 2.1 Preliminaries of Decentralized Min-Max Optimization

**1) Network Consensus Formulation:** Consider an undirected connected network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ are the sets of nodes (agents) and edges, respectively, with $|\mathcal{N}| = m$. Each agent has local computation capability and is able to communicate with the set of its neighboring agents defined as $\mathcal{N}_i \triangleq \{i' \in \mathcal{N}, : (i, i') \in \mathcal{L}\}$. For presentation simplicity, we assume that each agent $i$ has $n$ data samples and thus there are $mn$ data samples in total[1]. In decentralized min-max optimization, the agents in the network distributively and collaboratively solve the following decentralized min-max optimization problem:

[1]We note that with more complex notation, all our proofs and results continue to hold in cases with unequal sized local datasets.

$$\min_{\mathbf{x}\in\mathcal{X}}\max_{\mathbf{y}\in\mathcal{Y}}\Big[\frac{1}{m}\sum_{i=1}^{m}F_i(\mathbf{x},\mathbf{y})+h(\mathbf{x})\Big], \tag{1}$$

where $\mathbf{x}\in\mathcal{X}$ and $\mathbf{y}\in\mathcal{Y}$ are parameters to be trained for the outer-min and inner-max problems, respectively, the sets $\mathcal{X}\subseteq\mathbb{R}^{p_1}$ and $\mathcal{Y}\subseteq\mathbb{R}^{p_2}$ are closed and convex sets, $F_i(\mathbf{x},\mathbf{y})\triangleq\frac{1}{n}\sum_{j=1}^{n}f_{ij}(\mathbf{x}_i,\mathbf{y}_i|\boldsymbol{\xi}_{ij})$ denotes the local objective function, and $h(\mathbf{x}_i)$ is a proper convex function (possibly non-differentiable) that usually plays the role of regularization. Here, $F_i(\mathbf{x},\mathbf{y})$ is only observable to node $i$ and is assumed to be non-convex with respect to $\mathbf{x}$ for a fixed $\mathbf{y}$, and strongly concave with respect to $\mathbf{y}$ for a fixed $\mathbf{x}$. To solve Problem (1) in a decentralized fashion, a common approach is to rewrite it in the following equivalent form:

$$\min_{\{\mathbf{x}_i\in\mathcal{X},\forall i\}}\max_{\{\mathbf{y}_i\in\mathcal{Y},\forall i\}}\left[\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}f_{ij}(\mathbf{x}_i,\mathbf{y}_i|\boldsymbol{\xi}_{ij})+h(\mathbf{x}_i)\right],$$
$$\text{subject to}\quad \mathbf{x}_i=\mathbf{x}_{i'},\mathbf{y}_i=\mathbf{y}_{i'},\ \forall(i,i')\in\mathcal{L}, \tag{2}$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ are the local copies of the original parameters $\mathbf{x}$ and $\mathbf{y}$ at agent $i$, respectively. The equality constraints in (2) ensure that the local copies at all agents are equal to each other, hence the name "consensus form." Clearly, Problems (1) and (2) share the same solution. In the rest of this paper, we will focus on solving Problem (2), which will be referred to as a decentralized non-convex-strongly-concave (NCX-SCV) consensus min-max optimization problem. The goal of decentralized consensus min-max optimization is to design an algorithm to attain a collective $\epsilon$-stationary point $\{\mathbf{x}_i,\mathbf{y}_i,\forall i\}$ that satisfies the following condition:

$$\underbrace{\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{x}_i-\bar{\mathbf{x}}\|^2}_{\text{Outer consensus error}}+\underbrace{\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{y}_i-\bar{\mathbf{y}}\|^2}_{\text{Inner consensus error}}+\underbrace{\mathbb{E}\|\mathbf{y}^*-\bar{\mathbf{y}}\|^2}_{\text{Saddle point error}}+\underbrace{\left\|\frac{1}{m}\sum_{i=1}^{m}\nabla_{\mathbf{x}}F_i(\mathbf{x},\mathbf{y})\right\|^2}_{\text{Global gradient norm}}\leq\epsilon^2,$$

where $\bar{\mathbf{x}}\triangleq\frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i$, $\bar{\mathbf{y}}\triangleq\frac{1}{m}\sum_{i=1}^{m}\mathbf{y}_i$, and $\mathbf{y}^*$ represents the maximizer point of $F$ over $\mathbf{y}$, where $\mathbf{y}^*(\bar{\mathbf{x}})\in\arg\max_{\mathbf{y}\in\mathcal{Y}}F(\bar{\mathbf{x}},\mathbf{y})$,

As mentioned in Section 1, two of the most important performance metrics in decentralized optimization are the sample and communication complexities. In this paper, we adopt two definitions of sample and communication complexities that are widely used in the decentralized optimization literature (e.g., [45]) to measure the efficiency of our algorithms:

Definition 1 (Sample Complexity). *The sample complexity is defined as the total number of incremental first-order oracle (IFO) calls required across all nodes until an algorithm converges to an $\epsilon$-stationary point, where one IFO call evaluates a pair of gradients $(\nabla_{\mathbf{x}}f_{ij}(\mathbf{x},\mathbf{y}),\nabla_{\mathbf{y}}f_{ij}(\mathbf{x},\mathbf{y}))$ at node $i$.*

Definition 2 (Communication Complexity). *Let a round of communications be a time window during which each node sends a vector to its neighboring nodes while receiving a set of vectors from all its neighboring nodes. Then, the communication complexity is defined as the total number of rounds of communications required until an algorithm converges to an $\epsilon$-stationary point.*

**2) Motivating Application Examples:** With the basics of decentralized constrained NCX-SCV min-max optimization, we provide two examples to further motivate its practical relevance:

- *Multi-Agent Fair ML:* Consider a machine learning task with dataset $\{b_{ij},[\tilde{\boldsymbol{\xi}}_{ij}^{\top},\boldsymbol{\xi}_{ij}^{*\top}]^{\top}\}$ over a multi-agent network, where $b_{ij}$ is the observed label of the $j$-th sample at the $i$-th agent, $\tilde{\boldsymbol{\xi}}_{ij}\in\mathbb{R}^{d_1}$ denotes the corresponding nonsensitive features and $\boldsymbol{\xi}_{ij}\in\mathbb{R}^{d_2}$ represents the sensitive features. In the problem of Fair ML, fairness is imposed by adding a regularization term that penalizes the statistical correlation between the learning model output $\hat{b}_{ij}$ and the sensitive attributes $\boldsymbol{\xi}_{ij}^*$. In binary case, one example is the Renyi correlation [2] as a regularization to impose fairness, under which the multi-agent fair ML problem can be written as a decentralized NCX-SCV min-max problem [2]: $\min_{\mathbf{x}\in\mathcal{X}}\max_{\mathbf{y}\in\mathcal{Y}}\mathbb{E}_i\big[\mathbb{L}(F_i(\mathbf{x},\mathbf{y}|\boldsymbol{\xi}_i),b_i)-\lambda_l\sum_{j=1}^{c}y_{ij}^2\mathbf{f}_{ij}(\mathbf{x}_i,\boldsymbol{\xi}_i)+\lambda_l\cdot\sum_{j=1}^{c}y_{ij}\tilde{S}\mathbf{f}_{ij}(\mathbf{x},\boldsymbol{\xi}_i)\big]$, where $\tilde{S}=2S-1$, $S=\{0,1\}$, denotes the sensitive attribute, $\mathbb{L}$ is the loss function, $\lambda_l$ is a positive scalar balancing fairness and goodness-of-fit, $c$ is the class label and $\mathbf{f}_{ij}(\mathbf{x},\boldsymbol{\xi}_i)$ represents the vector-valued output of a neural network after soft-max layer.
- *Data Poisoning Attack:* Consider a decentralized learning problem with $m$ agents trying to learn a common model. An adversary has the ability to inject noise into the training samples of a subset of agents. Let $\mathbf{y}_i$ denote the model parameter and let $\mathbf{x}_i$ denote the injected poisoned data parameter. In this problem, the adversary tries to maximize the loss function while the other agents aim at minimizing the loss function. Thus, the data poisoning attack problem has the following NCX-SCV min-max problem: $\max_{\mathbf{x}\in\mathcal{X}}\min_{\mathbf{y}\in\mathcal{Y}}\sum_{i=1}^{m}\frac{1}{|\boldsymbol{\xi}_i|}\sum_{\ell\in\boldsymbol{\xi}_i}\log\big(1+\exp\big((-v_\ell\mathbf{y}_i^T(w_\ell+\mathbf{x}_i)\big)\big)$, where $v_\ell\in\mathbb{R}$ and $w_\ell\in\mathbb{R}^d$ denote the $\ell$-th data point's label and the feature vector, respectively.

## 2.2 Related Work

**1) Centralized NCX-SCV Min-Max Optimization:** In the literature, the state-of-the-art algorithms for solving NCX-SCV optimization problems in the centralized setting are GDA [20], min-max-PPA [21], and SREDA [26]. Specifically, Lin et al. [20] proposed a gradient-based GDA method to find a first-order Nash equilibrium point. In each iteration, GDA performs gradient descent over the $\mathbf{x}$-variable and gradient ascent over the $\mathbf{y}$-variable. GDA has an $O(1/T)$ convergence rate for NCX-SCV min-max optimization problems, where $T$ is the maximum number of iterations. Also, it requires a full gradient evaluation in each iteration, which implies an $O(n\epsilon^{-2})$ sample complexity to achieve an $\epsilon$ convergence error. The Minimax-PPA method is proposed in [21] to solve NCX-NCV problem and achieves an $\tilde{O}\left(n\varepsilon^{-2}\right)$ sample complexity. These methods have a high sample complexity in the big-data regime with a large $n$. To overcome this issue, several variance reduction methods have also been proposed. For example, in [26], a variance reduction algorithm named SREDA is proposed, which is further enhanced by [52] to allow a larger step-size. SREDA achieves an $\tilde{O}\left(n+\sqrt{n}\epsilon^{-2}\right)$ sample complexity for large $n$, thus having a lower sample complexity than GDA and minimax-PPA. However, SREDA can only handle min-max problems with constraints on $\mathbf{x}$ but not on $\mathbf{y}$. We

summarize the above comparisons in Table 1. While the above algorithms achieve varying degrees of success in solving NCX-SCV min-max problems, they are developed for the centralized setting, which is fundamentally different from our work.

**Table 1: Comparisons among algorithms for NCX-SCV min-max problems ($m$ is the number of agents, $n$ is the size of dataset for each agent, and $\epsilon$ is the convergence error. Our proposed algorithms are marked in bold.**

| Algorithm* | Proximal Operator | Sample Complex. | Commun. Complex. | Decentralized |
|---|---|---|---|---|
| GDA [20] | $\mathbf{y}$ | $\tilde{O}\left(n\varepsilon^{-2}\right)$ | - | ✗ |
| Minmax-PPA [21] | $\mathbf{x}$ and $\mathbf{y}$ | $\tilde{O}\left(n\varepsilon^{-2}\right)$ | - | ✗ |
| SREDA [26] | $\mathbf{x}$ | $\tilde{O}\left(n+\sqrt{n}\varepsilon^{-2}\right)$ | - | ✗ |
| **PRECISION PRECISION$^+$** | $\mathbf{x}$ and $\mathbf{y}$ | $O(m\sqrt{n}\epsilon^{-2})$ | $O(\epsilon^{-2})$ | ✓ |

**2) Decentralized Min-Max Optimization:** As mentioned in Section 1, existing results on decentralized min-max optimization are quite limited. The earliest attempt is the CSPSG method [29], which considered the most ideal convex-concave (CX-CV) setting. Due to its simplistic SGD-type updates, CSPSG has high sample and communication complexities of $O(\epsilon^{-4})$. DPOSG [22] considered unstructured nonconvex-nonconave (NCX-NCV) unconstrained decentralized min-max problems in the context of large-scale GANs, and proposed to leverage the classical DSGD [32] approach to decentralize the centralized counterpart algorithm called OGDA [30]. Due to the limitations inherent in DSGD, DPOSG suffers from a high sample complexity of $O(\epsilon^{-12})$. In contrast, DPPSP [23] also studied unstructured NCX-NCV decentralized min-max optimization problems with constraints. Due to the use of basic proximal SGD-type updates, DPPSP also suffers high sample and communication complexities of $O(\epsilon^{-4})$.

Compared to the simplistic algorithmic techniques in [22, 23], our PRECISION algorithms is a triple hybrid algorithm that integrates proximal operators, variance reductions, and gradient tracking, thus achieving much lower sample and communication complexities. We note that although our significantly lower sample and communication complexities are achieved under the more structured NCX-SCV setting, we believe our techniques can also be applied to NCX-NCV to improve the sample and communication complexities of existing works. This will be left in our future work.

The most related work to ours is GT-GDA [46], which also studied constrained decentralized NCX-SCV min-max optimization. The key difference between GT-GDA and our work is that only one constraint set is imposed on either $\mathbf{x}$ or $\mathbf{y}$, but not on both. In contrast, we consider the more complex case where both $\mathbf{x}$ and $\mathbf{y}$ are constrained. GT-GDA also requires several inner updates for $\mathbf{y}$ and then performs one update for $\mathbf{x}$, which is similar to alternating direction method of multipliers [4] (ADMM) update scheme. Also, our algorithms achieve a lower sample complexity $O(m\sqrt{n}\epsilon^{-2})$ than that of $O(mn\epsilon^{-2})$ in GT-GDA. To conclude this section, we summarize the above comparisons in Table 2. Another closely related work can be found in [54], where the authors developed a

**Table 2: Comparisons among algorithms for decentralized min-max problems.**

| Algorithm* | Proximal Operator | Sample Complex. | Commun. Complex. | Problem |
|---|---|---|---|---|
| DPOSG [22] | - | $O(\epsilon^{-12})$ | $O(\log(1/\epsilon))$ | NCX-NCV |
| CSPSG [29] | $\mathbf{x}$ and $\mathbf{y}$ | $O(\epsilon^{-4})$ | $O(\epsilon^{-4})$ | CX-CV |
| DPPSP [23] | $\mathbf{x}$ and $\mathbf{y}$ | $O(\epsilon^{-4})$ | $O(\epsilon^{-4})$ | NCX-NCV |
| GT-GDA [46] | $\mathbf{x}$ or $\mathbf{y}$ | $O(mn\epsilon^{-2})$ | $O(\epsilon^{-2})$ | NCX-SCV |
| **PRECISION PRECISION$^+$** | $\mathbf{x}$ and $\mathbf{y}$ | $O(m\sqrt{n}\epsilon^{-2})$ | $O(\epsilon^{-2})$ | NCX-SCV |

decentralized optimization method for a multi-agent reinforcement learning policy evaluation problem based on the mean squared projected Bellman error (MSPBE), which can be formulated as a finite-sum minimax problem. However, our work differs from [54] in the following aspects: (i) Unlike [54], our method can handle *non-smooth* objectives. However, the direct proximal extension of the algorithm in [54] may diverge in solving the decentralized problem [14]. To this end, we propose a specialized proximal operator $\tilde{\mathbf{x}}_i\left(\mathbf{x}_{i,t}\right)$ to address this challenge, see detailed discussions in our Remark 1; (ii) Our approach addresses *general* decentralized min-max optimization problems, while [54] is limited to RL policy evaluation.

## 3 SOLUTION APPROACH

In this section, we first present our PRECISION and PRECISION$^+$ algorithms in Sections 3.1 and 3.2, respectively. Then, we provide the main theoretical results and the key insights of the PRECISION and PRECISION$^+$ algorithms in Section 3.3. Due to space limitation and for better readability, we relegate some proof details of the theoretical results to our online technical report [24].

### 3.1 The PRECISION Algorithm

To solve the consensus form of decentralized min-max problem in Problem (2), we adopt the network consensus mixing approach in the literature [32]. Toward this end, we let $\mathbf{M} \in \mathbb{R}^{m\times m}$ denote the consensus weight matrix and let $[\mathbf{M}]_{ii'}$ denote the element in the $i$-th row and the $i'$-th column in $\mathbf{M}$. $\mathbf{M}$ satisfies the following properties [32, 47]:

(a) *Doubly stochastic:* $\sum_{i=1}^{m}[\mathbf{M}]_{ii'} = \sum_{i'=1}^{m}[\mathbf{M}]_{ii'} = 1$;
(b) *Symmetric:* $[\mathbf{M}]_{ii'} = [\mathbf{M}]_{i'i}, \forall i, i' \in \mathcal{N}$;
(c) *Network-Defined Sparsity:* $[\mathbf{M}]_{ii'} > 0$ if $(i, i') \in \mathcal{L}$; otherwise $[\mathbf{M}]_{ii'} = 0, \forall i, i' \in \mathcal{N}$.

Note that the above properties imply that the eigenvalues of $\mathbf{M}$ are real and can be sorted as $-1 < \lambda_m(\mathbf{M}) \leq \cdots \leq \lambda_2(\mathbf{M}) < \lambda_1(\mathbf{M}) = 1$. For notational convenience, we define the second-largest eigenvalue in magnitude of $\mathbf{M}$ as $\lambda \triangleq \max\{|\lambda_2(\mathbf{M})|, .., |\lambda_m(\mathbf{M})|\}$, which will play an important role in the step-size selection and analysis of the algorithm's convergence rate. With the above notation, we are now in a position to describe our proposed algorithms.

As mentioned in Section 1, our PRECISION algorithm can be viewed as a triple hybrid of proximal, gradient tracking, and variance reduction techniques. Next, we will see that these techniques can be organized into three key algorithmic steps:

- *Step 1 (Local Proximal Operations):* In each iteration $t$, each agent $i$ first performs the following proximal operations to cope with the constraint sets $\mathcal{X}$ and $\mathcal{Y}$ for the outer and inner variables, respectively:

$$\tilde{\mathbf{x}}_i(\mathbf{x}_{i,t}) = \arg\min_{\mathbf{x}_i \in \mathcal{X}} \langle \mathbf{p}_{i,t}, \mathbf{x}_i - \mathbf{x}_{i,t} \rangle + \frac{\tau}{2}\|\mathbf{x}_i - \mathbf{x}_{i,t}\|^2 + h(\mathbf{x}_i), \tag{3}$$

$$\tilde{\mathbf{y}}_i(\mathbf{y}_{i,t}) = \arg\min_{\mathbf{y}_i \in \mathcal{Y}} \left\|\mathbf{y}_i - \left(\mathbf{y}_{i,t} + \alpha \mathbf{d}_{i,t}\right)\right\|^2, \tag{4}$$

where $\mathbf{p}_{i,t}$ and $\mathbf{d}_{i,t}$ are two auxiliary vectors for gradient tracking purposes and will be defined shortly, $\tau > 0$ is a constant proximal control parameter, and $\alpha > 0$ is a constant parameter to control the magnitude of the updates of $\mathbf{y}$.

- *Step 2 (Consensus Update):* Next, each agent $i$ updates the outer and inner model parameters $\mathbf{x}_i, \mathbf{y}_i$:

$$\mathbf{x}_{i,t+1} = \underbrace{\sum_{i' \in \mathcal{N}_i} [\mathbf{M}]_{ii'} \mathbf{x}_{i',t}}_{\text{(a)}} + \underbrace{\nu \left(\tilde{\mathbf{x}}_i(\mathbf{x}_{i,t}) - \mathbf{x}_{i,t}\right)}_{\text{(b)}}, \tag{5}$$

$$\mathbf{y}_{i,t+1} = \underbrace{\sum_{i' \in \mathcal{N}_i} [\mathbf{M}]_{ii'} \mathbf{y}_{i',t}}_{\text{(a)}} + \underbrace{\eta (\tilde{\mathbf{y}}_i(\mathbf{y}_{i,t}) - \mathbf{y}_{i,t})}_{\text{(b)}}, \tag{6}$$

where $\nu$ and $\eta$ are the step-sizes for updating $\mathbf{x}$- and $\mathbf{y}$-variables, respectively. Note that in (5) and (6), component $(a)$ is a local weighted average at agent $i$, which is also referred to as "consensus step," and component $(b)$ performs a local update in the spirit of Frank-Wolfe given the proximal points $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$, which is different from the conventional decentralized stochastic gradient updates [31].

- *Step 3 (Local Gradient Estimate):* In the next step, each agent $i$ estimates its local gradients using the following gradient estimators:

$$\mathbf{v}_{i,t} = \begin{cases} \nabla_{\mathbf{x}} F_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}), & \text{if } \mathrm{mod}(t,q) = 0, \\ \mathbf{v}_{i,t-1} + \frac{1}{|\mathcal{S}_{i,t}|} \sum_{j \in \mathcal{S}_{i,t}} \big(\nabla_{\mathbf{x}} f_{ij}(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) & \\ \quad - \nabla_{\mathbf{x}} f_{ij}(\mathbf{x}_{i,t-1}, \mathbf{y}_{i,t-1})\big), & \text{o.w.} \end{cases} \tag{7a}$$

$$\mathbf{u}_{i,t} = \begin{cases} \nabla_{\mathbf{y}} F_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}), & \text{if } \mathrm{mod}(t,q) = 0, \\ \mathbf{u}_{i,t-1} + \frac{1}{|\mathcal{S}_{i,t}|} \sum_{j \in \mathcal{S}_{i,t}} \big(\nabla_{\mathbf{y}} f_{ij}(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) & \\ \quad - \nabla_{\mathbf{y}} f_{ij}(\mathbf{x}_{i,t-1}, \mathbf{y}_{i,t-1})\big), & \text{o.w.} \end{cases} \tag{7b}$$

Here, $\mathcal{S}_{i,t}$ is the sample mini-batch in the $t$-th iteration, and $q$ is a preset inner loop iteration number.

- *Step 4 (Gradient Tracking): Each agent $i$ updates $\mathbf{p}_i$ and $\mathbf{d}_i$ by averaging over its neighboring tracked gradients:*

$$\begin{cases} \mathbf{p}_{i,t} = \sum_{i' \in \mathcal{N}_i} [\mathbf{M}]_{ii'} \mathbf{p}_{i',t-1} + \mathbf{v}_{i,t} - \mathbf{v}_{i,t-1}, \\ \mathbf{d}_{i,t} = \sum_{i' \in \mathcal{N}_i} [\mathbf{M}]_{ii'} \mathbf{d}_{i',t-1} + \mathbf{u}_{i,t} - \mathbf{u}_{i,t-1}. \end{cases} \tag{8}$$

Our PRECISION algorithm can be intuitively understood as follows: In PRECISION, each agent conducts both descent and ascent steps, since Problem (2) minimizes over $\mathbf{x}$ and maximizes over $\mathbf{y}$. Note that $\mathbf{v}_{i,t}$ and $\mathbf{u}_{i,t}$ in (7) only contain the gradient information of the local objective function $F_i(\mathbf{x}, \mathbf{y})$. Merely updating with directions $\mathbf{v}_{i,t}$ and $\mathbf{u}_{i,t}$ cannot guarantee the convergence of the global objective function $F(\mathbf{x}, \mathbf{y})$. Therefore, we introduce two auxiliary variables $\mathbf{p}_{i,t}$ and $\mathbf{d}_{i,t}$ for global gradient tracking purposes. As each agent $i$ updates these two variables by performing the local weighted aggregation shown in (8), $\mathbf{p}_{i,t}$ and $\mathbf{d}_{i,t}$ track the directions of the global gradients.

**Algorithm 1** PRECISION/PRECISION$^+$ at Agent $i$.

If PRECISION : $|\mathcal{R}_{i,t}| = n$;
If PRECISION$^+$ :

$$|\mathcal{R}_{i,t}| = \min\{c_\gamma \sigma^2 (\gamma_t)^{-1}, c_\epsilon \sigma^2 \epsilon^{-1}, n\}.$$

1: Set prime-dual parameter pair $(\mathbf{x}_{i,0}, \mathbf{y}_{i,0}) = (\mathbf{x}^0, \mathbf{y}^0)$.
2: Draw $\mathcal{R}_{i,0}$ samples without replacement and calculate local stochastic gradient estimators as

$$\mathbf{p}_{i,0} = \mathbf{v}_{i,0} = \frac{1}{|\mathcal{R}_{i,0}|} \sum_{j \in \mathcal{R}_{i,0}} \nabla_{\mathbf{x}} f_{ij}(\mathbf{x}_{i,0}, \mathbf{y}_{i,0});$$

$$\mathbf{d}_{i,0} = \mathbf{u}_{i,0} = \frac{1}{|\mathcal{R}_{i,0}|} \sum_{j \in \mathcal{R}_{i,0}} \nabla_{\mathbf{y}} f_{ij}(\mathbf{x}_{i,0}, \mathbf{y}_{i,0});$$

3: **for** $t = 1, \cdots, T$ **do**
4: Update local parameters $(\mathbf{x}_{i,t+1}, \mathbf{y}_{i,t+1})$ as in Eq. (3)-(6);
5: Compute local estimators $(\mathbf{v}_{i,t+1}, \mathbf{u}_{i,t+1})$ as in Eq. (7);
6: Track global gradients $(\mathbf{p}_{i,t+1}, \mathbf{d}_{i,t+1})$ as in Eq. (8);
7: **end for**

It is insightful to compare PRECISION with our most related work, the GT-GDA method in [46]. In GT-GDA, agent $i$ computes the local *full gradients* in the $t$-th iteration as follows:

$$\mathbf{v}_{i,t} = \nabla_{\mathbf{x}} F_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}), \quad \mathbf{u}_{i,t} = \nabla_{\mathbf{y}} F_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}). \tag{9}$$

Different from GT-GDA [46], PRECISION estimates the local gradients in Eq. (7) at agent $i$. In Eq. (7), the algorithm evaluates a full gradient $\nabla F_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$ only every $q$ steps. For other iterations with $\mathrm{mod}(t,q) \neq 0$, PRECISION uses local stochastic gradients estimated by a mini-batch $\frac{1}{|\mathcal{S}_{i,t}|} \sum_{j \in \mathcal{S}_{i,t}} \nabla_{\mathbf{y}} f_{ij}(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$ and a recursive correction term $\mathbf{u}_{i,t-1} - \frac{1}{|\mathcal{S}_{i,t}|} \sum_{j \in \mathcal{S}_{i,t}} \nabla_{\mathbf{y}} f_{ij}(\mathbf{x}_{i,t-1}, \mathbf{y}_{i,t-1})$. Thanks to the periodic full gradients and recursive correction terms, PRECISION is able to achieve a convergence rate of $O(1/T)$. Moreover, due to the stochastic subsampling of $\mathcal{S}_{i,t}$, PRECISION has a *lower* sample complexity than GT-GDA [46]. The full description of PRECISION is shown in Algorithm 1.

## 3.2 The PRECISION$^+$ Algorithm

Note that in PRECISION, full gradients are required for every $q$ steps, which may still incur high computational costs in some situations. Also, in the initialization phase of PRECISION (before the main loop), agents need to evaluate full gradients, which could be time-consuming. To address these challenges, we enhance the PRECISION with an adaptive batch size technique, and this enhanced version is called PRECISION $^+$. Specifically, we modify the gradient estimators in (7a) and (7b) in iteration $t$ with $\mathrm{mod}(t,q) = 0$ as follows :

$$\mathbf{v}_{i,t} = \frac{1}{|\mathcal{R}_{i,t}|} \sum_{j \in \mathcal{R}_{i,t}} \nabla_{\mathbf{x}} f_{ij}(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}), \tag{10}$$

$$\mathbf{u}_{i,t} = \frac{1}{|\mathcal{R}_{i,t}|}\sum_{j\in\mathcal{R}_{i,t}} \nabla_{\mathbf{y}} f_{ij}(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}), \tag{11}$$

where $\mathcal{R}_{i,t}$ is a subsample set (sampling without replacement), whose size is chosen as

$$|\mathcal{R}_{i,t}| = \min\{c_\gamma \sigma^2 (\gamma_t)^{-1}, c_\epsilon \sigma^2 \epsilon^{-1}, n\}. \tag{12}$$

Here, $c_\gamma$ and $c_\epsilon$ are problem-dependent constants to be defined later, $\sigma^2$ is the variance bound of data heterogeneity across agents (also defined later), and $\gamma_{t+1} \triangleq \frac{1}{q}\sum_{i=(n_t-1)q}^{t} \|\tilde{\mathbf{x}}_t - 1\otimes\bar{\mathbf{x}}_t\|^2$, where $\otimes$ represents the Kronecker product operator.

The selection of $|\mathcal{R}_{i,t}|$ is motivated by the fact that the periodic full gradient evaluation only plays an important role in the later stage of the convergence process: in the later stage of the convergence process, we need more accurate update direction. Later, we will see that under some mild assumptions and parameter settings, PRECISION$^+$ has the same convergence rate as that of PRECISION. The full description of the PRECISION$^+$ algorithm is also illustrated in Algorithm 1.

## 3.3 Theoretical Results of the PRECISION and PRECISION$^+$ Algorithms

Before presenting the theoretical results of our algorithms, we first state the following assumptions:

Assumption 1 (Global Objective). *The functions $F(\mathbf{x},\mathbf{y}) = \frac{1}{m}\sum_{i=1}^m [F_i(\mathbf{x}_i,\mathbf{y}_i)]$ and $J(\mathbf{x}) = \max_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{x},\mathbf{y})$ satisfy:*

- *(a) (Boundness from Below): There exists a finite lower bound $Q^* = Q(\mathbf{x}^*) = \inf_{\mathbf{x}}(J(\mathbf{x}) + h(\mathbf{x})) > -\infty$;*
- *(b) (Strong Concavity in $\mathbf{y}$): Local objective function $F_i(\mathbf{x},\cdot)$ is $\mu$-strongly concave for fixed $\mathbf{x}\in\mathbb{R}^{p_1}$, i.e., there exists a positive constant $\mu$ such that $\|\nabla_{\mathbf{y}} F_i(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{y}} F_i(\mathbf{x},\mathbf{y}')\| \geq \mu\|\mathbf{y}-\mathbf{y}'\|, \forall\, \boldsymbol{x},\mathbf{y},\mathbf{y}'\in\mathbb{R}^{p_2}, i\in[m]$.*
- *(c) (Bounded Gradient at Maximum): The partial gradient at every $(\mathbf{x}, \nabla_{\mathbf{x}} F(x,\mathbf{y}^*(\mathbf{x})))$ pair is bounded, i.e., $\|\nabla_{\mathbf{x}} F(\mathbf{x},\mathbf{y}^*(\mathbf{x}))\| < \infty$, $\forall\,\mathbf{x}\in\mathbb{R}^{p_1}$.*

Assumptions 1(a) and 1(b) are standard in the literature. Assumption 1(c) guarantees that $\nabla J(\mathbf{x}) = \nabla_{\mathbf{x}} F(\mathbf{x},\mathbf{y}^*(\mathbf{x}))$.

Assumption 2 (Lipschitz Smoothness of Local Objectives). *The function $f_{ij}(\mathbf{x},\cdot)$ is $L_f$-Lipschitz smooth, i.e., there exists a constant $L_f > 0$, such that $\nabla f_{ij}(\mathbf{x},\mathbf{y}) = [\nabla_{\mathbf{x}} f_{ij}(\mathbf{x},\mathbf{y}]^\top, \nabla_{\mathbf{y}} f_{ij}(\mathbf{x},\mathbf{y})^\top)^\top$ satisfies $\|\nabla f_{ij}(\mathbf{x},\mathbf{y}) - \nabla f_{ij}(\mathbf{x}',\mathbf{y}')\|^2 \leq L_f^2\|\mathbf{x}-\mathbf{x}'\|^2 + L_f^2\|\mathbf{y}-\mathbf{y}'\|^2$, $\forall\,\mathbf{x},\mathbf{x}'\in\mathcal{X},\mathbf{y},\mathbf{y}'\in\mathcal{Y}, i\in[m], j\in[n]$.*

Further, we have the following assumption only for the algorithm PRECISION$^+$:

Assumption 3 (Bounded Variance). *There exists a constant $\sigma^2 > 0$, such that $\mathbb{E}\|\nabla f_{ij}(\mathbf{x},\mathbf{y}) - \nabla F_i(\mathbf{x},\mathbf{y})\|^2 \leq \sigma^2, \forall\,\mathbf{x},\mathbf{y},\in\mathbb{R}^p, i\in[m], j\in[n]$.*

To address the challenges in characterizing the convergence rate for NCX-SCV decentralized constrained min-max problems, we propose the following *new* metric, which is the key to the success of establishing all convergence results in this paper:

$$\begin{aligned}\mathfrak{M}_t \triangleq& \mathbb{E}[\|\tilde{\mathbf{x}}_t - 1\otimes\bar{\mathbf{x}}_t\|^2 + \|\mathbf{x}_t - 1\otimes\bar{\mathbf{x}}_t\|^2 \\ &+ \|\mathbf{y}_t - 1\otimes\bar{\mathbf{y}}_t\|^2 + \|\mathbf{y}_t^* - \bar{\mathbf{y}}_t\|^2],\end{aligned} \tag{13}$$

where $\mathbf{y}_t^*$ denotes $\mathbf{y}^*(\bar{\mathbf{x}}_t) = \arg\max_{y\in\mathbb{R}^p} F(\bar{\mathbf{x}}_t,\mathbf{y})$. The first two terms in (13) are inspired by the metric in SONATA [42], which measures the converging progress of non-convex decentralized *minimization* problems (not min-max). The third term in (13) measures the consensus error of local copies on $\mathbf{y}$. The fourth term in (13) quantifies $\bar{\mathbf{y}}_t$'s convergence to the point $\mathbf{y}_t^*$ for $F(\bar{\mathbf{x}}_t,\cdot)$. Thus, as $\mathfrak{M}_t \to 0$, we have that the algorithm reaches a consensus on a first-order stationary point (FOSP) of the original decentralized constrained min-max optimization problem.

With the metric in (13), the convergence rates of algorithms PRECISION /PRECISION$^+$ can be characterized as follows:

Theorem 1 (Convergence of PRECISION). *Under Assumption 1 (a)-(d) and Assumption 2, suppose that $\beta \leq \min\left\{\frac{\tau}{12}, \frac{1}{3}\right\}$, $\alpha \leq \frac{1}{4L_f}, q = |\mathcal{S}_{i,t}| = \lceil\sqrt{n}\rceil$ hold and let $c_1 = \frac{1-\lambda^2}{1+\lambda^2}$, if the step-sizes satisfy: $\eta \leq \min\left\{\frac{1}{8}, \frac{c_1 m\mu}{375\alpha L_f^2}, \frac{15L_f^2}{\beta\mu\alpha^2 c_1}, \frac{3c_1^2 m}{10(1+c_1)\mu\alpha}\right\}, \nu \leq \min\left\{\frac{c_1 m\beta}{40L_f^2}, \frac{2c_1 m\beta}{5\tau}, \frac{2c_1\beta\mu^2 m}{375L_f^4}, \frac{5\tau}{3mc_1}, \frac{\tau}{6m(1+1/c_1)}, \frac{3\mu\eta\alpha\tau}{17L_f^2}, \frac{\tau}{3(L_f+L_f^2/\mu)}\right\}$, then the following convergence result for the* PRECISION *algorithm holds:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\mathfrak{M}_t] \leq \frac{\mathbb{E}[\mathfrak{p}_0 - Q^*]}{\min\{C_1, C_2, C_3, \nu L_f^2/2\}(T+1)},$$

*where $Q^* = Q(\mathbf{x}^*)$ and $\mathfrak{p}_t$ is a potential function defined as:*

$$\begin{aligned}\mathfrak{p}_t \triangleq Q(\bar{\mathbf{x}}_t) + \frac{4\nu L_f^2}{\beta\mu\eta^2}\|\bar{\mathbf{y}}_t - \mathbf{y}_t^*\|^2 \\ + \frac{1}{m}\sum_{i=1}^m[\|\mathbf{x}_{i,t} - \bar{\mathbf{x}}_t\|^2 + \|\mathbf{y}_{i,t} - \bar{\mathbf{y}}_t\|^2],\end{aligned} \tag{14}$$

*and $C_1, C_2, C_3 \geq 0$ are constants. Due to space limitation, detailed definition of these constants are relegated to our technical report [24]. Also, in (14), $Q(\mathbf{x}_t) \triangleq \max_{\mathbf{y}} F(\mathbf{x}_t,\mathbf{y}) + h(\mathbf{x}_t)$, and $\mathbf{y}_t^* = \arg\max_{\mathbf{y}} F(\mathbf{x}_t,\mathbf{y})$.*

Theorem 2 (Convergence of PRECISION$^+$). *Under Assumption 1 (a)-(d), Assumptions 2-3, and the same parameter settings as in Theorem 1, with additional parameters $c_\gamma$ and $c_\epsilon$ satisfying the conditions:*

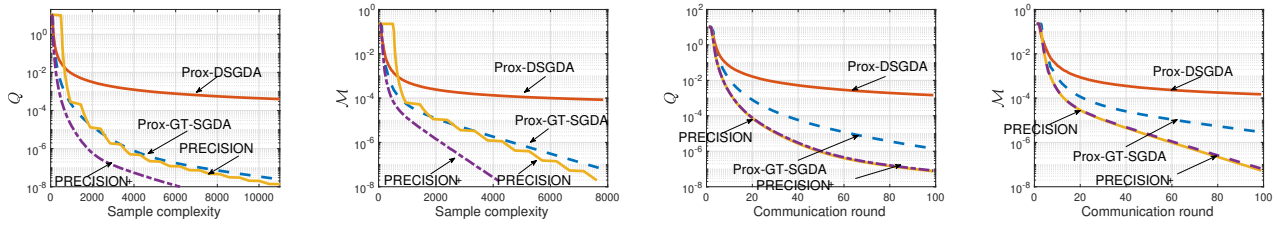
$$c_\gamma \geq \left(\frac{75\eta\alpha}{8\mu}\frac{1}{m} + \frac{\nu}{\beta}\frac{1}{m}\right)\frac{\nu\tau}{12}, \quad c_\epsilon > 0, \tag{15}$$

*and the potential function as stated in Theorem 1, the following convergence result for* PRECISION$^+$ *holds:*

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\mathfrak{M}_t] \leq& \frac{\mathbb{E}[\mathfrak{p}_0 - Q^*]}{(T+1)\min\{C_1, C_2', C_3, \nu L_f^2/2\}} \\ &+ \left(\frac{75\eta\alpha}{16\mu}\frac{2}{m} + \frac{\nu}{2\beta}\frac{2}{m}\right)\frac{\epsilon}{c_\epsilon},\end{aligned} \tag{16}$$

*where the constant $C_2' \geq$ and the definition of $C_2'$ is relegated to our technical report [24].*

Remark 1. Compared to existing works on decentralized min-max optimization[46, 54], it is worth noting that the main difficulty in establishing convergence results in Theorem 1 and Theorem 2 arises from the proximal operator in the outer-level subproblem. This operator precludes the use of conventional descent lemmas




(a) Loss function value vs. sample complexity. (b) Convergence metric vs. sample complexity. (c) Loss function value vs. communication complexity. (d) Convergence metric vs. communication complexity.

**Figure 1: Comparisons of algorithms for decentralized NCX-SCV min-max optimization problems.**

for convergence analysis, as outlined in Lemma 3 in the Appendix. Furthermore, unlike in single-agent constrained bilevel optimization, the direct proximal extension of the algorithm in [14] ($\widetilde{\mathbf{x}}_{i,t} = \arg\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - (\mathbf{x}_{i,t} - \tau \mathbf{p}_{i,t})\|^2$) will diverge for the decentralized constrained min-max problem in this paper. To address this challenge, we employ a special proximal update rule in (3). The proximal operator $\tilde{\mathbf{x}}_{i,t}$ in (3), consensus updating (5), and the corresponding local update (5) are the key in addressing the *non-smooth* objective challenge encountered in decentralized learning.

Remark 2. In Theorems 1 and 2, the step-sizes and convergence rates depend on the network topology. For a sparse network, $\lambda$ is close to (but not exactly) one (recall that $\lambda = \max\{|\lambda_2|, |\lambda_m|\} < 1$), the step-size needs to be smaller as $\lambda$ gets close to one, which leads to a slower convergence. Additionally, the convergence performance of PRECISION$^+$ is affected by constant $(\frac{75\eta\alpha}{16\mu}\frac{2}{m} + \frac{\nu}{2\beta}\frac{2}{m})\frac{\epsilon}{c_\epsilon}$, which depends on the inexact gradient estimation at the $t$-th iteration with $\text{mod}(t, q) = 0$. Intuitively, a larger value of $c_\epsilon$ allows us to use a larger batch size as shwon in (12), which in turn leads to faster convergence. Theoretically, we can observe that a larger value of $c_\epsilon$ results in a smaller constant $(\frac{75\eta\alpha}{16\mu}\frac{2}{m} + \frac{\nu}{2\beta}\frac{2}{m})\frac{\epsilon}{c_\epsilon}$ in (16), thereby yielding a more accurate estimation.

Following from Theorems 1 and 2, we immediately have the sample and communication complexity results for the PRECISION and PRECISION$^+$ algorithms:

Corollary 3. *Under the conditions in Theorems 1 and 2, and with $q = \sqrt{n}$, to achieve an $\epsilon$-stationary solution, the following results for the* PRECISION *and* PRECISION$^+$ *algorithms hold:*

- *Communication Complexity: the numbers of total communication rounds are upper bounded by $O(\epsilon^{-2})$*
- *Sample Complexity: The total samples evaluated across the network are upper bounded by $O(m\sqrt{n}\epsilon^{-2})$).*

Remark 3. The PRECISION/PRECISION$^+$ algorithms have the same communication complexity as GT-GDA [46], but the sample complexity is a $\sqrt{n}$-factor lower than that of GT-GDA [46]. This is particularly advantageous in "big data" scenarios, where $n$ is large (i.e., the size of local datasets is large). Although the theoretical complexity bounds for PRECISION$^+$ is the same as PRECISION, the fact that PRECISION$^+$ does not need full gradient evaluations implies that PRECISION$^+$ uses significantly fewer samples than PRECISION in practice. Our numerical results in the next section will also empirically confirm this.

## 4 EXPERIMENTAL RESULTS

In this section, we conduct numerical experiments to demonstrate the performance of our proposed PRECISION and PRECISION$^+$ algorithms using a decentralized NCX-SCV regression problem on "a9a" dataset from LIBSVM repository, which is publicly available in [7]. In the supplementary material, we also provide additional experiments for environments of AUC maximization problem on dataset "a9a"[7] and 'MNIST"[18]. Due to the lack of existing algorithms for decentralized NCX-SCV with simultaneous outer and inner constraint sets (cf. Section 2.2 for details), we compare our algorithms with two stochastic algorithms as the baselines in our experiments. These baselines can be viewed as "stripped-down" versions of PRECISION /PRECISION$^+$ by removing gradient tracking or variance reduction techniques. Due to the space limitation, detailed experimental settings are relegated to our Appendix [24].

**1) Logistic Regression Model and Datasets:** We use the following decentralized NCX-SCV min-max regression problem with datasets $\{(\mathbf{a}_{ij}, b_{ij})\}_{j=1}^n$, where $\mathbf{a}_{ij} \in \mathbb{R}^d$ is the feature of the $j$-th sample of agent $i$ and $b_{ij} \in \{1, -1\}$ is the associated label:

$$\min_{\mathbf{x}_i\in\mathcal{X}} \max_{\mathbf{y}_i\in\mathcal{Y}} \frac{1}{m}\sum_{i=1}^m F_i(\mathbf{x}_i, \mathbf{y}_i), \tag{17}$$

where $F_i(\mathbf{x}_i, \mathbf{y}_i)$ is defined as:

$$F_i(\mathbf{x}_i, \mathbf{y}_i) \triangleq \frac{1}{n}\sum_{j=1}^n \left(y_{ij} l_{ij}(\mathbf{x}_i) - V(\mathbf{y}_i) + g(\mathbf{x}_i)\right). \tag{18}$$

In (18), the loss function is $l_{ij}(\mathbf{x}_i) \triangleq \log\left(1 + \exp\left(-b_{ij}\mathbf{a}_{ij}^\top \mathbf{x}_i\right)\right)$ and $g(\mathbf{x}_i)$ is a non-convex regularizer defined as: $g(\mathbf{x}_i) \triangleq \lambda_2 \sum_{k=1}^d \frac{\alpha x_{ik}^2}{1+\alpha x_{ik}^2}$, where $V(\mathbf{y}_i) = \frac{1}{2}\lambda_1 \|n\mathbf{y}_i - \mathbf{1}\|_2^2$ and we set the constraints $\mathcal{X} = [0, 10]^d, \mathcal{Y} = [0, 10]^n$. We choose constants $\lambda_1 = 1/n^2, \lambda_2 = 10^{-3}$ and $\alpha = 10$. We test the convergence performance of our algorithms using the "a9a" dataset from LIBSVM repository, which is publicly available at [7].

**2) Algorithms comparision:** Due to the very limited results of decentralized constrained min-max optimization in the literature, in our experiments, we adopt the following algorithms as our baselines for performance comparisons:

- *Prox-DSGDA* (proximal decentralized stochastic gradient descent ascent): This algorithm is motivated by DSGD [15, 32]. Each agent updates its local parameters as $\boldsymbol{\theta}_{i,t+1} = \sum_{j\in\mathcal{N}_i} [\mathbf{M}]_{ij}\boldsymbol{\theta}_{j,t} -$

$\gamma \frac{1}{|\mathcal{S}_{i,t}|} \sum_{j\in\mathcal{S}_{i,t}} \nabla_{\theta} f_{ij}(\boldsymbol{\theta}_{i,t}, \boldsymbol{\omega}_{i,t})$ and $\boldsymbol{\omega}_{i,t+1} = \sum_{j\in\mathcal{N}_i} [\mathbf{M}]_{ij} \boldsymbol{\omega}_{j,t} - \eta \frac{1}{|\mathcal{S}_{i,t}|} \sum_{j\in\mathcal{S}_{i,t}} \nabla_{\omega} f_{ij}(\boldsymbol{\theta}_{i,t}, \boldsymbol{\omega}_{i,t})$.

- *Prox-GT-SGDA* (proximal gradient-tracking-based stochastic gradient descent ascent): This algorithm is motivated by the GT-SGD algorithm [25, 50]. GT-SGDA has the same structure as that of GT-GDA, but it updates $\mathbf{v}_{i,t}$ and $\mathbf{u}_{i,t}$ using stochastic gradients as follows: $\mathbf{v}_{i,t} = \frac{1}{|\mathcal{S}_{i,t}|} \sum_{j\in\mathcal{S}_{i,t}} \nabla_{\theta} f_{ij}(\boldsymbol{\theta}_{i,t}, \boldsymbol{\omega}_{i,t})$ and $\mathbf{u}_{i,t} = \frac{1}{|\mathcal{S}_{i,t}|} \sum_{j\in\mathcal{S}_{i,t}} \nabla_{\omega} f_{ij}(\boldsymbol{\theta}_{i,t}, \boldsymbol{\omega}_{i,t})$.

**3) Results:** From Fig. 1(a) and 1(b), we can see that our proposed PRECISION$^+$ algorithm converges much faster than other algorithms (PRECISION, Prox-GT-SGDA and Prox-DSGDA) in terms of the total number of first-order oracle evaluations. We can also observe that both PRECISION and PRECISION$^+$ have lower sample complexities than those of the other two algorithms. As shown in Figs. 1(c) and 1(d), PRECISION and PRECISION$^+$ have much lower communication costs than those of Prox-DSGDA and Prox-GT-SGDA. Our experimental results thus verify our theoretical analysis that PRECISION /PRECISION$^+$ have both low sample and communication complexities in decentralized constrained min-max optimization problems.

## 5 CONCLUSION

In this paper, we studied the decentralized constrained non-convex-strongly-concave (NCX-SCV) min-max optimization and developed two algorithms called PRECISION and PRECISION$^+$. We showed that, to achieve an $\epsilon$-stationary point of a decentralized constrained NCX-SCV min-max problem, PRECISION and PRECISION$^+$ achieve the communication complexity of $O(\epsilon^{-2})$ and sample complexity of $O(m\sqrt{n}\epsilon^{-2})$, where $m$ is the number of agents and $n$ is the size of dataset for each agent. Our numerical studies also verified the theoretical performance of our proposed algorithms. We note that decentralized constrained min-max learning remains an under-explored area, and our work opens up several interesting directions for future research. For example, the agents need to send outer and inner model parameter pairs to their neighbors in our algorithm, both of which could be high dimensional. In our future work, it would be interesting to adopt communication-efficient mechanisms (e.g., compression techniques) to further reduce the communication cost, especially for large-scale deep learning models.

## ACKNOWLEDGMENTS

This work has been supported in part by NSF grants CAREER CNS-2110259 and CNS-2112471.

## REFERENCES

[1] Kamal Ali and Wijnand Van Stam. 2004. TiVo: Making show recommendations using a distributed collaborative filtering architecture. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 394–401.

[2] Sina Baharlouei, Maher Nouiehed, Ahmad Beirami, and Meisam Razaviyayn. 2019. Rényi Fair Inference. *arXiv preprint arXiv:1906.12005* (2019).

[3] Anouer Bennajeh, Slim Bechikh, Lamjed Ben Said, and Samir Aknine. 2019. Bi-level decision-making modeling for an autonomous driver agent: application in the car-following driving behavior. *Applied Artificial Intelligence* 33, 13 (2019), 1157–1178.

[4] Stephen Boyd, Neal Parikh, and Eric Chu. 2011. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.

[5] Duncan S Callaway and Ian A Hiskens. 2010. Achieving controllability of electric loads. *Proc. IEEE* 99, 1 (2010), 184–199.

[6] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. 2012. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics* 9, 1 (2012), 427–438.

[7] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2, 3 (2011), 1–27.

[8] Jianshu Chen, Zaid J Towfic, and Ali H Sayed. 2014. Dictionary learning over distributed models. *IEEE Transactions on Signal Processing* 63, 4 (2014), 1001–1016.

[9] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. 2004. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation* 20, 2 (2004), 243–255.

[10] Emiliano Dall'Anese, Hao Zhu, and Georgios B Giannakis. 2013. Distributed optimal power flow for smart microgrids. *IEEE Transactions on Smart Grid* 4, 3 (2013), 1464–1475.

[11] Yashar Deldjoo, Tommaso Di Noia, and Felice Antonio Merra. 2020. Adversarial machine learning in recommender systems (aml-recsys). In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 869–872.

[12] Damien Ernst, Mevludin Glavic, and Louis Wehenkel. 2004. Power systems stability control: reinforcement learning framework. *IEEE Transactions on Power Systems* 19, 1 (2004), 427–435.

[13] Mevludin Glavic, Raphaël Fonteneau, and Damien Ernst. 2017. Reinforcement learning for electric power system decision and control: Past considerations and perspectives. *IFAC-PapersOnLine* 50, 1 (2017), 6918–6927.

[14] Mingyi Hong, Siliang Zeng, Junyu Zhang, and Haoran Sun. 2022. On the Divergence of Decentralized Nonconvex Optimization. *SIAM Journal on Optimization* 32, 4 (2022), 2879–2908.

[15] Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. 2017. Collaborative deep learning in fixed topology networks. In *Advances in Neural Information Processing Systems*. 5904–5914.

[16] Zhanhong Jiang, Kushal Mukherjee, and Soumik Sarkar. 2018. On Consensus-Disagreement Tradeoff in Distributed Optimization. In *2018 Annual American Control Conference (ACC)*. IEEE, 571–576.

[17] Jens Kober, J Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (2013), 1238–1274.

[18] Yann LeCun, Corinna Cortes, and CJ Burges. 1998. MNIST handwritten digit database. *Available: http://yann. lecun. com/exdb/mnist* (1998).

[19] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. 2017. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*. 5330–5340.

[20] Tianyi Lin, Chi Jin, and Michael Jordan. 2020. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*. PMLR, 6083–6093.

[21] Tianyi Lin, Chi Jin, and Michael I Jordan. 2020. Near-optimal algorithms for minimax optimization. In *Proceedings of Conference on Learning Theory*. PMLR, 2738–2779.

[22] Mingrui Liu, Wei Zhang, Youssef Mroueh, Xiaodong Cui, Jarret Ross, Tianbao Yang, and Payel Das. 2020. A Decentralized Parallel Algorithm for Training Generative Adversarial Nets. In *Proceedings of Advances in Neural Information Processing Systems*, Vol. 33.

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

[24] Zhuqing Liu, Xin Zhang, Songtao Lu, and Jia Liu. [n. d.]. *PRECISION: Decentralized Constrained Min-Max Learning with Low Communication and Sample Complexities*.

[25] Songtao Lu, Xinwei Zhang, Haoran Sun, and Mingyi Hong. 2019. GNSD: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*. IEEE, 315–321.

[26] Luo Luo, Haishan Ye, Zhichao Huang, and Tong Zhang. 2020. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *arXiv preprint arXiv:2001.03724* (2020).

[27] Ali Mansoor, Xiaoxu Diao, and Carol Smidts. 2021. BACKWARD FAILURE PROPAGATION FOR CONCEPTUAL SYSTEM DESIGN USING ISFA.

[28] Ali Mansoor, Xiaoxu Diao, and Carol Smidts. 2023. A Method for Backward Failure Propagation in Conceptual System Design. *Nuclear Science and Engineering* (2023).

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

[30] Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. 2020. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1497–1507.

[31] Angelia Nedić, Alex Olshevsky, and Michael G Rabbat. 2018. Network topology and communication-computation tradeoffs in decentralized optimization. *Proc. IEEE* 106, 5 (2018), 953–976.

[32] Angelia Nedic and Asuman Ozdaglar. 2009. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Automat. Control* 54, 1 (2009), 48.

[33] Yuji Nozaki and Takamichi Nakamoto. 2018. Predictive modeling for odor character of a chemical using machine learning combined with natural language processing. *PloS one* 13, 6 (2018), e0198475.

[34] Petter Ogren, Edward Fiorelli, and Naomi Ehrich Leonard. 2004. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Trans. Automat. Control* 49, 8 (2004), 1292–1302.

[35] Fatih Özyurt. 2020. Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures. *The Journal of Supercomputing* 76, 11 (2020), 8413–8431.

[36] Athanasios S Polydoros and Lazaros Nalpantidis. 2017. Survey of model-based reinforcement learning: Applications on robotics. *Journal of Intelligent & Robotic Systems* 86, 2 (2017), 153–173.

[37] Shuang Qiu, Zhuoran Yang, Xiaohan Wei, Jieping Ye, and Zhaoran Wang. 2020. Single-Timescale Stochastic Nonconvex-Concave Optimization for Smooth Nonlinear TD Learning. *arXiv preprint arXiv:2008.10103* (2020).

[38] Michael Rabbat and Robert Nowak. 2004. Distributed optimization in sensor networks. In *Proceedings of Iternational Symposium on Information Processing in Sensor Networks*. 20–27.

[39] Wei Ren, Randal W Beard, and Ella M Atkins. 2007. Information consensus in multivehicle cooperative control. *IEEE Control systems magazine* 27, 2 (2007), 71–82.

[40] Seung Hyong Rhee, Hwa-Sung Kim, and Seung-Won Sohn. 2012. The effect of decentralized resource allocation in network-centric warfare. In *The International Conference on Information Network 2012*. IEEE, 478–481.

[41] Prasanna Sattigeri, Samuel C Hoffman, Vijil Chenthamarakshan, and Kush R Varshney. 2018. Fairness gan. *arXiv preprint arXiv:1805.09910* (2018).

[42] Gesualdo Scutari and Ying Sun. 2019. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming* 176, 1 (2019), 497–544.

[43] Keng Siau and Weiyu Wang. 2018. Building trust in artificial intelligence, machine learning, and robotics. *Cutter business technology journal* 31, 2 (2018), 47–53.

[44] William D Smart and L Pack Kaelbling. 2002. Effective reinforcement learning for mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 4. 3404–3410.

[45] Haoran Sun, Songtao Lu, and Mingyi Hong. 2020. Improving the Sample and Communication Complexity for Decentralized Non-Convex Optimization: Joint Gradient Estimation and Tracking. In *Proceedings of International Conference on Machine Learning*. PMLR, 9217–9228.

[46] Ioannis Tsaknakis, Mingyi Hong, and Sijia Liu. 2020. Decentralized Min-Max Optimization: Formulations, Algorithms and Applications in Network Poisoning Attack. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5755–5759.

[47] Hoi-To Wai, Zhuoran Yang, Zhaoran Wang, and Mingyi Hong. 2018. Multi-agent reinforcement learning via double averaging primal-dual optimization. *arXiv preprint arXiv:1806.00877* (2018).

[48] Abraham Wald. 1945. Statistical decision functions which minimize the maximum risk. *Annals of Mathematics* (1945), 265–280.

[49] Weiran Wang, Jialei Wang, Mladen Kolar, and Nathan Srebro. 2018. Distributed stochastic multi-task learning with graph regularization. *arXiv preprint arXiv:1802.03830* (2018).

[50] Ran Xin, Usman A Khan, and Soummya Kar. 2020. An improved convergence analysis for decentralized online stochastic non-convex optimization. *arXiv preprint arXiv:2008.04195* (2020).

[51] Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. 2021. To be robust or to be fair: Towards fairness in adversarial training. In *International Conference on Machine Learning*. PMLR, 11492–11501.

[52] Tengyu Xu, Zhe Wang, Yingbin Liang, and H Vincent Poor. 2020. Enhanced first and zeroth order variance reduced algorithms for min-max optimization. *arXiv preprint arXiv:2006.09361* (2020).

[53] Xin Zhang, Jia Liu, and Zhengyuan Zhu. 2019. Distributed Linear Model Clustering over Networks: A Tree-Based Fused-Lasso ADMM Approach. *arXiv preprint arXiv:1905.11549* (2019).

[54] Xin Zhang, Zhuqing Liu, Jia Liu, Zhengyuan Zhu, and Songtao Lu. 2021. Taming communication and sample complexities in decentralized policy evaluation for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 18825–18838.

[55] Ke Zhou and Stergios I Roumeliotis. 2011. Multirobot active target tracking with combinations of relative observations. *IEEE Transactions on Robotics* 27, 4 (2011), 678–695.


# A PROOF SKETCH OF MAIN RESULTS

Due to space limitation, we outline the key steps of the proofs of Theorems 1 and 2. The complete version of our proofs is available in our technical report [24]. Before diving in our theoretical analysis, we first provide the following notations:

- $\bar{\mathbf{x}}_t = \frac{1}{m}\sum_{i=1}^m \mathbf{x}_{i,t}$ and $\mathbf{x}_t = [\mathbf{x}_{1,t}^\top, \cdots, \mathbf{x}_{m,t}^\top]^\top$ for any vector $\mathbf{x}$;
- $\nabla_{\mathbf{x}} F_t = [\nabla_{\mathbf{x}} F(\mathbf{x}_{1,t}, \mathbf{y}_{1,t})^\top, \cdots, \nabla_{\mathbf{x}} F(\mathbf{x}_{m,t}, \mathbf{y}_{m,t})^\top]^\top$;
- $\nabla_{\mathbf{y}} F_t = [\nabla_{\mathbf{y}} F(\mathbf{x}_{1,t}, \mathbf{y}_{1,t})^\top, \cdots, \nabla_{\mathbf{y}} F(\mathbf{x}_{m,t}, \mathbf{y}_{m,t})^\top]^\top$;
- $\mathcal{E}(\mathbf{x}_t) = \frac{1}{m}\sum_{i=1}^m \|\mathbf{x}_{i,t} - \bar{\mathbf{x}}_t\|^2$ for any vector $\mathbf{x}$.

Also, the result below is useful for our subsequent analysis.

Lemma 1. *Under Assumption 1, the funciton $J(\mathbf{x}) = F(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ w.r.t $\mathbf{x}$ is Lipschitz smooth, i.e., there exists a positive constant $L_J$, such that*

$$\|\nabla J(\mathbf{x}) - \nabla J(\mathbf{x}')\| \le L_J \|\mathbf{x} - \mathbf{x}'\|, \ \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d, \tag{19}$$

*where the Lipschitz constant is $L_J = L_f + L_f^2/\mu$ for Algorithm 1. This lemma follows immediately from Lemma 4.3 in [20].*

Lemma 2. *Under Assumption 1, $\mathbf{y}^*(\mathbf{x}) = \arg\max_{\mathbf{y}} F(\mathbf{x}, \mathbf{y})$ is Lipschitz continuous, i.e., there exists a positive constant $L_y$, such that*

$$\|\mathbf{y}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x}')\| \le L_y \|\mathbf{x} - \mathbf{x}'\|, \ \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d, \tag{20}$$

*where the Lipschitz constant is $L_y = L_f/\mu$.*

## A.1 Important Lemmas for Proving Main Theorems

We first show the following descent property of PRECISION algorithm on the function $Q(\cdot)$, which is stated in the following lemma:

Lemma 3 (Descent Inequality on $Q(\mathbf{x})$). *Under Assumption 1, the following descent inequality holds:*

$$Q(\bar{\mathbf{x}}_{t+1}) - Q(\bar{\mathbf{x}}_t) \le \frac{\nu L_F^2}{2\beta}\left\|\mathbf{y}_t^* - \bar{\mathbf{y}}_t\right\|^2 + \frac{\nu}{2\beta}\left\|\nabla_{\mathbf{x}} F(\bar{\mathbf{x}}_t, \bar{\mathbf{y}}_t) - \bar{\mathbf{p}}_t\right\|^2 + \frac{\nu\tau}{2\beta m}\|\mathbf{x}_t - 1\otimes\bar{\mathbf{x}}_t\|^2 - \left(\frac{\nu\tau}{m} - \frac{\nu^2 L_J}{2m} - \frac{\nu\beta}{m} - \frac{\nu\tau\beta}{2m}\right)\|\tilde{\mathbf{x}}_t - 1\otimes\bar{\mathbf{x}}_t\|^2. \tag{21}$$

*where $Q(\mathbf{x}_t) = \max_{\mathbf{y}} F(\mathbf{x}_t, \mathbf{y}) + h(\mathbf{x}_t)$ and $\mathbf{y}_t^* = \arg\max_{\mathbf{y}} F(\bar{\mathbf{x}}_t, \mathbf{y})$.*

Next, consider the error bound $\left\|\bar{\mathbf{y}}_t - \mathbf{y}_t^*\right\|^2$ in Lemma 3, we have the following Lemma:

Lemma 4 (Error Bound on $\mathbf{y}^*(\mathbf{x})$). *Under Assumption 1, the following inequality holds for* PRECISION/PRECISION$^+$ *:*

$$\left\|\bar{\mathbf{y}}_{t+1} - \mathbf{y}_{t+1}^*\right\|^2 \le \left(1 - \frac{\mu\eta\alpha}{4}\right)\left\|\bar{\mathbf{y}}_t - \mathbf{y}_t^*\right\|^2 - \frac{3\eta}{4}\|\tilde{\mathbf{y}}_t - 1\otimes\bar{\mathbf{y}}_t\|^2 + \frac{75\eta\alpha}{16\mu}\left\|\bar{\mathbf{d}}_t - \nabla_{\mathbf{y}} F(\bar{\mathbf{x}}_t, \bar{\mathbf{y}}_t)\right\|^2 + \frac{17L_y^2\nu^2}{2\mu\eta\alpha m}\|\tilde{\mathbf{x}}_t - 1\otimes\bar{\mathbf{x}}_t\|^2. \tag{22}$$

By telescoping the combined results of previous lemmas from 0 to $T+1$ and after some rearrangements, we arrive at the following results:

Lemma 5. *Under Assumption 1 and condition $\eta \le 1/2L_f$, the following inequality holds for* PRECISION/PRECISION$^+$ *:*

$$
\begin{aligned}
&Q(\bar{\mathbf{x}}_{T+1})-Q(\bar{\mathbf{x}}_0)+\frac{4\nu L_F^2}{\beta\mu\eta^2}\big[\|\bar{\mathbf{y}}_{T+1}-\mathbf{y}^*_{T+1}\|^2-\|\mathbf{y}^*_0-\bar{\mathbf{y}}_0\|^2\big]\\
\le&\frac{4\nu L_F^2}{\beta\mu\eta\alpha}\Big\{-\frac{3\eta}{4}\|\tilde{\mathbf{y}}_t-1\otimes\bar{\mathbf{y}}_t\|^2+\frac{75\eta\alpha}{16\mu}\frac{2}{m}\|\nabla_{\mathbf{y}}F(\mathbf{x}_t,\mathbf{y}_t)-\bar{\mathbf{d}}_t\|^2\\
&+\frac{17L_y^2\nu^2}{2\mu m\eta\alpha}\|\tilde{\mathbf{x}}_t-1\otimes\bar{\mathbf{x}}_t\|^2\Big\}+\frac{\nu}{2\beta}\frac{2}{m}\|\nabla_{\mathbf{x}}F(\mathbf{x}_t,\mathbf{y}_t)-\bar{\mathbf{p}}_t\|^2\\
&+\frac{\nu\tau}{2\beta m}\|\mathbf{x}_t-1\otimes\bar{\mathbf{x}}_t\|^2-\Big(\frac{\nu\tau}{m}-\frac{\nu^2L_J}{2m}-\frac{\nu\beta}{m}-\frac{\nu\tau\beta}{2m}\Big)\\
&\cdot\|\tilde{\mathbf{x}}_t-1\otimes\bar{\mathbf{x}}_t\|^2+\Big[\frac{\nu}{\beta}\frac{L_F^2}{m}+\frac{4\nu L_F^2}{\beta\mu\eta\alpha}\frac{75\eta\alpha}{16\mu}\frac{2L_F^2}{m}\Big]\sum_{i=1}^{m}\big[\|\bar{\mathbf{x}}_t-\mathbf{x}_{i,t}\|^2\\
&+\|\bar{\mathbf{y}}_t-\mathbf{y}_{i,t}\|^2\big]-\frac{\nu L_F^2}{2\beta}\|\bar{\mathbf{y}}_t-\mathbf{y}_t^*\|^2. \qquad (23)
\end{aligned}
$$

Next, we bound the iterates contraction of $\|\mathbf{x}_t-1\otimes\bar{\mathbf{x}}_t\|^2$ and $\|\mathbf{y}_t-1\otimes\bar{\mathbf{y}}_t\|^2$ in (23).

Lemma 6 (Iterates Contraction). *The following contraction properties of the iterates hold:*

$$
\begin{aligned}
\|\mathbf{x}_t-1\otimes\bar{\mathbf{x}}_t\|^2&\le(1+c_1)\lambda^2\|\mathbf{x}_{t-1}-1\otimes\bar{\mathbf{x}}_{t-1}\|^2\\
&\quad+(1+\frac{1}{c_1})\nu^2\|\tilde{\mathbf{x}}_{t-1}-\mathbf{x}_{t-1}\|^2,\\
\|\mathbf{y}_t-1\otimes\bar{\mathbf{y}}_t\|^2&\le(1+c_2)\lambda^2\|\mathbf{y}_{t-1}-1\otimes\bar{\mathbf{y}}_{t-1}\|^2\\
&\quad+(1+\frac{1}{c_2})\eta^2\|\tilde{\mathbf{y}}_{t-1}-\mathbf{y}_{t-1}\|^2, \qquad (24)
\end{aligned}
$$

*where $c_1$ and $c_2$ are arbitrary positive constants. Additionally, we have*

$$
\begin{aligned}
\|\mathbf{x}_t-\mathbf{x}_{t-1}\|^2&\le 8\mathcal{E}(\mathbf{x}_{t-1})+2\nu^2\|\tilde{\mathbf{x}}_{t-1}-\mathbf{x}_{t-1}\|^2,\\
\|\mathbf{y}_t-\mathbf{y}_{t-1}\|^2&\le 8\mathcal{E}(\mathbf{y}_{t-1})+2\eta^2\|\tilde{\mathbf{y}}_{t-1}-\mathbf{y}_{t-1}\|^2. \qquad (25)
\end{aligned}
$$

Next, we bound the gradient tracking errors $\sum_{t=0}^T\|\bar{\mathbf{d}}_t-\nabla_{\mathbf{x}}F_t\|^2$ and $\sum_{t=0}^T\|\bar{\mathbf{p}}_t-\nabla_{\mathbf{y}}F_t\|^2$ in (23).

Lemma 7 (Error of Gradient Estimator). *Under Assumption 2, we have the following error bounds for the gradient trackers:*

$$
\begin{aligned}
\sum_{t=0}^T\|\bar{\mathbf{d}}_t-\nabla_{\mathbf{x}}F_t\|^2&\le\sum_{t=1}^T\mathbb{E}\|\bar{\mathbf{d}}_{(n_t-1)q}-\nabla_{\mathbf{x}}F(\mathbf{x}_{(n_t-1)q},\\
&\quad\mathbf{y}_{(n_t-1)q})\|^2+L_f^2\big(\|\mathbf{x}_t-\mathbf{x}_{t-1}\|^2+\|\mathbf{y}_t-\mathbf{y}_{t-1}\|^2\big), \qquad (26)\\
\sum_{t=0}^T\|\bar{\mathbf{p}}_t-\nabla_{\mathbf{y}}F_t\|^2&\le\sum_{t=1}^T\mathbb{E}\|\bar{\mathbf{p}}_{(n_t-1)q}-\nabla_{\mathbf{y}}F(\mathbf{x}_{(n_t-1)q},\\
&\quad\mathbf{y}_{(n_t-1)q})\|^2+L_f^2\big(\|\mathbf{x}_t-\mathbf{x}_{t-1}\|^2+\|\mathbf{y}_t-\mathbf{y}_{t-1}\|^2\big), \qquad (27)
\end{aligned}
$$

*where $n_t$ is the largest positive integer satisfing $(n_t-1)q\le t$.*

Proof Sketch of Lemma 7. Define

$$
\begin{aligned}
A_{i,t}=&\bar{\mathbf{d}}_{i,t}-\nabla_{\mathbf{x}}F_{i,t};\ B_{i,t}=\frac{1}{|\mathcal{S}_{i,t}|}\sum_{j\in\mathcal{S}_{i,t}}\nabla_{\mathbf{x}}f_{i,t}(\mathbf{x}_{i,t},\mathbf{y}_{i,t})\\
&-\nabla_{\mathbf{x}}f_{i,t}(\mathbf{x}_{i,t-1},\mathbf{y}_{i,t-1})+\nabla_{\mathbf{x}}F_{i,t-1}-\nabla_{\mathbf{x}}F_{i,t}. \qquad (28)
\end{aligned}
$$

Note that $\mathbb{E}_t[B_{i,t}]=0$, where the expectation is taken over the randomness of data sampling at the $t$-th iteration. Thus,

$$
\mathbb{E}_t\|A_{i,t}\|^2=\|A_{i,t-1}\|^2+\mathbb{E}_t\|B_{i,t}\|^2. \qquad (29)
$$

Also, with $|\mathcal{S}_{i,t}|=q$, we have

$$
\mathbb{E}_t\|B_{i,t}\|^2\le\frac{L_f^2}{q}\big(\|\mathbf{x}_{i,t}-\mathbf{x}_{i,t-1}\|^2+\|\mathbf{y}_{i,t}-\mathbf{y}_{i,t-1}\|^2\big). \qquad (30)
$$

Taking full expectation and telescoping (30) over $t$ from $(n_t-1)q+1$ to $t$, where $t\le n_tq-1$, we have $\mathbb{E}\|A_t\|^2\le\mathbb{E}\|A_{(n_t-1)q}\|^2+\sum_{r=(n_t-1)q+1}^t\frac{L_f^2}{q}\mathbb{E}\big(\|\mathbf{x}_r-\mathbf{x}_{r-1}\|^2+\|\mathbf{y}_r-\mathbf{y}_{r-1}\|^2\big)$. Thus, $\sum_{k=0}^t\mathbb{E}\|A_k\|^2\le\sum_{r=0}^t\|A_{(n_r-1)q}\|^2+\sum_{r=1}^tL_f^2\big(\|\mathbf{x}_r-\mathbf{x}_{r-1}\|^2+\|\mathbf{y}_r-\mathbf{y}_{r-1}\|^2\big)$. We have similar result while $A_{i,t}=\bar{\mathbf{p}}_{i,t}-\nabla_{\mathbf{y}}F_{i,t}$. This completes the proof of of Lemma. 7. □

## A.2 Proof Sketch of Theorem 1

Proof. Following the defined potential function $\mathfrak{p}$ and the result of Lemma 3-7, we have

$$
\begin{aligned}
\mathbb{E}\mathfrak{p}_{T+1}-\mathfrak{p}_0\le&\nu{L_f}^2 2\sum_{t=0}^T\big\|\bar{\mathbf{y}}_t-\mathbf{y}_t^*\big\|^2\\
&-C_1\sum_{t=0}^T\sum_{i=1}^m\|\bar{\mathbf{x}}_t-\mathbf{x}_{i,t}\|^2-C_2\sum_{t=0}^T\|\tilde{\mathbf{x}}_t-1\otimes\bar{\mathbf{x}}_t\|^2\\
&-C_3\sum_{t=0}^T\sum_{i=1}^m[\|\bar{\mathbf{y}}_t-\mathbf{y}_{i,t}\|^2]-C_4\sum_{t=0}^T\|\widetilde{\mathbf{y}}_t-1\otimes\bar{\mathbf{y}}_t\|^2, \qquad (31)
\end{aligned}
$$

$C_1,C_2,C_3,C_4$ are some constants and can be found in Eqs.(98) - Eqs.(101) in our technical report [24]. Suppose that $\beta\le\min\left\{\frac{\tau}{12},\frac{1}{3}\right\}$, $\alpha\le\frac{1}{4L_f}$ hold and let $c_1=\frac{1-\lambda^2}{1+\lambda^2}$, if step-sizes satisfy Thm. 1 to ensure $C_1,C_2,C_3,C_4\ge0$. We can conclude that

$$
\frac{1}{T+1}\sum_{t=0}^T\mathfrak{M}_t\le\frac{\mathbb{E}[\mathfrak{p}_0-Q^*]}{\min\{C_1,C_2,\nu L_f^2/2\}(T+1)}. \qquad (32)
$$

This completes the proof Theorem 1. □

## A.3 Proof Sketch of Theorem 2

Proof. For PRECISION$^+$ , we have

$$
\begin{aligned}
&\mathbb{E}\|\bar{\mathbf{d}}_{(n_t-1)q}-\nabla_{\mathbf{x}}F_{(n_t-1)q}\|^2\\
&=\mathbb{E}\|\bar{\mathbf{p}}_{(n_t-1)q}-\nabla_{\mathbf{y}}F_{(n_t-1)q}\|^2=\frac{I_{(\mathcal{N}_s<M)}}{\mathcal{N}_s}\sigma^2. \qquad (33)
\end{aligned}
$$

Recall that $\mathcal{N}_s=\min\{c_\gamma\sigma^2(\gamma^{(k)})^{-1},c_\epsilon\sigma^2\epsilon^{-1},M\}$, we have

$$
\frac{I_{(\mathcal{N}_s<M)}}{\mathcal{N}_s}\le\max\{\frac{\gamma^{(k)}}{c_\gamma\sigma^2},\frac{\epsilon}{c_\epsilon\sigma^2}\}\le\frac{\gamma^{(k)}}{c_\gamma\sigma^2}+\frac{\epsilon}{c_\epsilon\sigma^2}. \qquad (34)
$$

Since $\gamma_{t+1}=\frac{1}{q}\sum_{i=(n_t-1)q}^t\|\tilde{\mathbf{x}}_t-1\otimes\bar{\mathbf{x}}_t\|^2$. Plugging (34) to Lemma 5, we have the following result, with additional parameter setting $c_\gamma\ge(\frac{75\eta\alpha}{8\mu}\frac{1}{m}+\frac{\nu}{\beta}\frac{1}{m})\frac{\nu\tau}{12}$. For PRECISION$^+$, following the defined potential function $\mathfrak{p}$ and the result of Lemma 3-7, with $\mathfrak{p}_{T+1}\ge Q^*$, we reach the conclusion. □